

Барский А.Б.

Логические нейронные сети



ИНТУИТ

НАЦИОНАЛЬНЫЙ ОТКРЫТЫЙ УНИВЕРСИТЕТ

Барский А.Б.

Логические нейронные сети

2-е издание, исправленное

Барский А.Б.

Национальный Открытый Университет "ИНТУИТ"

2016

УДК 004.032.26(07)

ББК 8

Б26

Логические нейронные сети / Барский А.Б. - М.: Национальный Открытый Университет "ИНТУИТ", 2016 (Основы информационных технологий)

ISBN 978-5-94774-646-4

На основе положений математической логики событий исследуется возможность построения логических нейронных сетей, выполняющих операции вывода в составе систем искусственного интеллекта, имитирующих механизмы работы мозга.

Предлагаются методы построения обученных нейронных сетей, созданных «под задачу», простые методы обучения-трассировки, методы преобразования описаний систем принятия решений для повышения достоверности выводов. Рассматривается возможность применения логических нейронных сетей в самообучающихся системах управления, а также в различных системах экономики, транспорта, безопасности, защиты информации, при решении задач интеллектуального отображения, в бизнесе туризма и развлечений, при политическом и социальном прогнозировании и в других задачах науки и искусства.

(с) ООО "ИНТУИТ.РУ", 2007-2016

(с) Барский А.Б., 2007-2016

Математическая логика событий

Приводятся основные положения раздела математической логики - алгебры высказываний. Совершается переход от булевых значений переменных - высказываний - к действительным переменным, отображающим достоверность высказываний о событиях. Представляются основные положения и выводы, необходимые при построениях в области логических нейронных сетей.

1.1 Булева концепция алгебры высказываний о событиях

" Ученые объяснения большей частью производят то впечатление, что бывшее ясно и понятно становится темно и запутанно".

Определение 1. Предполагаемое или свершившееся действие, его фигурант, результат, а также условия свершения, называются событием .

Определение 2. Событие выражается высказыванием о его свершении.

Высказыванию о событии (далее - просто высказывание, считая событие и высказывание о нем синонимами) можно поставить в соответствие переменную, которая в рамках булевой концепции может принимать значение ИСТИНА (1) или ЛОЖЬ (0).

Например:

$x = \langle \text{поезд опоздал на пять минут} \rangle;$

$y = \langle \text{в данной операции принимал участие Вася} \rangle$
(достаточно сообщить лишь имя);

$z = \langle \text{скорость автомобиля принадлежит диапазону (120-140 км/ч)} \rangle$
(достаточно кратко обозначить диапазон в известном контексте, как условие свершения некоторого действия, приведшего к автокатастрофе).

Очевидно, что каждая переменная x , y , z может принимать одно из двух значений - 0 или 1.

Над высказываниями производятся логические операции. В рамках

последующих построений потребуются четыре операции: отрицание ($\neg x$, НЕ x , \bar{x}), конъюнкция (\wedge , И, AND, \times), дизъюнкция (\vee , ИЛИ, OR) импликация или операция следования (\rightarrow). Результаты операций определяются таблично.

Предполагая достаточные знания слушателей, можно напомнить:

- одноместная операция отрицания меняет значение переменной на противоположное;
- двуместная операция конъюнкции над двумя и (рекурсивно) более переменными порождает значение 1 тогда и только тогда, когда все переменные имеют значение 1;
- двуместная операция дизъюнкции над двумя и (рекурсивно) более переменными порождает значение 1, когда хотя бы одна переменная имеет значение 1;
- переменная справа от знака операции следования (импликации) принимает значение 1 тогда и только тогда, когда выражение слева от этого знака имеет значение 1.

Кроме того, ниже используется операция ИСКЛЮЧАЮЩЕЕ ИЛИ, предполагающая возможность лишь единственного вхождения переменной со значением 1 в операцию дизъюнкции, объединяющую несколько переменных.

Переход от высказываний к их булевой интерпретации, к булевым переменным, вводит в действие все законы, свойства и правила эквивалентных преобразований, известные из булевой алгебры.

Закон коммутативности:

$$\begin{aligned}x \wedge y &= y \wedge x; \\x \vee y &= y \vee x\end{aligned}\quad (1.1)$$

Закон ассоциативности:

$$\begin{aligned}x \vee (y \vee z) &= (x \vee y) \vee z; \\x \wedge (y \wedge z) &= (x \wedge y) \wedge z\end{aligned}\quad (1.2)$$

Закон дистрибутивности:

$$\begin{aligned}x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z); \\x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z)\end{aligned} \quad (1.3)$$

Закон де Моргана:

$$\begin{aligned}\overline{x \vee y} &= \bar{x} \wedge \bar{y}; \\ \overline{x \wedge y} &= \bar{x} \vee \bar{y}\end{aligned} \quad (1.4)$$

Закон идемпотенции:

$$\begin{aligned}x \vee x &= x; \\ x \wedge x &= x\end{aligned} \quad (1.5)$$

Закон поглощения:

$$\begin{aligned}x \vee (x \wedge y) &= x; \\ x \wedge (x \vee y) &= x\end{aligned} \quad (1.6)$$

Закон склеивания:

$$\begin{aligned}(x \wedge y) \vee (\bar{x} \wedge y) &= y; \\ (x \vee y) \wedge (\bar{x} \vee y) &= y\end{aligned} \quad (1.7)$$

Операция переменной с инверсией:

$$\begin{aligned}x \vee \bar{x} &= 1; \\ x \wedge \bar{x} &= 0\end{aligned} \quad (1.8)$$

Операция с константами:

$$\begin{aligned}x \wedge 0 &= 0, \quad x \wedge 1 = x; \\ x \vee 0 &= x, \quad x \vee 1 = 1\end{aligned} \quad (1.9)$$

Двойное отрицание:

$$\overline{\bar{x}} = x \quad (1.10)$$

Несмотря на наличие дистрибутивных операций, существует

ранжирование операций - в сторону понижения (ранга) слева направо: $\neg(x)$, \wedge , \vee . То есть если написано без скобок $\neg x \vee y \wedge z$, то с помощью эквивалентного обозначения и скобок можно выявить следующий порядок действий: $\overline{x} \vee (y \wedge z)$.

1.2. Логические функции высказываний

Множество логических переменных - высказываний о событиях $\{x_1, x_2, \dots, x_n\}$ в контексте некоторого приложения образует пространство событий размерности n . Точка этого пространства является ситуацией.

Можно записать произвольную композицию на основе заданного множества переменных-высказываний и логических операций, например, $\neg \vee xy \wedge z, (x \vee \overline{y}) \wedge (y \wedge z)$. Почему первую композицию следует считать бессмысленной? По-видимому, потому, что она содержит конструкции, не определенные в терминах алгебры логики, и не может быть исчерпывающим образом преобразована в таковые на основе применения (1)-(10). Тогда вторая приведенная композиция имеет смысл, т.к. полностью подвержена основным определениям операций алгебры логики и правилам преобразования в ней.

Высказывания (о событиях) в качестве переменных могут входить в состав сложных формирований - логических функций, принимающих (булевы) значения 1 (ИСТИНА) или 0 (ЛОЖЬ).

Определение 3. Имеющая смысл линейно-скобочная композиция операций \neg, \vee, \wedge над переменными-высказываниями x_1, x_2, \dots, x_n , образующими пространство событий, задает логическую функцию $f(x_1, x_2, \dots, x_n)$, принимающую для различных ситуаций, т.е. наборов значений переменных, значения 0 или 1.

Таким образом, логическая функция является булевой функцией ситуаций.

В классической теории булевых функций [1] показывается, что каждая

такая функция может быть представлена дизъюнктивной и (или) конъюнктивной нормальной формой. В первом случае ее структура выражается как дизъюнкция конъюнкций, во втором - как конъюнкция дизъюнкций.

Рассмотрим две логические функции

$$Y = x_1 \vee (x_2 \wedge \overline{x_1} \vee x_3) \text{ и}$$

$$Z = (x_1 \vee \overline{x_2}) \wedge (\overline{x_3} \vee x_2).$$

Выражение Y представлено дизъюнктивной нормальной формой (ДНФ). Выражение Z соответствует конъюнктивной нормальной форме (КНФ), практически не применяемой.

Преобразуем

$$Z = (x_1 \wedge \overline{x_3}) \vee (x_1 \wedge x_2) \vee (\overline{x_2} \wedge \overline{x_3}). \quad (\text{Учитывается, что } \overline{x_2} \wedge x_2 = 0.)$$

Это - дизъюнктивная нормальная форма.

Практически, например, при конструировании электронных устройств, известно наперед, какой сигнал на отдельно взятом выходе должен формироваться при различных значениях сигналов на входе. Тогда значения логической функции, описывающей формирование сигнала на данном выходе, задаются таблично, в зависимости от всех возможных ситуаций на входе. По такой таблице аналитическое выражение для искомой логической функции формируется в виде совершенной дизъюнктивной нормальной формы (СДНФ). Ее общий вид продемонстрируем на примере трех переменных:

$$f = f(0, 0, 0) \wedge (\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3}) \vee f(0, 0, 1) \wedge (\overline{x_1} \wedge \overline{x_2} \wedge x_3) \vee f(0, 1, 0) \wedge (\overline{x_1} \wedge x_2 \wedge \overline{x_3}) \vee f(0, 1, 1) \wedge (\overline{x_1} \wedge x_2 \wedge x_3) \vee f(1, 0, 0) \wedge (x_1 \wedge \overline{x_2} \wedge \overline{x_3}) \vee f(1, 0, 1) \wedge (x_1 \wedge \overline{x_2} \wedge x_3) \vee f(1, 1, 0) \wedge (x_1 \wedge x_2 \wedge \overline{x_3}) \vee f(1, 1, 1) \wedge (x_1 \wedge x_2 \wedge x_3)$$

Для всех значений переменных рассчитаем значения приведенных

выше логических функций Y и Z (табл. 1.1).

Таблица 1.1.

Значения
логических
функций

x_1	x_2	x_3	Y	Z
0	0	0	0	1
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

Попытаемся построить приведенные выше функции Y и Z на основе их СДНФ, т.е. проверим правильность такого подхода:

$$Y^* = \overline{x_1} \wedge x_2 \wedge x_3 \vee x_1 \wedge \overline{x_2} \wedge \overline{x_3} \vee x_1 \wedge \overline{x_2} \wedge x_3 \vee x_1 \wedge x_2 \wedge \overline{x_3} \vee x_1 \wedge x_2 \wedge x_3$$

После эквивалентных преобразований (начинающихся с вынесения x_1 "за скобку") получим

$$Y^* = \overline{x_1} \wedge x_2 \wedge x_3 \vee x_1, \text{ что совпадает с видом } Y.$$

Аналогично,

$$Z^* = \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \vee x_1 \wedge \overline{x_2} \wedge \overline{x_3} \vee x_1 \wedge x_2 \wedge \overline{x_3} \vee x_1 \wedge x_2 \wedge x_3$$

После эквивалентных преобразований находим

$$Z^* = (\overline{x_2} \wedge \overline{x_3}) \vee (x_1 \wedge x_2)$$

Из табл. 1.1 видно, что все значения Z и Z^* от одних и тех же наборов значений переменных совпадают. Однако Z^* образуется только двумя

"слагаемыми" Z . Конъюнкция $x_1 \wedge \overline{x_3}$ оказалась "лишней", не влияющей на результат. Это говорит о том, что формирование аналитического вида логической функции по ее табличному заданию, с помощью СДНФ, позволяет получить простейшее (лаконичное) представление, без лишних конструкций, не влияющих на результаты вычислений.

В заключение этого раздела представим обобщение, построенное над СДНФ, - в соответствии с теоремой разложения, широко используемой при конструировании электронных схем на основе стандартного набора элементов. Как и ранее, продемонстрируем суть данной теоремы на примере четырех переменных:

$$f = f(x_1 \wedge x_2 \wedge 0 \wedge 0) \wedge \overline{x_3} \wedge \overline{x_4} \vee f(x_1 \wedge x_2 \wedge 0 \wedge 1) \wedge \overline{x_3} \wedge x_4 \vee f(x_1 \wedge x_2 \wedge 1 \wedge 0) \wedge x_3 \wedge \overline{x_4} \vee f(x_1 \wedge x_2 \wedge 1 \wedge 1) \wedge x_3 \wedge x_4.$$

1.3. Исчерпывающее множество событий

Следующие ниже определения не могут не затронуть смысловых особенностей высказываний о событиях. Кроме чисто формальных свойств высказываний, выражающихся в их истинности или ложности, невозможно полностью абстрагироваться от содержательной сути или от контекста, в котором они звучат.

Определение 4. Исчерпывающее множество событий (ИМС) образуют те события, совокупность высказываний, о которых покрывает весь возможный смысловой диапазон проявления объекта высказывания, и каждая допустимая ситуация характеризуется тем, что значение ИСТИНА (1) может принимать единственное высказывание из этой совокупности. (Значение 0 могут принимать все высказывания.)

Рассмотрим примеры.

1. В состав редколлегии входят трое: Иванов, Петров, Сидоров. Тогда провозглашение фамилий этих фигурантов определяет исчерпывающее множество событий при выдвижении единственного представителя коллектива в президиум собрания.
2. Наказуемое превышение скорости автомобиля делится на

диапазоны: до 10%, от 10% до 20%, свыше 20%. Однако если в регламентирующем документе заданы только диапазоны до 10% и от 10% до 100%, то это не будет соответствовать исчерпывающему множеству событий. Такие нестрогие определения возможного диапазона ситуаций являются причиной юридической казуистики, требующей дальнейшего исследования прецедента.

Итак, ИМС, которому соответствует множество высказываний $A = \{x_1, \dots, x_n\}$, характеризуется тем, что при соответствующих обстоятельствах одно и только одно высказывание из этого множества может принимать значение 1. Это и определяется операцией ИСКЛЮЧАЮЩЕЕ ИЛИ, которую будем обозначать $\dot{\vee}$.

Очевидны главные свойства высказываний о событиях из ИМС:

$$\overline{x_i} = x_1 \dot{\vee} \dots \dot{\vee} x_{i-1} \dot{\vee} x_{i+1} \dot{\vee} \dots \dot{\vee} x_n \quad (1.12)$$

$$x_i \cdot x_j = \begin{cases} 0, & \text{при } i \neq j \\ 1, & \text{при } i = j \end{cases} \quad (1.13)$$

Теорема. Логическая функция от переменных-высказываний о событиях, образующих исчерпывающее множество событий, преобразуется в дизъюнкцию ИСКЛЮЧАЮЩЕЕ ИЛИ переменных-высказываний о событиях из этого множества.

Доказательство. Поясним доказательство теоремы анализом примера. На множестве высказываний $\{x_1, x_2, x_3\}$, СДНФ имеет вид

$$\begin{aligned} f(x_1, x_2, x_3) = & f(0, 0, 0) \wedge \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \vee \\ & \vee f(0, 0, 1) \wedge \overline{x_1} \wedge \overline{x_2} \wedge x_3 \vee f(0, 1, 0) \wedge \overline{x_1} \wedge x_2 \wedge \overline{x_3} \vee \\ & \vee f(0, 1, 1) \wedge \overline{x_1} \wedge x_2 \wedge x_3 \vee f(1, 0, 0) \wedge x_1 \wedge \overline{x_2} \wedge \overline{x_3} \vee \\ & \vee f(1, 0, 1) \wedge x_1 \wedge \overline{x_2} \wedge x_3 \vee f(1, 1, 0) \wedge x_1 \wedge x_2 \wedge \overline{x_3} \vee \\ & \vee f(1, 1, 1) \wedge x_1 \wedge x_2 \wedge x_3 \end{aligned}$$

Применив (1.12) и воспользовавшись свойством дистрибутивности ("раскрыв скобки") в соответствии с (1.3), обнаружим, что каждая вновь полученная конъюнкция может принимать значение некоторой

переменной, согласно (1.13), тогда и только тогда, когда эта переменная входит в нее ровно 3 раза (п раз). Это возможно лишь в тех конъюнкциях исходной СДНФ, где переменная, а не ее отрицание, входит при условии, что другие переменные входят со знаком отрицания. На такую ситуацию указывают те значения \bar{f} , которые отмечены единственной единицей в составе переменных. Для приведенного примера СДНФ принимает вид:

$$f(x_1, x_2, x_3) = f(0, 0, 1) \wedge x_3 \vee f(0, 1, 0) \wedge x_2 \vee f(1, 0, 0) \wedge x_1$$

Теорема доказана.

Чтобы подчеркнуть, что задание ситуаций подчиняется условию операции \vee , используем обозначение этой операции для получения окончательного вида СДНФ логической функции, заданной на ИМС:

$$f(x_1, \dots, x_n) = f(1, 0, \dots, 0, 0) \wedge x_1 \vee f(0, 1, \dots, 0, 0) \wedge x_2 \vee \dots \vee f(0, 0, \dots, 1, 0) \wedge x_{n-1} \vee f(0, 0, \dots, 0, 1) \wedge x_n$$

Отметим важные свойства выражения (1.14).

1. Каждая переменная, участвующая в формировании этого выражения, входит в него единственный раз.
2. Единственность вхождения переменных достигнута на основе применения закона дистрибутивности с учетом свойств высказываний на исчерпывающем множестве событий.

Назовем преобразование логической функции, приведшее к единственности вхождения переменных в каждую образующую его конъюнкцию, дистрибутивным.

В чем еще смысл Теоремы 1?

Представим себе ход рассуждения следователя, сокращающего круг подозреваемых. Первоначально он установил, что преступником мог быть либо Иванов, либо Петров, либо Сидоров, первоначально составляющие ИМС. Однако, исследовав некоторую логическую функцию \bar{f} алиби или участия в преступлении, он установил, что $\bar{f}(\text{Иванов} = 1, \text{Петров} = 0, \text{Сидоров} = 0) = 0$, в то

время как $f(\text{Иванов} = 0, \text{Петров} = 1, \text{Сидоров} = 0) = 1$,
 $f(\text{Иванов} = 0, \text{Петров} = 0, \text{Сидоров} = 1) = 1$. Таким образом, $f(\text{Иванов}, \text{Петров}, \text{Сидоров}) = \text{Петров} \dot{\vee} \text{Сидоров}$. Это резко сужает круг подозреваемых, т.к. скорректированным исчерпывающим множеством событий является $\{\text{Петров}, \text{Сидоров}\}$.

1.4. Композиция исчерпывающих множеств событий. Дерево логических возможностей. Факторное пространство событий

Для строгого логического мышления, исключающего неопределенность, приходится оперировать не отдельными событиями и даже не исчерпывающими множествами таких событий (высказываниями о них), а композициями таких множеств. Между событиями, принадлежащими различным множествам, возможна зависимость, порождающая сложные высказывания. Да и сами ИМС могут определяться и инициироваться обстоятельствами, обусловленными событиями из других ИМС. Связи между ИМС, образующие сложные высказывания, отображаются деревом логических возможностей.

Рассмотрим пример.

Пансионат для ветеранов труда обеспечивает постояльцам активный отдых круглый год. Представим схемой (рис. 1.1) распорядок дня отдыхающих. Такая схема и определит дерево логических возможностей.

Уровни ветвления могут формироваться разными способами. Например, первый уровень можно сформировать на основе времен года и т.д. Однако в порядке рекомендации можно следовать правилу "события располагаются на более низких уровнях по сравнению с теми уровнями, которые занимают события, от которых зависят данные события".

Бабушка пишет внуку: "Зимой я после завтрака катаюсь на лошади, и летом я после завтрака катаюсь на лошади, а также весной после завтрака прогулка бывает на лошади". ... Что-то ей не нравится, и она

строит схему своего составного высказывания:

$$f = x_1 \wedge x_7 \wedge x_{14} \vee x_1 \wedge x_5 \wedge x_{14} \vee x_1 \wedge x_4 \wedge x_{10} \wedge x_{14}.$$

Несколько поразмыслив, бабушка использует вынесение за скобку:

$f = x_1 \wedge x_{14} \wedge (x_5 \vee x_7 \vee x_4 \wedge x_{10})$. Тогда окончательный текст сообщения принимает вид: "После завтрака я катаюсь на лошади летом или зимой, а также, бывает, и весной, - вместо прогулки". Как же бабушка определила форму того логического выражения - функции, отображающей все возможные варианты, и даже пути, ведущие к свершению интересующего события?

Ответ следующий: необходимо на каждом пути в дереве логических возможностей, ведущем к заданному событию, построить конъюнкцию событий, образующих этот путь. Затем все такие конъюнкции объединить операцией дизъюнкции. Поскольку используются только исчерпывающие множества событий, очевидно, что эта дизъюнкция выполняется с помощью операции \vee , т.е. ИСКЛЮЧАЮЩЕЕ ИЛИ (хотя можно пользоваться значком \vee , опираясь на действительный, "физический" смысл возможных событий).

Полученная таким способом функция подвергается дистрибутивному преобразованию - "вынесению за скобки".

Отметим, что в результате такого способа построения искомая функция принимает вид, при котором каждая используемая переменная-высказывание входит не более одного раза.

Например, функция, отображающая такое событие в жизни бабушки, как езда на велосипеде, имеет вид

$$g = x_1 \wedge x_4 \wedge x_{10} \wedge x_{13} \dot{\vee} x_1 \wedge x_5 \wedge x_{13} = x_1 \wedge x_{13} \wedge (x_4 \wedge x_{10} \dot{\vee} x_5).$$

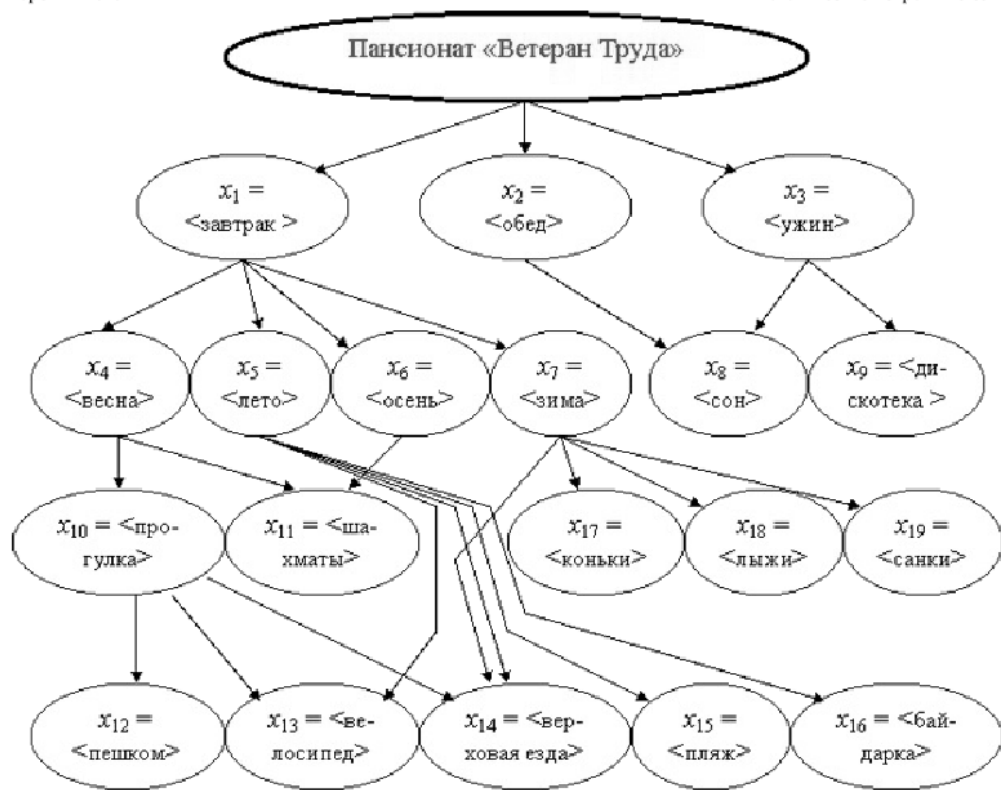


Рис. 1.1. Полное дерево логических возможностей

Однако далее будет показано, что не всегда единственного вхождения переменных можно добиться с помощью дистрибутивных преобразований. Иногда требуются дополнительные действия для его осуществления.

Определение 5. Совокупность всех исследуемых в данном контексте событий, т.е. множество - объединение всех рассматриваемых ИМС - образует факторное пространство событий .

Как и ранее, точку факторного пространства (ситуацию) будем обозначать $\{x_1, \dots, x_n\}$.

Итак, показана возможность построения логических функций на основе высказываний о событиях из факторного пространства.

Как видно из примера, факторное пространство событий отображается

ветвящейся структурой на основе отдельных исчерпывающих множеств событий, входящих в его состав. Тогда подмножества, состоящие из таких ИМС, тоже являются факторными подпространствами, которые в некотором контексте можно исследовать отдельно.

Например, можно отдельно исследовать факторное подпространство, сформированное на основе первых двух уровней ветвления (рис. 1.2) в приведенном на рис. 1.1 дереве логических возможностей. Это может быть необходимо при планировании финансовых расходов пансионата на питание.

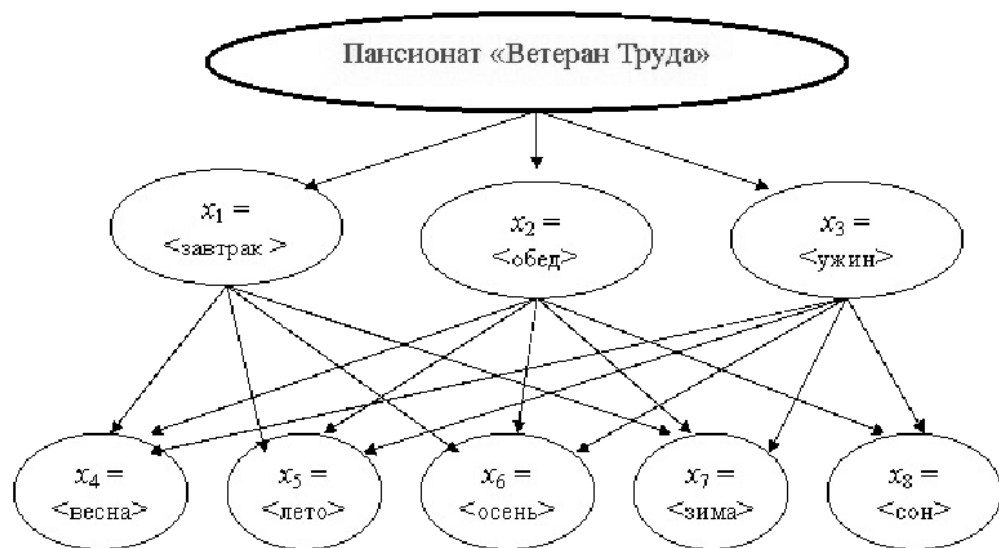


Рис. 1.2. Факторное подпространство для исследований финансовых затрат на питание

Можно, в соответствии с поставленной задачей (в контексте исследований), формировать другие факторные пространства событий. Например, планирование использования спортивного инвентаря по времени года приводит к целесообразности факторного пространства, структура которого показана на рис. 1.3.

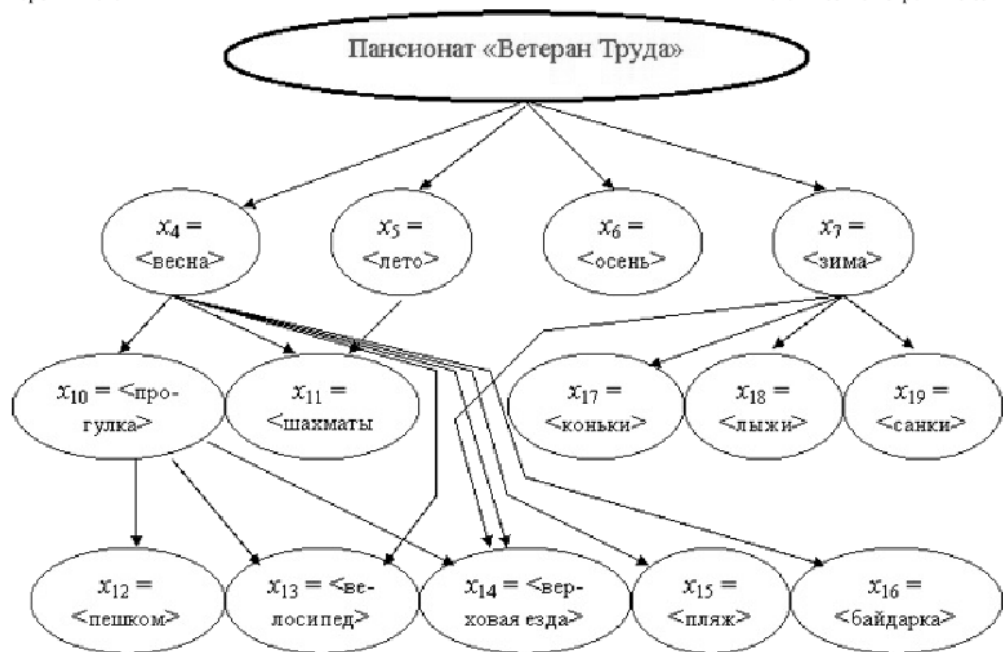


Рис. 1.3. Факторное пространство для планирования использования спортивного инвентаря

1.5. Система принятия решений

Для некоторой логической функции f от переменных из факторного пространства событий воспользуемся операцией следования (импликации) и сформируем логическое выражение вида

$$f(x_1, \dots, x_n) \rightarrow R \quad (1.15)$$

Здесь f следует рассматривать как выражение, определяющее условие, сложившуюся ситуацию, посылку, а R - высказывание, которое рассматривается как следствие: правило поведения, значение векторной функции, указание к действию и т.д. Таким образом, возможно формирование связей вида "посылка - следствие", "если ..., то ...". При этом функция f задается на множестве ситуаций и указывает на то, что, если на некоторой ситуации она принимает значение 1 (ИСТИНА), то такое же значение принимает высказывание R , являясь руководством к действию, к принятию определенного решения.

Подобно (1.16), можно описать множество логических выражений, определяющих стройную систему управления или принятия решений в соответствии со складывающейся ситуацией в факторном пространстве событий:

$$\begin{aligned} f_1(x_1, \dots, x_n) &\rightarrow R_1 \\ \dots\dots\dots\dots\dots\dots & \\ f_m(x_1, \dots, x_n) &\rightarrow R_m \end{aligned} \quad (1.16)$$

Определение 6. Система логических выражений вида (1.16), заданная на факторном пространстве (подпространстве) событий, обладающая полнотой и непротиворечивостью, называется системой принятия решений.

Поясним важность свойств, указанных в определении.

То, что система функций f_1, \dots, f_m является полной, означает, что любая точка факторного пространства событий входит в область задания хотя бы одной из этих функций. Непротиворечивость означает, что по каждой ситуации одна и только одна из этих функций принимает значение 1, приводящее к истинности соответствующего высказывания - решения. Однако отметим, что в действительности на основе смыслового содержания задачи по каждой или некоторой ситуации может быть известно более одного правильного решения, приводящего к успешным действиям. В таком случае высказывания об этих решениях могут быть объединены операцией ИЛИ, что приводит к приведенному выше предположению о непротиворечивости.

Продолжим рассмотрение примера.

Пусть известная нам бабушка планирует занятия физкультурой и спортом во все времена года по времени дня: после завтрака, после обеда и после ужина. Объединяя высказывания по принципу "если . . . , то" и пользуясь обозначениями на рис. 1.1, она формирует систему принятия решений, которой, не полагаясь на память, намерена строго следовать, добившись согласия администрации.

Система имеет вид

- | | | | | |
|----|--|---|-------------------------|-----|
| 1. | $x_1 \wedge x_4 \rightarrow R_1$ | = | Прогулка на велосипеде; | |
| 2. | $x_1 \wedge x_6 \vee x_2 \wedge x_4 \rightarrow R_2$ | = | Шахматы; | |
| 3. | $x_2 \wedge x_5 \vee x_1 \wedge x_7 \rightarrow R_3$ | = | Верховая езда; | |
| 4. | $x_1 \wedge x_5 \vee x_2 \wedge x_6 \rightarrow R_4$ | = | Байдарка; | (1. |
| 5. | $x_3 \wedge (x_4 \vee x_6) \rightarrow R_5$ | = | Дискотека; | |
| 6. | $x_2 \wedge x_7 \rightarrow R_6$ | = | Пешая прогулка; | |
| 7. | $x_3 \wedge (x_5 \vee x_7) \rightarrow R_6$ | = | Пешая прогулка. | |

Планируя пешую прогулку, бабушка первоначально получила следующее выражение:

$$x_2 \wedge x_7 \vee x_3 \wedge x_5 \vee x_3 \wedge x_7 \rightarrow R_6 = \text{Пешая прогулка}$$

Однако выше не напрасно обращается внимание на целесообразность однократного вхождения переменных в подобное выражение (это будет изучено в [лекциях 9 и 10](#)). Выражение, полученное первоначально, с помощью дистрибутивных преобразований привести к такому виду не удастся. Тогда бабушка решает разбить это выражение на два подобных, сформировав получение одного и того же решения на основе двух условий. Это и послужило появлению в (1.17) двух выражений, определяющих одно решение R_6 .

Легко убедиться, что все возможные ситуации факторного пространства событий охвачены, демонстрируя полную ясность действий бабушки.

Системы принятия решений могут образовывать сложные иерархические структуры. В этом случае необходимо, чтобы высказывания-решения R_1, \dots, R_m отображали события, образующие ИМС.

1.6. "Схемотехническое" представление системы принятия решений

Отобразим (с нарушением некоторых стандартов) схемотехнически бабушкину СПР, подобно электронной схеме ([рис. 1.4](#)) с помощью конъюнкторов и дизъюнкторов. На вход будем подавать значения истинности переменных-высказываний (ситуации) так, чтобы на одном

из выходов формировалась единица - значение истинности соответствующего решения. Задавать значение ситуаций следует корректно, чтобы соблюдать требования вхождения переменных в исчерпывающие множества событий.

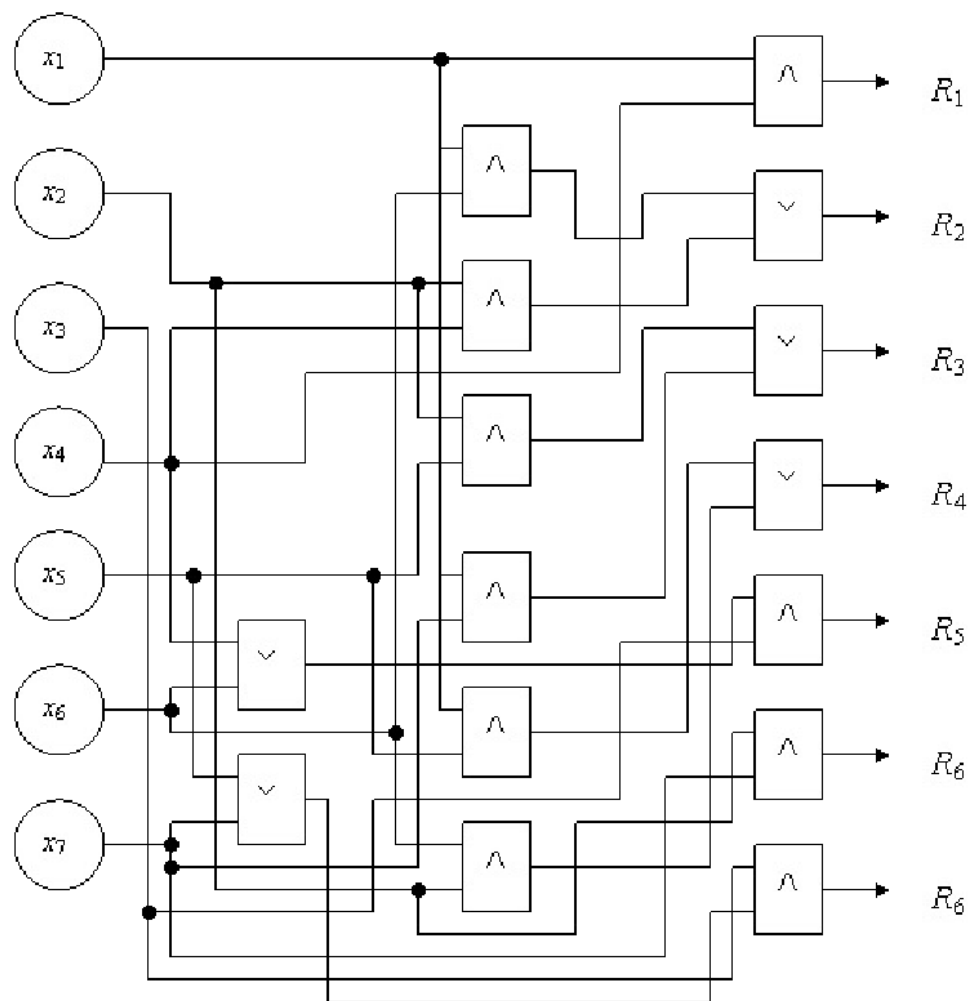


Рис. 1.4. "Электронная" схема системы принятия решений

Реализовав эту схему на логических элементах, бабушка получит реальное средство подсказки: что она должна делать в данное время года и суток.

Например, бабушка хочет вспомнить, чем она должна заниматься летом

после обеда. Она полагает $x_2 = x_5 = 1$ при нулевых значениях других переменных и запускает программу, моделирующую работу электронной схемы. На выходе R_3 формируется сигнал, соответствующий высказыванию "Верховая езда".

Электронная схема, имитирующая систему принятия решений, требует корректного задания исходной информации, т.е. корректной формулировки вопроса в соответствии со смыслом задачи. Одновременно должна существовать уверенность в получении единственного решения, в том числе предполагающего задание альтернативных вариантов, если это предусмотрено при построении СПР. В таком случае окончательный выбор альтернативы может быть предусмотрен при развитии системы, при анализе новых вводимых факторов. При некорректной формулировке вопроса возникнут коллизии, неоднозначные или неправильные выводы. Таким образом, СПР может отвечать только на те вопросы, на которые она настроена.

1.7. Достоверность высказываний о событиях

Говоря о высказываниях как о логических переменных, мы, несомненно, предполагаем наличие субъективного фактора: ведь это кто-то сказал. И мы говорим верному другу: "Я мало доверяю этому человеку, но он сказал . . . , а дыма без огня не бывает".

Очевидно, недостаточно резких, кардинальных, взаимоисключающих, крайних суждений о высказываниях, подразделяющих их на истинные и ложные. Жизненный опыт говорит, что стопроцентной правды не бывает.

Таким образом, необходимо ввести понятие достоверность высказывания, которая идеально представляет вероятность того, что данное высказывание о свершении события истинно, т.е. представляющая его логическая переменная равна 1.

В этой оценке достоверности вновь практически преобладает субъективный фактор. Поэтому при построении экспертных систем применяется двойная оценка: оценка, данная экспертом по запросу, и вес самого эксперта. Здесь эффективно используется аппарат нечетких

множеств [30].

Говоря о сложных высказываниях, отображаемых деревьями логических возможностей, тем более трудно судить о достоверности высказываний о событиях, особенно тех, которые отображаются концевыми вершинами.

Рассмотрим пример. Информатор сообщает Агенту о том, что видел своими глазами, как Марина передала Васе пачку денег. Призвав на помощь свой богатый опыт, Агент рассуждает логически:

1. Насколько можно доверять Информатору, который три дня не брился и от которого дурно пахнет?
2. Мог ли находиться Информатор в это время в нужном месте, чтобы "видеть своими глазами"?
3. Насколько верно то, что в переданной пачке были деньги?
4. Можно ли оперативно воспользоваться существующими математическими методами оценки (например, аппаратом нечетких множеств), заодно учитывая другие факторы, как, например, личные финансовые трудности, или необходимые оценки следует выполнить интуитивно, "с потолка"?
5. Так какова же должна быть формулировка отчета в Центр для получения максимального вознаграждения за поимку взяточника?

Таким образом, видно, что оценка достоверности высказываний неизбежна, но представляет значительные практические трудности, тем более - при требуемой оперативности этих оценок, столь важной в реальных системах управления и принятия решений.

Итак, на основе алгебры высказываний можно создавать электронные системы принятия решений: на входе задавать ситуацию, на выходе получать указание на правильную реакцию. Все дело лишь в интерпретации предметной области - во вскрытии причинно-следственных связей, в исследовании личного или коллективного опыта, в изучении теории. Необходимы и схемотехнические навыки запоминания связей.

Однако смущают два обстоятельства:

1. Неопределенность исходной информации о ситуациях, исключая точный ответ на вопрос о наличии или отсутствии события и делающая неправомерным использование исключительно булевых переменных. Высказывания не бывают истинными и ложными, как это предполагается в классической математической логике. Высказывания оцениваются своей достоверностью, которая принимает действительные значения на отрезке $[0, 1]$ и подчиняется известным положениям теории вероятности.
2. Способность человека логично мыслить на неформальном уровне не реализуется с помощью конъюнкторов и дизъюнкторов в составе мозга. Именно вскрытие механизмов мышления, особенно того, которое мы называем рефлекторным, привлекает внимание исследователей. Необходимо искать механизмы мышления, оперирующие не с булевыми переменными, а с действительными, несущими смысл достоверности.

Пусть рассмотренные выше переменные-высказывания $\{x_i\}$, образующие факторное пространство, могут принимать значения достоверности $\{P_i\}$, $0 \leq P_i \leq 1, i = 1, \dots, M$.

Так как факторное пространство формируется на основе исчерпывающих множеств событий, то внутри каждого такого множества выполняется известное правило нормировки: сумма достоверностей событий каждого такого множества равна единице.

Перегруппируем события и дополним дерево логических возможностей, представленное на [рис. 1.1](#), указав на его ветвях (стрелках), в качестве весов этих ветвей, значения достоверности событий ([рис. 1.5](#)). Получим вероятностное дерево логических возможностей.

При организации ветвления в этом дереве также предполагаются все возможные альтернативы, т.е. исчерпывающие множества событий. Поэтому сумма вероятностей всех событий, отображаемых вершинами, которые связаны входящими стрелками, исходящими из некоторой вершины, равна единице.

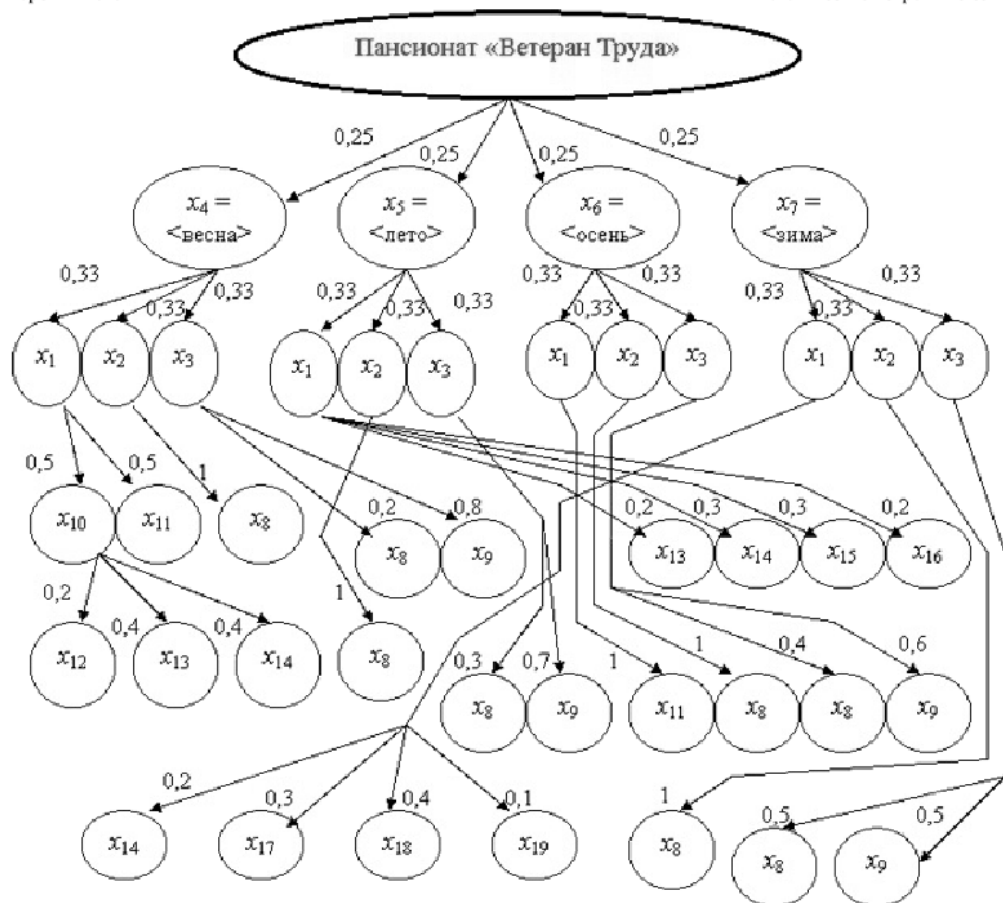


Рис. 1.5. Вероятностное дерево логических возможностей

В отличие от дерева логических возможностей, вероятностное дерево явно отображает зависимость событий. События, зависящие от данного, отображаются более низкими уровнями ветвления. Такая зависимость определяется на уровне смыслового анализа факторного пространства.

Например, логично предположить, что формы труда, отдыха и спортивных развлечений зависят от времени года, затем - от распорядка приема пищи. Такая зависимость и отображается на рис. 1.5.

Тогда достоверность событий формируется с помощью условных вероятностей, зависящих от путей, по которым достигаются эти события. Поэтому на вероятностном дереве логических возможностей

целесообразно повторять вершины одного смыслового содержания, в результате чего размножаются варианты ветвления, а дерево существенно разрастается. От совмещения путей страдает наглядность, и не более того.

Чтобы найти вероятность некоторого события b при условии свершения события a (событию a может соответствовать корневая вершина, тогда речь идет о полной, а не условной вероятности), необходимо найти все пути, ведущие из a в b . По каждому пути необходимо перемножить все веса ветвей. Полученные по всем путям произведения необходимо сложить.

Пример 1. Найдем вероятность того, что отдыхающий весной и летом в произвольно выбранный момент времени совершает прогулку верхом:

$$0,25 \times 0,33 \times 0,5 \times 0,4 + 0,25 \times 0,33 \times 0,3 = 0,04125.$$

Пример 2. Найдем вероятность того, что в произвольно выбранный момент времени в течение года отдыхающий совершает прогулку верхом:

$$0,25 \times 0,33 \times 0,5 \times 0,4 + 0,25 \times 0,33 \times 0,3 + 0,25 \times 0,33 \times 0,2 = 0,05775.$$

1.8. Системы принятия решений на основе достоверности высказываний о событиях

Изменения, внесенные в дерево логических возможностей и представленные на рис. 1.5, отобразим в электронной схеме системы принятия решений, представленной на рис. 1.4. Теперь исходными данными (рис. 1.6) для конъюнкторов и дизъюнкторов, принимающих эти данные, становятся не булевы, а действительные значения, для которых логические операции не определены.

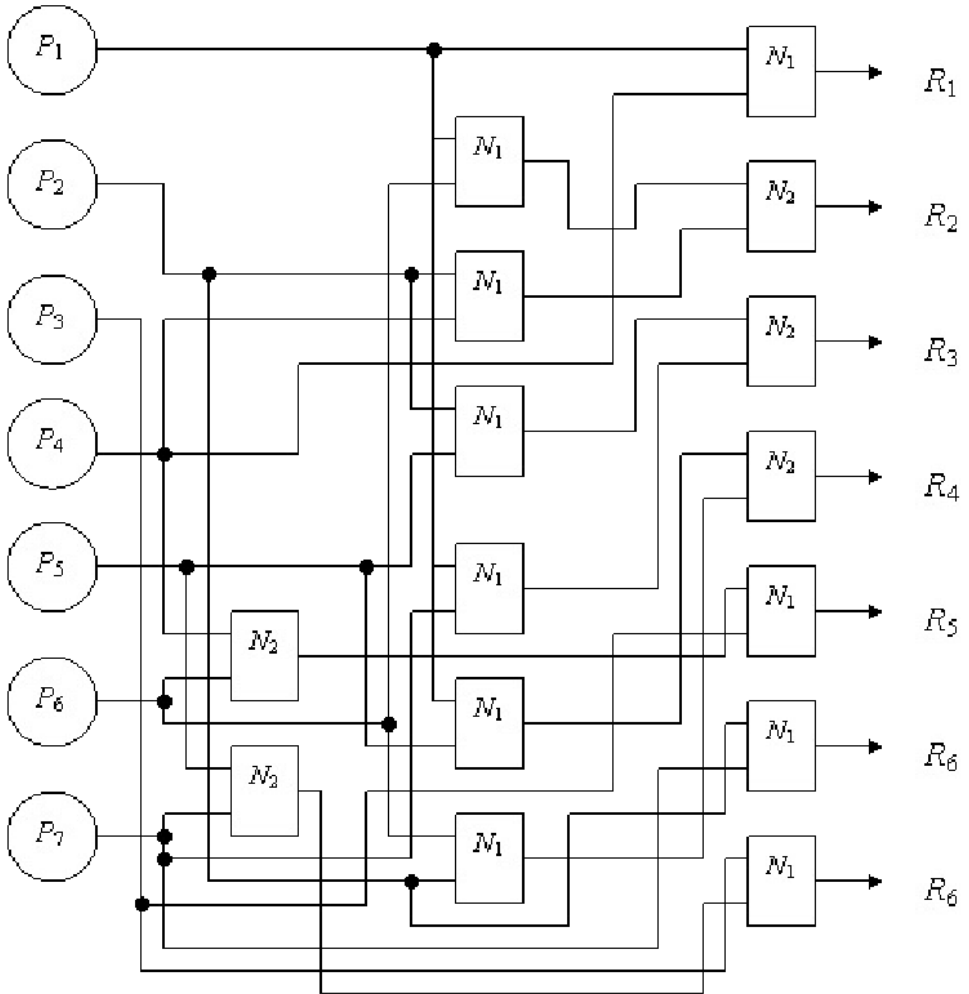


Рис. 1.6. Система принятия решений на основе достоверности событий

Следуя далее по пути приблизительных оценок (ибо практически достоверность, как категория теории вероятностей, принадлежит области весьма приблизительных оценок), разработаем некоторый суррогат операций конъюнкции N_1 и дизъюнкции N_2 на основе передаточной функции порогового элемента, преобразующего сумму входных величин в выходные значения, которые приближенно "напоминают" результаты упомянутых логических операций. Этот путь - путь ухода от точного выбора решения в сторону выбора решения на основе степени похожести ситуаций на уже известные, - путь ассоциативного мышления.

Существует множество вариантов подбора пороговой передаточной функции, лежащей в основе такого элемента.

Введем сквозную нумерацию всех узлов схемы, реализующих дизъюнкцию и конъюнкцию. Пусть i - номер такого узла, j - номер входа этого узла при количестве m_i активных входов (в данном примере каждый узел имеет два входа), ω_j - вес входа. Тогда простейшая передаточная функция f_i , реализуемая i -м узлом для замены логических операций конъюнкции и дизъюнкции, имеет вид

$$f_i = \begin{cases} \sum_{j=1}^m f_j \omega_j, & \text{если } \sum_{j=1}^m f_j \omega_j \geq 0 \\ 0, & \text{в противном случае} \end{cases} \quad (1.18)$$

Здесь f_j - величина сигнала, поступающая на j -й вход.

Тогда элемент N_1 , подобный конъюнктору, может быть реализован при $\omega_j = 1/m_i$, $j = 1, \dots, m_i$, с помощью существенно высокого порога (рис. 1.7), где значение δ_i обусловлено некоторой поправкой, достаточной, чтобы для преодоления порога сигналы возбуждения с большой степенью уверенности поступали обязательно по всем входам.

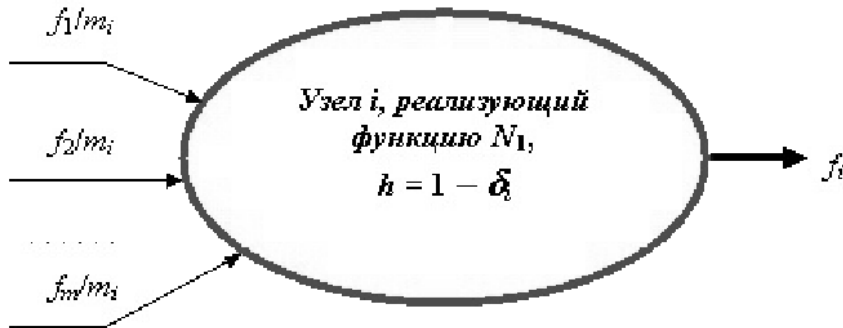


Рис. 1.7. Элемент N_1

На этапе настройки и верификации СПР предполагается, что входные сигналы - булевы переменные, принимающие значения 0, 1. Тогда целесообразно выбрать значение $\delta_i < 1/m_i$. Очевидно, что для того, чтобы преодолеть порог, на всех входах должны быть "1"; недостаток

хотя бы одной "1" приведет к тому, что сумма поступивших сигналов будет более чем на $1/m_{\perp}$ меньше указанной суммы весов.

При переходе к действительным переменным, когда вместо событий рассматриваются, например, лишь предполагаемые вероятности их наступления, экспериментальный выбор значения δ_i может обусловить ту границу, когда считается с возможностью данной комбинации событий нецелесообразно.

Элемент N_2 , подобный дизъюнктору, реализуется, наоборот, при низком значении порога и при $\omega_j = 1, j = 1, \dots, m_i$. Порог выбирается так, чтобы уже при возбуждении на одном входе возникал сигнал возбуждения на выходе. При этом сигнал на выходе не превышает "1" (рис. 1.8), а значение δ_i выбирается экспериментально достаточно небольшим.

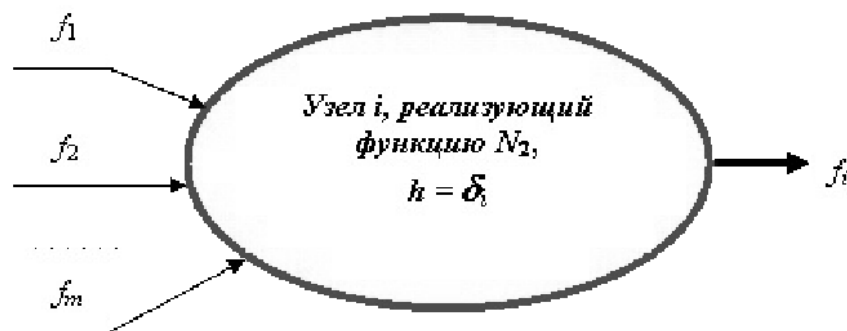


Рис. 1.8. Элемент N_2

Задав на входе СПР значения достоверности переменных-высказываний и рассчитав значения на выходах пороговых элементов, на выходах схемы получим некоторые значения. Максимальное из этих значений "голосует" в пользу соответствующего решения.

Предложения, касающиеся создания пороговых элементов N_1 и N_2 , носят лишь рекомендательный характер. Здесь неограниченный простор для творчества.

Напомним, что корректность задания исходной информации (соблюдение условия нормировки на исчерпывающих множествах

событий, оценки достоверности с помощью вероятностного дерева логических возможностей) гарантируют практически приемлемый результат. Если же на входах задавать что угодно, то СПР преобразует это в какую угодно рекомендацию по принципу "каков вопрос - таков и ответ".

На рассмотренном жизненном примере проанализируем принимаемые бабушкой решения на основе двух вариантов СПР: с помощью электронной схемы (рис. 1.4), использующей определенность знания о ситуации, и с помощью схемы, основанной на неопределенности, на предполагаемой достоверности этих знаний (рис. 1.6). Положим (на основе интуиции) $\delta_i = 0,3$ для всех i .

Данные сведены в табл. 1.2.

Таблица 1.2. Сравнительные оценки получаемых решений

Переменные							Решение по электронной схеме	Решение на основе достоверности событий
x_1 (P_1)	x_2 (P_2)	x_3 (P_3)	x_4 (P_4)	x_5 (P_5)	x_6 (P_6)	x_7 (P_7)		
1				1			R_1	R_1
	1			1		1	R_3	R_3
1		1					Нет решения	Нет решения
0,8	0,2			0,4	0,5		Решение не определено	R_2
0,2	0,4	0,6	1			0,5	Решение не определено	R_2

Более точный выбор значения δ_i производится на основе верификации СПР по известным вариантам нахождения решения. В данном случае представляется, что этот выбор произведен успешно.

1.9. Минимизация длины логической цепочки в системе принятия решений

Предваряя подробное исследование в последующих лекциях, отметим,

что замена логических операций операцией суммирования при счете передаточной функции приводит к актуальности однократного учета всех входящих переменных, т.е. к единственности вхождения переменных в каждое логическое выражение, составляющее описание системы принятия решений. Выше с целью обеспечения такого единственного вхождения переменной был использован прием "размножения" решения R_6 .

При разработке электронных схем исследуется понятие "длина логической цепочки" - под ней подразумевается максимальное количество электронных элементов, которое должен преодолеть сигнал на входе схемы, пройдя последовательное тактируемое преобразование, чтобы на выходе схемы сформировался сигнал. От этой длины, определяющей время переходного процесса, зависит быстрдействие схемы. Поэтому актуальной задачей является минимизация максимальной длины логической цепочки при возможности параллельного выполнения всех таких цепочек (что характерно для прохождения электрического сигнала по схеме).

Очевидно, что в схеме на рис. 1.4 максимальная длина логической цепочки равна двум.

Применим ко всем выражениям (1.17), каждое из которых является или может быть преобразовано в дизъюнкцию конъюнкций, прием "размножения" решений. Теперь (рис. 1.9) схема состоит из цепочек единичной длины. Каждый входной сигнал подвергается обработке только конъюнктом. Так как электронная схема полностью определяет конструкцию системы принятия решений на основе достоверности событий, то можно преобразовать полученную электронную схему в однослойную схему СПР, показанную на том же рисунке.

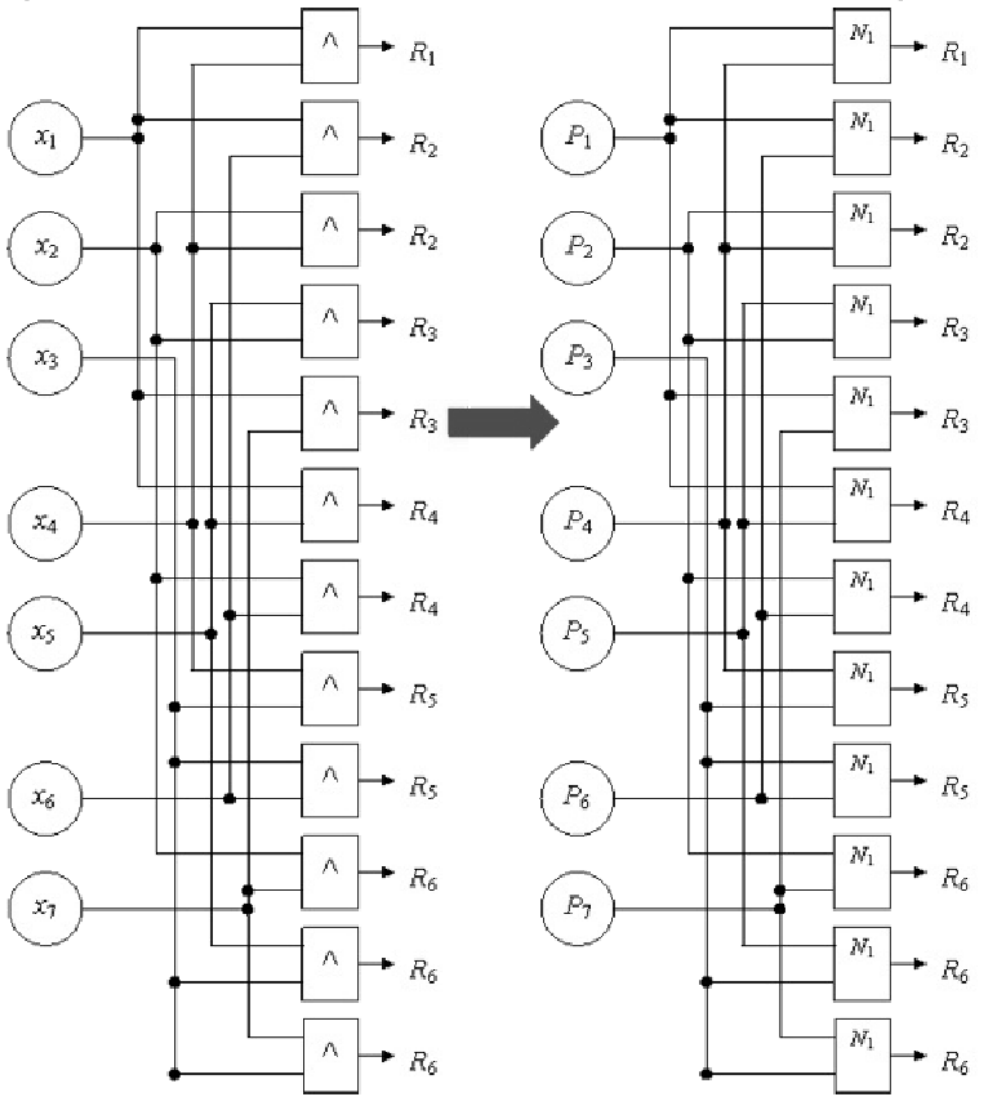


Рис. 1.9. Преобразование электронной схемы с единичной длиной логической цепочки в однослойную систему принятия решений

Таким образом, доказано следующее утверждение:

Лемма 1. Любая СПР, сформированная на основе логического описания булевыми функциями, способом "размножения" решений преобразуется в однослойную СПР на основе достоверности событий.

Преимуществом таких СПР является то, что они представляют собой

таблицы с ассоциативной выборкой по принципу наибольшей схожести.

Конечно, можно за каждым решением закреплять один выход СПР, на котором объединить общее решение, полученное по разным путям, в виде текста. При корректно заданных исходных данных - на основе правил использования исчерпывающих множеств событий - СПР будет "работать" правильно, выдавая адекватные ответы. Тогда рекомендация "Прими решение R" будет выдана, а информация о пути, приведшему к этому решению, будет утрачена.

При составлении "электронной" схемы такое объединение производится с помощью операции дизъюнкции, что приводит к длине логической цепочки, равной двум. Но ведь если, формируя структуру СПР, строго следовать порядку построения "электронная схема \rightarrow система принятия решений", то и СПР будет иметь максимальную логическую цепочку с длиной, равной 2.

Таким образом, "размножение" решений - операция, свойственная СПР, что делает ее (СПР) построение отличающимся, развивающим "схемотехнический" подход. "Электронные" схемы целесообразно использовать на начальном этапе исследования логического описания СПР, а далее, оттолкнувшись от них, перейти к более совершенной однослойной структуре.

"Размножение" решений имеет важное достоинство. Оно позволяет установить причину, найти объяснение принимаемого решения. Это означает, что текст решения может быть дополнен указанием причины принятия именно такого решения.

Например, получив информацию о необходимости заказа велосипеда в отделе спортивного инвентаря, бабушка может воспользоваться и важным объяснением: "...потому что сейчас, скорее всего, весна, а вы, вероятно, только что сытно позавтракали" (рис. 1.10).

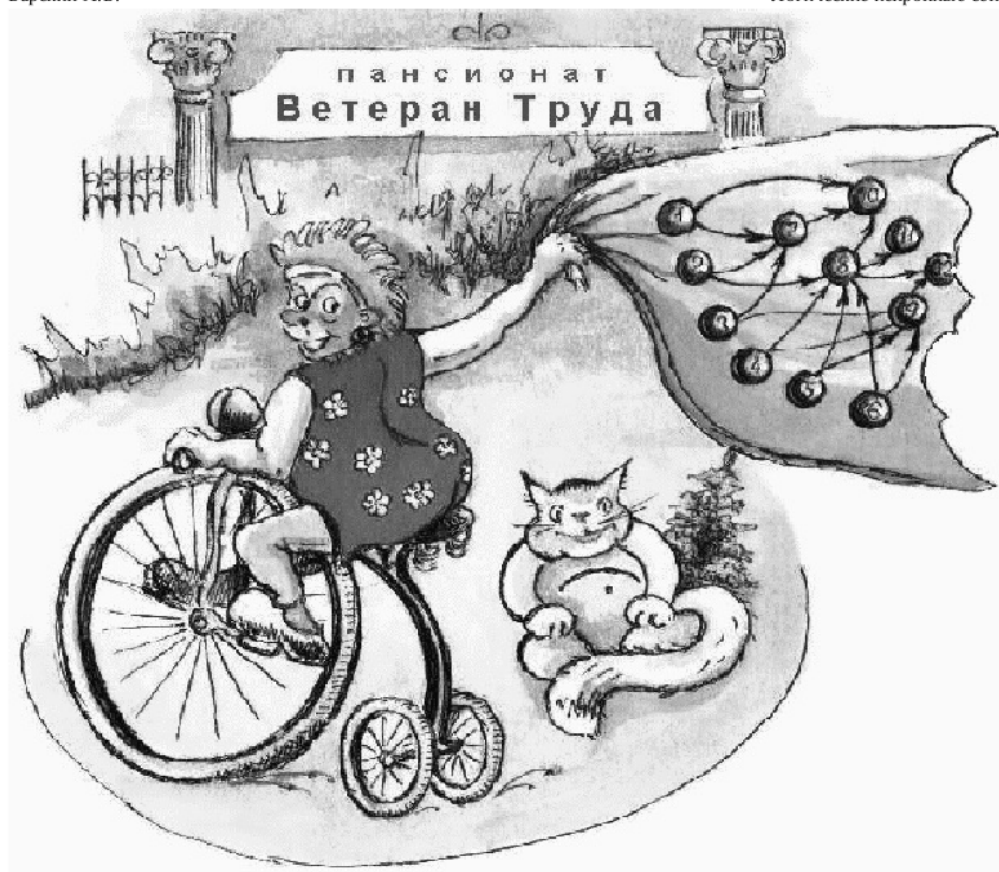


Рис. 1.10. Бабушка

В заключение данной лекции следует отметить, что построен алгоритм параллельных вычислений [6] сложных логических конструкций в области действительных переменных, предназначенный для реализации высокого быстродействия в системах управления и принятия решений. Более того, сведение СПР к однослойной приводит к применению лишь тех передаточных функций (элементов N_1 на рис. 1.6 и рис. 1.9), которые имитируют конъюнкторы. Это служит повышению достоверности оценок, стандартизации и адекватности природным процессам.

Основы нейросетевых технологий

Приводятся основные понятия нейронных сетей. Демонстрируется возможность их построения на основе логической схемы взаимодействия событий принимаемых решений. Обсуждаются режимы работы, проблемы обучения, интерпретации и ввода исходной информации, представления пространства признаков, помехозащищенности и локализации максимального возбуждения.

"...письмоводитель градоначальника, вошедши утром с докладом в его кабинет, увидел такое зрелище: градоначальниково тело, обложенное в вицмундир, сидело за письменным столом, а перед ним, на кипе недоимочных реестров, лежала, в виде щегольского пресс-папье, совершенно пустая градоначальникова голова".

М.Е. Салтыков-Шедрин. "История одного города"

2.1. Нейронная сеть - средство распознавания

2.1.1. Построение обученной нейросети

Пусть перед нами экран, разбитый на двенадцать клеток, 4x3. Клетки отображают дискретность элементов изображения. При фокусировании изображения клетка либо засвечивается, либо нет. "Засветка" определяет единичное значение величины ее возбуждения, "не засветка" - нулевое. Так, буква О определяет засветку клеток, определяемую на [рис. 2.1](#). Буква А засвечивает экран, как показано на [рис. 2.2](#).

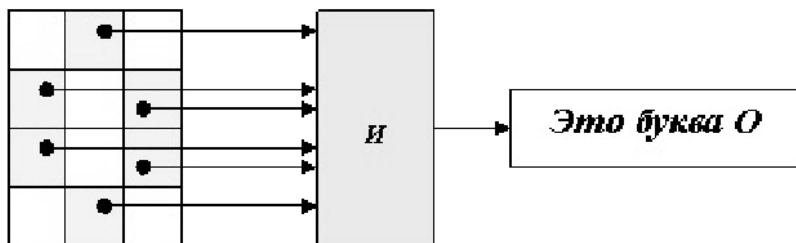


Рис. 2.1. Обучение букве "О"

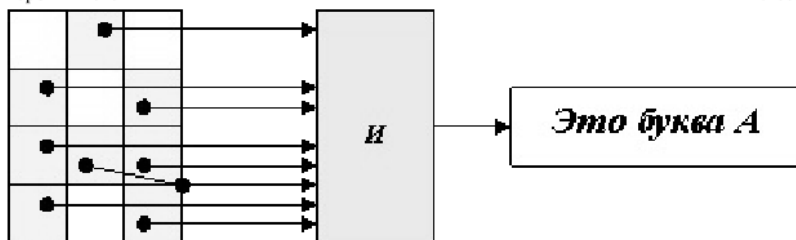


Рис. 2.2. Обучение букве "А"

Что надо сделать для *распознавания* этих двух букв? Для того, чтобы некий конструируемый нами прибор мог сказать, какая это буква?

Очевидно, надо все сигналы возбуждения клеток экрана, засвечиваемой буквой О, подать на *конъюнктор*, реализующий схему И. Единичный сигнал на выходе *конъюнктора*, как показано на рис. 2.1, сформируется тогда и только тогда, когда засветятся все клетки экрана, на которое ложится изображение буквы О. Наличие единичного сигнала на выходе *конъюнктора* и определит ответ "Это буква О".

То же необходимо сделать и для буквы А.

Пометим каждую клетку экрана ее координатами. Тогда на языке математической логики сделанное нами можно записать в виде логических высказываний - предикатов:

$$(1, 2) \wedge (2, 1) \wedge (2, 3) \wedge (3, 1) \wedge (3, 3) \wedge (4, 2) \rightarrow O$$

$$(1, 2) \wedge (2, 1) \wedge (2, 3) \wedge (3, 1) \wedge (3, 2) \wedge (3, 3) \wedge (4, 1) \wedge (4, 3) \rightarrow A$$

Эти предикаты определяют "электронное" воплощение методами схемотехники.

При этом буквы не будут "мешать" друг другу, так как засветка соответствующих им клеток экрана частично не совпадает и единичное значение конъюнкции определится только для одной из них.

А если на экран подать букву К? Тогда ни один из двух конъюнкторов не выработает единичное значение, так как не произойдет полное совпадение засветки соответствующих им клеток экрана. Чтобы

"научить" систему букве К, необходимо ввести еще один *конъюнктор* и прodelать те же построения, что и выше.

Таким образом, мы можем говорить, что построили систему *распознавания* двух "правильно" заданных букв.

Но что делать, если буквы на экране пишутся дрожащей рукой? Тогда мы должны разрешить альтернативную засветку каких-то соседних клеток экрана и учитывать это с помощью операции дизъюнкции, ИЛИ. Как известно, в результате выполнения этой операции формируется единичный сигнал в том случае, если на входе есть хоть один единичный сигнал.

Рассмотрим возможность *распознавания* буквы О, допустив возможность засветки клеток $(1, 1)$, $(1, 3)$, $(4, 1)$, $(4, 3)$. Тогда ранее построенный предикат примет вид

$$((1,1) \vee (1,2) \vee (1,3)) \wedge (2,1) \wedge (2,3) \wedge (3,1) \wedge (3,3) \wedge ((4,1) \vee (4,2) \vee (4,3)) \rightarrow O$$

Аналогично, для буквы А допустим засветку клеток $(1, 1)$ и $(1, 3)$:

$$((1,1) \vee (1,2) \vee (1,3)) \wedge (2,1) \wedge (2,3) \wedge (3,1) \wedge (3,2) \wedge (3,3) \wedge (4,1) \wedge (4,3) \rightarrow A$$

Объединив оба предиката, получим схему на рис. 2.3.

Таким образом, мы реализовали для *обучения* и *распознавания* "схемотехнический" подход, основанный на применении булевых функций и оперирующий булевыми переменными 0, 1.

Теперь совершим тот шаг, тот переход, который и определяет гениальную простоту природного воплощения, рассчитанного на неполноту данных, недостоверность, "зашумленность", высокое быстродействие, надежность и унифицированность. Ибо мы не можем представить себе электронную схему, укрытую в черепной коробке.

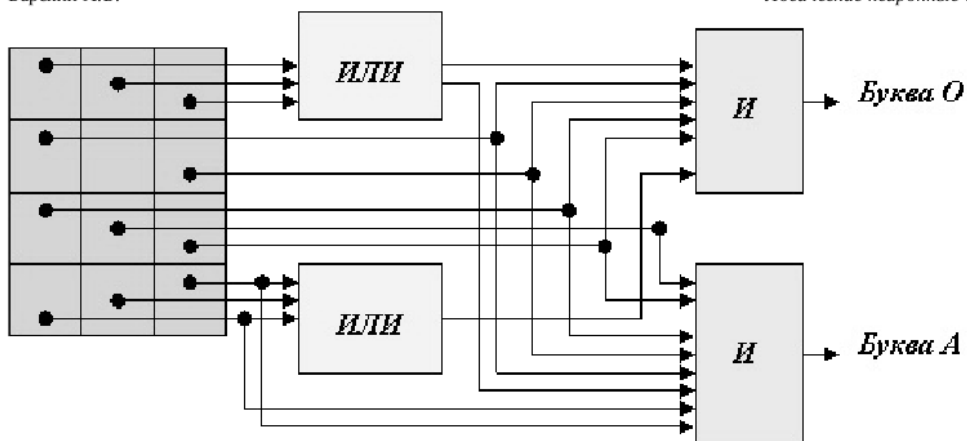


Рис. 2.3. Совместное обучение буквам "О" и "А"

Природа (и мы, как ее часть) никогда не располагает точной, определенной и достоверной информацией. Засветка клеток экрана, как и *рецепторов* нашего глаза, не бывает полной, образ не бывает правильным, присутствуют шумы, пропуски и т.д. Тогда жизненную важность обретают понятия схожести, ассоциаций. "На что более всего похож "показанный" образ, возникшая ситуация, и какие ответные действия наиболее обоснованы?" - вот вопрос, определяющий принцип нашей жизни среди многих опасностей и свершений. Ассоциативность нашего мышления является абсолютной.

Значит, надо уйти от вполне определенных булевых переменных (0, 1, "да - нет", "белое - черное" и т.д.) в сторону неопределенности, достоверности или других оценок информации, - в сторону действительных переменных.

Но тогда необходимо уйти и от булевой алгебры, так как понятия конъюнкции и дизъюнкции для действительных переменных не определены.

Тут и приходит на помощь анализ и применение принципов природной реализации - принципов нейронной сети, воплощенных в нашем мозге.

Преобразуем полученную нами обученную схему в нейронную сеть (рис. 2.4).

Каждая клетка экрана - это *нейрон - рецептор*, который в результате засветки обретает некоторую величину возбуждения, принимающую значение между нулем и единицей. *Рецепторы*, заменившие экран, образуют входной, или рецепторный, слой нейросети. Каждый *конъюнктор* и *дизъюнктор* заменим единой для всей сети моделью *нейрона*. Введем выходной слой сети, состоящий в нашем примере из двух *нейронов*, возбуждение которых определяет результат *распознавания*. Назовем *нейроны* выходного слоя по названиям букв - О и А.

Рецепторы, подобно экрану, возбуждаются извне. Все же другие нейроны, имитируя распространение возбуждения в мозге, реализуют *передаточную функцию*. Она преобразует сигналы на входе *нейрона*, с учетом весов этих входов (пока отложим их рассмотрение), в величину возбуждения данного *нейрона*, передаваемого далее по сети в соответствии со связями *нейронов* и достигающего выходного слоя.

Поскольку работа мозга имитируется на логическом уровне, *передаточная функция* выбирается достаточно простой. Так, в нашем примере достаточно выбрать следующую *передаточную функцию* для нахождения величины возбуждения i -го *нейрона*:

$$V = \sum_j V_j$$
$$V_i = \begin{cases} V, & \text{если } V > h \\ 0, & \text{в противном случае} \end{cases}$$

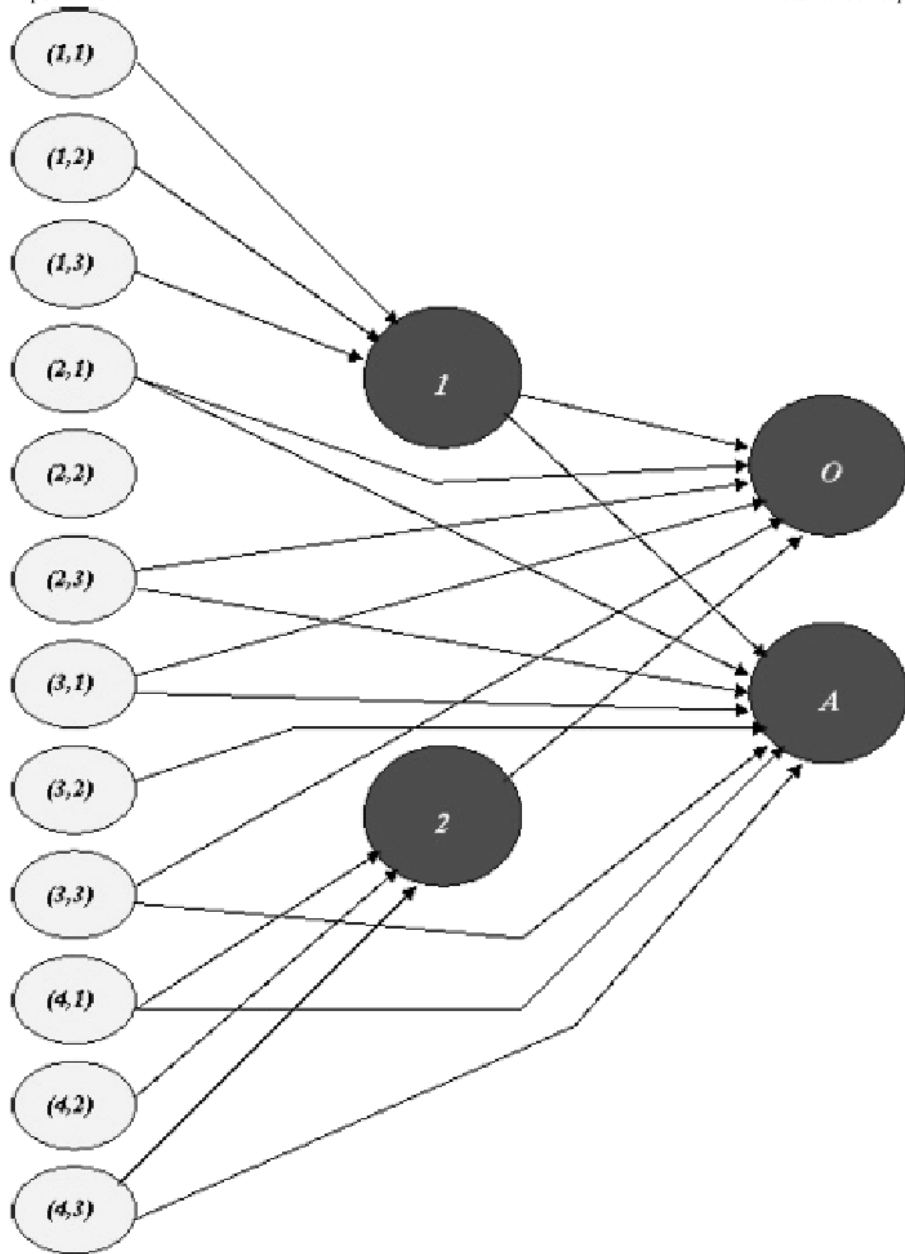


Рис. 2.4. Нейронная сеть для распознавания букв "О" и "А"

Здесь суммируются все сигналы (величины возбуждения), которые пришли от *нейронов*, связанных с данным. *Порог* может быть единым для всех *нейронов*.

Проверим, как построенная нейросеть реагирует на четко заданные эталоны букв. Под четкостью будем понимать то, что величина возбуждения, реально отражающая достоверность, максимальна и равна единице. Пусть при показе буквы О засветились нейроны - рецепторы $(1, 1), (1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 3), (4, 2)$. Тогда, при $h = 0$ величины возбуждения нейронов примут значения $V_1 = 3, V_2 = 1, V_O = 8, V_A = 7$. Нейрон О возбуждился более, чем нейрон А, указывая тем самым, что скорее всего была показана буква О. Аналогично можно рассчитать реакцию нейросети на все возможные конфигурации четко заданных эталонов букв.

А теперь введем ту самую неопределенность, к которой мы так стремились. Пусть в процессе показа буквы О четкость утрачена и величины возбуждения нейронов - рецепторов принимают значения $V_{(1,1)} = 0,2, V_{(1,2)} = 0,7, V_{(1,3)} = 0, V_{(2,1)} = 0,5, V_{(2,2)} = 0,1, V_{(2,3)} = 0,5, V_{(3,1)} = 0,5, V_{(3,2)} = 0,5, V_{(3,3)} = 0,1, V_{(4,1)} = 0,4, V_{(4,2)} = 0,4, V_{(4,3)} = 0,5$. Считаем: $V_1 = 0,9, V_2 = 1,3, V_O = 3,8, V_A = 3,9$.

Что ж, по-видимому, мы потребовали невозможного. Ведь при таком "крупнозернистом" экране налагаемые на него образы букв О и А пересекаются весьма существенно, и зашумленный показ букв обладает малой устойчивостью.

Рассмотренный принцип распознавания является обобщением принципа простейшего Перцептрона, предложенного Ф. Розенблатом в 1959 г. [4] и ставшего классическим.

2.1.2. Обучение нейросети

Теперь предположим, что структура сети (рис. 2.5), а также передаточная функция (та же), заданы, и нам предстоит обучить сеть распознаванию букв О и А.

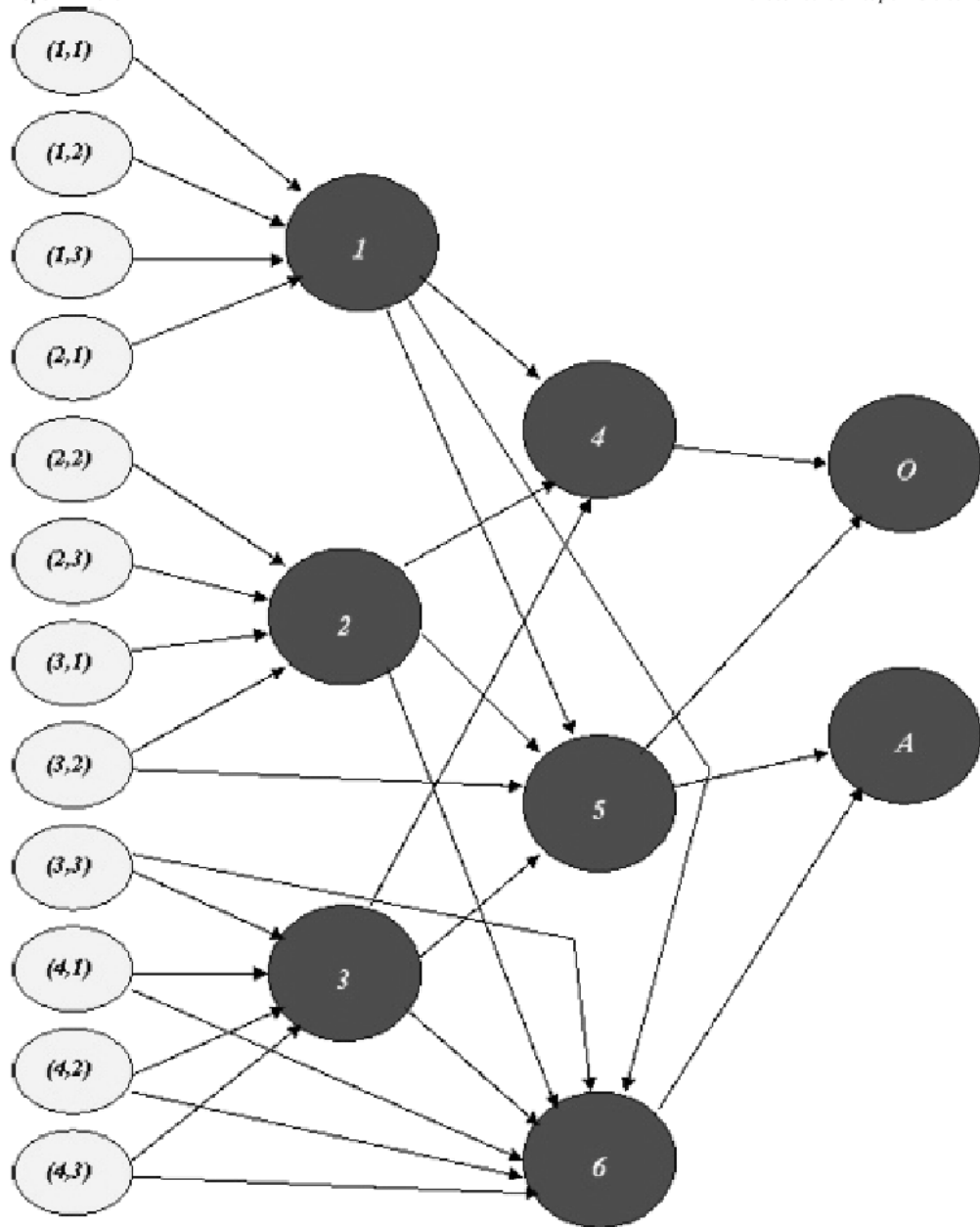


Рис. 2.5. Нейросеть, подлежащая обучению

Пусть экран разбит на столько же клеток, и мы убеждаемся лишь в том, что предложенная нейросеть содержит не менее 12 *нейронов* входного слоя, а также не менее двух *нейронов* выходного слоя.

Теперь возникла необходимость отразить истинный характер преобразования нейроном входной информации, т.е. величин возбуждения нейронов, связанных с ним. Дело в том, что все входы нейронов обладают регулируемыи (!) весами (синапсическими весами, весами связей), позволяющими направлять прохождение возбуждения в сети. С помощью этого регулирования можно связывать конфигурации и возбуждения *нейронов* входного слоя (посылка) с максимальным возбуждением определенных *нейронов* выходного слоя (следствие), формируя связи "если . . . , то . . . ".

В связи с этим *передаточная функция нейрона*, в частности заданная нам или выбранная нами сознательно, имеет общий вид

$$V = \sum_j \omega_j V_j,$$
$$V_i = \begin{cases} V, & \text{если } V > h \\ 0, & \text{в противном случае} \end{cases}$$

Здесь ω_j - синапсический вес входа или вес связи, по которой передается возбуждение от *нейрона* j *нейрону* i .

Тогда задачей *обучения* является выбор высоких весов некоторых связей и малых весов других связей так, чтобы в соответствии с функциональным назначением сети сложились отношения "посылка - следствие", подобные тем, которые мы создали выше при построении уже обученной сети. Они переводят возбуждение *рецепторов*, образующееся при "показе" *эталонов* и *образов*, в максимальное возбуждение тех *нейронов* выходного слоя, которые говорят, "что это такое".

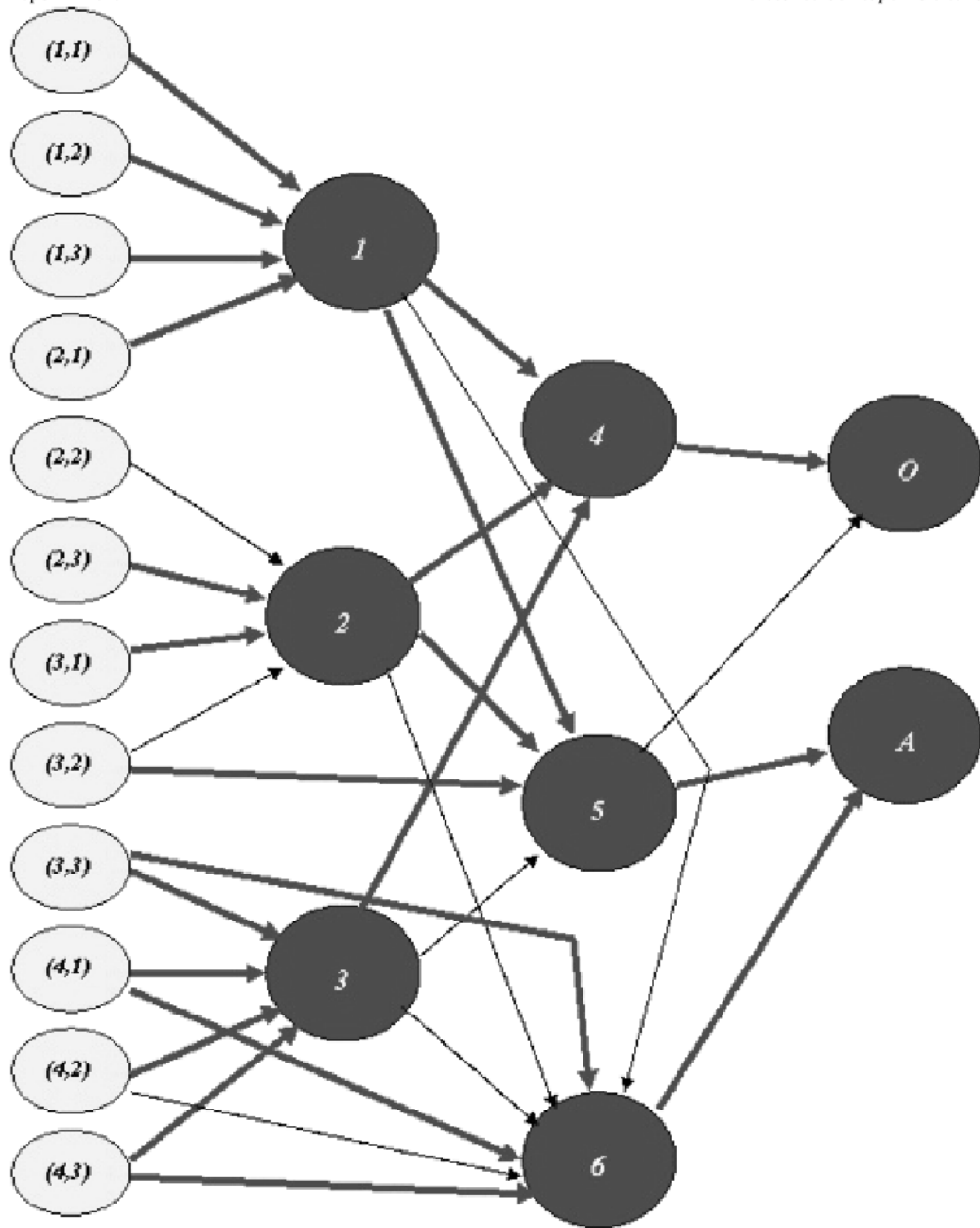


Рис. 2.6. Обученная нейросеть

Для простоты положим максимальный вес связи единичным, а минимальный оставим нулевым. Это соответствует тому, что, интерпретируя связи в сети как "проводочки", мы какие-то из них

оставим (считая их сопротивление нулевым), а другие, ненужные, "перекусим".

Изложенная идея обучения по четко заданным эталонам воплощена в алгоритме трассировки нейросети.

На рис. 2.6 приведен результат такой трассировки для нашего примера, где выделенные стрелки соответствуют связям с единичными весами, а другие - с нулевыми.

Просчитав несколько, в том числе "неопределенных", образов на входе, можно убедиться в ее правильной работе, хотя и демонстрирующей изначально выбранное слабое отличие O от A .

Получилась интересная игра. Мы задаем на входе случайные возбуждения и спрашиваем: на что это больше похоже из того, что "знает" нейросеть? Все, что надо сделать, это посчитать в едином цикле для каждого нейрона значение передаточной функции и проанализировать величины возбуждения нейронов выходного слоя.

2.2. Модель мозга

Нейросеть содержит узлы - аналоги нервных клеток - *нейронов* (рис. 2.7) (*нейроподобных элементов*, НПЭ) и их соединения - *синаптические связи*.

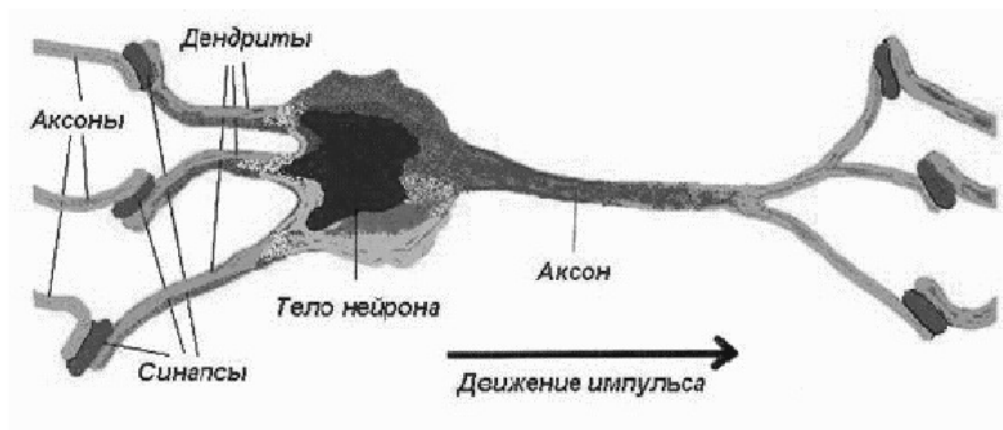


Рис. 2.7. Нейрон

Модель *нейрона* во взаимодействии с другими *нейронами* нейросети представлена на рис. 2.8.

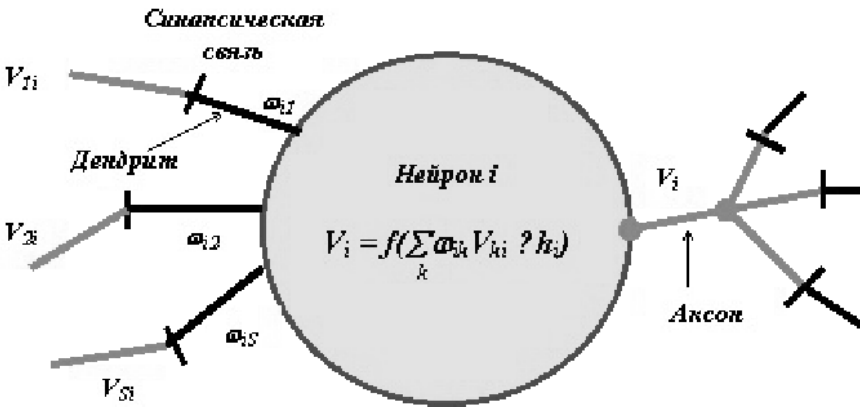


Рис. 2.8. Модель нейрона

Здесь V_{ki} - энергетические доли импульсов V_k , выработанных другими *нейронами* и поступивших на *дендриты* нейрона i , ω_{ik} - веса *дендритов*, h_i - *пороги*. В свою очередь, выработанный импульс V_i также распределяется между *дендритами* *нейронов*, с которыми связан *нейрон* i с помощью ветвящегося *аксона*. В соответствии с законом распределения энергии, величина V_i делится пропорционально значениям весов *дендритов* "принимающих" *нейронов*.

Впрочем, проблему деления (или неделения), лежащую в основе практических моделей, мы еще обсудим далее.

Каждый *нейрон* или управляем извне, или сети строятся по принципу самоуправления, используя обратные связи. А именно, можно регулировать значения весов *синаптических связей* $\{\omega_i\}$ и значения *порогов* h_i . Такое регулирование, во многих вариантах реализованное в разных моделях, и определяет возможность *обучения* и *самообучения* сети. Оно задает пути прохождения возбуждений через сеть, простейшим образом формируя связи "посылка - следствие".

На рис. 2.9 дается фрагмент нейросети, по которому мы можем представить следующее.

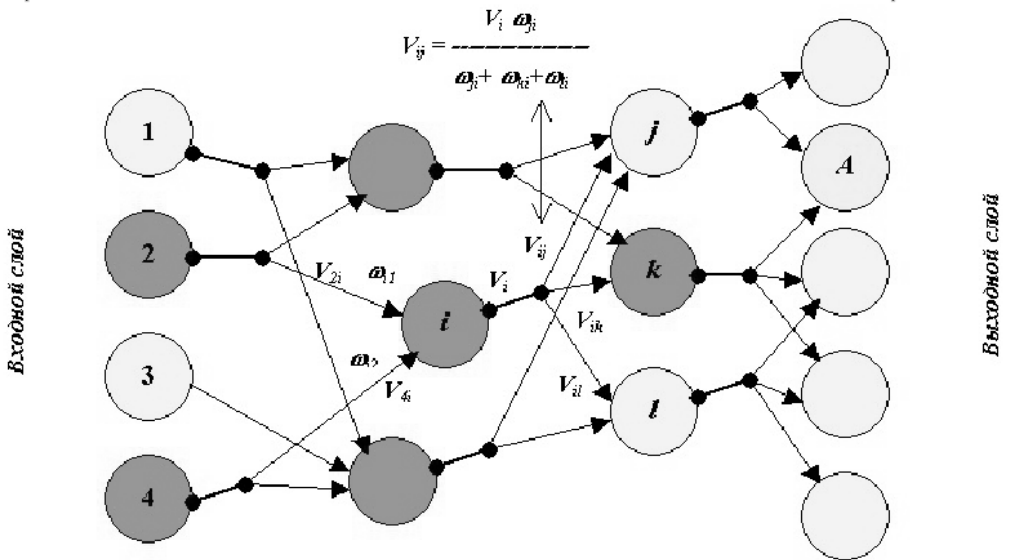


Рис. 2.9. Нейронная сеть с "правдоподобным" распределением сигнала возбуждения

1. Функции f бывают различны, но просты по объему вычислений. В простейшем случае f совпадает с *линейной формой* - указанным аргументом. Т.е. по всем *дендритам* с учетом их весов (на рис. 2.9) производится суммирование и сравнение с *порогом* "

$$V_i = \xi((\omega_{i1} V_2 + \omega_{i2} V_4) - h),$$

$$\xi(x) = \begin{cases} 0, & \text{при } x < 0 \\ x, & \text{при } x \geq 0 \end{cases}$$

2. Величина превышения *порога* является величиной возбуждения *нейрона* или определяет значение величины возбуждения (например, в некоторых моделях величина возбуждения всегда равна единице, а отсутствие возбуждения соответствует нулю). В некоторых моделях допускают и отрицательную величину возбуждения. Значение возбуждения передается через ветвящийся *аксон* в соответствии со связями данного *нейрона* с другими *нейронами*.
3. В общем случае по *дендритам* может передаваться как возбуждающее, так и тормозящее воздействие. Первое может соответствовать положительному значению веса *синаптической связи*, второе - отрицательному. Аналогичный эффект достигается

при передаче отрицательных значений возбуждения *нейронов*.

4. В сети распознают входной (рецепторный) слой, воспринимающий внешние возбуждения (например, на который подается видеоизображение), и выходной слой, определяющий результат решения задачи. Работа сети тактируется для имитации прохождения по ней возбуждения и управления им.
5. Существуют два режима работы сети: режим *обучения* и режим *распознавания* (рабочий режим).

Установим случайным образом начальные значения весов *дендритов* всей сети. Пусть нейросеть предназначена для *распознавания* рукописного текста. Тогда входной слой аналогичен сетчатке глаза, на который ложится изображение. Его подача на входной слой возбуждает в некоторой конфигурации множество *нейронов* - *рецепторов*. Пусть мы на входной слой подали и поддерживаем некоторый *эталон*, например букву А. Через некоторое время возбуждётся (максимально) *нейрон* выходного слоя, который мы можем отметить как *образ А* (прохождение возбуждения отмечено темным цветом). Т.е. его возбужденное состояние мы воспринимаем как ответ "Это буква А". Введем снова букву А, но с естественными искажениями, обусловленными почерком, дрожанием руки и т.д. Может возбуждётся тот же *нейрон*, но может и другой. Мы же хотим "научить" систему, заставить ее ответить, что это - тоже буква А, т.е. добиться возбуждения того же *нейрона* выходного слоя.

Тогда по некоторому алгоритму (один из известных алгоритмов называется *алгоритм обратного распространения* ошибки; в нем воспроизводится подход, используемый в динамическом программировании, здесь мы предлагаем алгоритм "трассировки") мы меняем веса и, возможно, *пороги* в сети на пути прохождения возбуждения так, чтобы заставить возбуждаться нужный *нейрон*.

Таким способом, предъявляя множество *эталонов* и регулируя параметры, мы производим *обучение* сети данному *образу*. (Математические проблемы несовместимости управления параметрами для разных *эталонов* оставим в стороне: в живой природе такой процесс проходит успешно, достаточно устойчиво при разумном отличии *эталонов*. Например, интуитивно ясно, что пытаться заставить один и тот же *нейрон* выходного слоя возбуждаться и на строчную, и на

прописную букву А вряд ли разумно. В лучшем случае он определит, что это буква вообще, а не, скажем, знак пунктуации.)

Обучение заканчивается тогда, когда вероятность "узнавания" достигнет требуемого значения, т.е. необходимость корректировки параметров по предъявляемым *эталонам* возникает все реже. Теперь можно работать в режиме *распознавания* - в том ответственном режиме, для которого сеть создавалась. Предъявляем сети различные буквы. Можем быть уверены, что с большой вероятностью, если мы предъявим случайно искаженную и даже зашумленную букву А (конечно, в допустимых пределах), сеть ее распознает, т.е. максимально возбудится соответствующий *нейрон* выходного слоя.

Можно существенно облегчить *обучение*, предъявляя *эталон* "в полном смысле", т.е., например, показывая букву, точно регламентируемую букварем. Тогда предусмотренная (!) степень отклонения от этого *эталона*, обусловленная почерком, будет влиять на вероятность *распознавания* этой буквы. Далее мы будем рассчитывать именно на такой способ *обучения*.

Теперь обобщим "специализацию" входного слоя, связав его с некоторыми характеристиками исходной ситуации (входного вектора), по которой необходимо принимать решение - формировать выходной вектор. Мы учим сеть по эталонным ситуациям, по которым мы знаем решение, а затем в рабочем режиме она выдает нам решение во всем диапазоне ситуаций. При этом она автоматически решает проблему, на какую "знакомую" ей ситуацию более всего похожа введенная ситуация, и, следовательно, какое решение следует выдать. Конечно, с определенной вероятностью правильности.

Такая обработка входной информации при возможном применении в сфере развлечений показана на [рис. 2.10](#) и [рис. 2.11](#).

Отметим, что в экспертных системах требуются точные данные о ситуации, чтобы выдать соответствующее ей заключение. Нейронной сети такой подход "чужд". Она выдает ответ на вопрос "На что похожа данная ситуация?" и, следовательно, какое должно быть заключение. Это еще раз подтверждает, что нейросеть *имитирует ассоциативное мышление*.

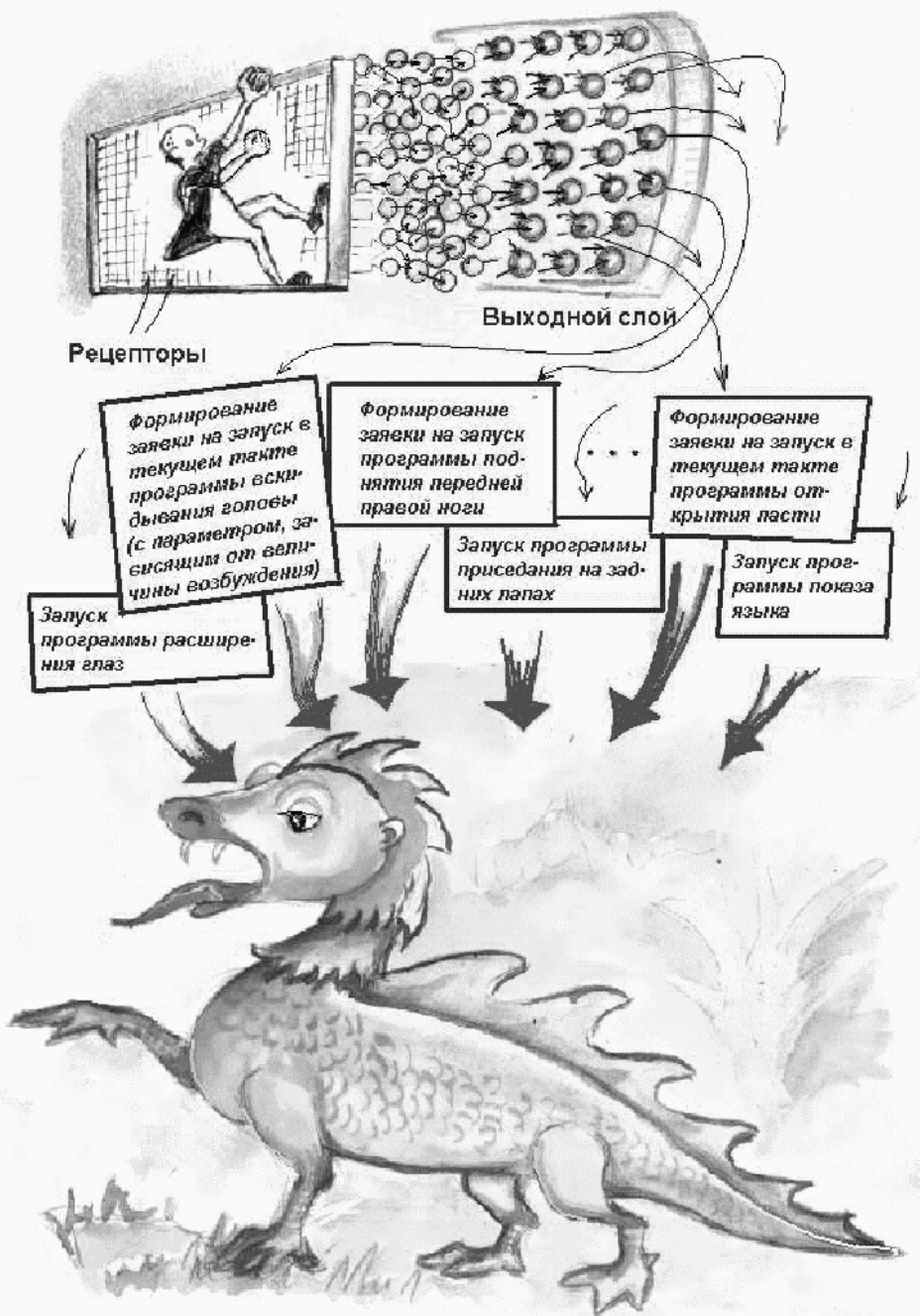


Рис. 2.10. Реакция на угрозу

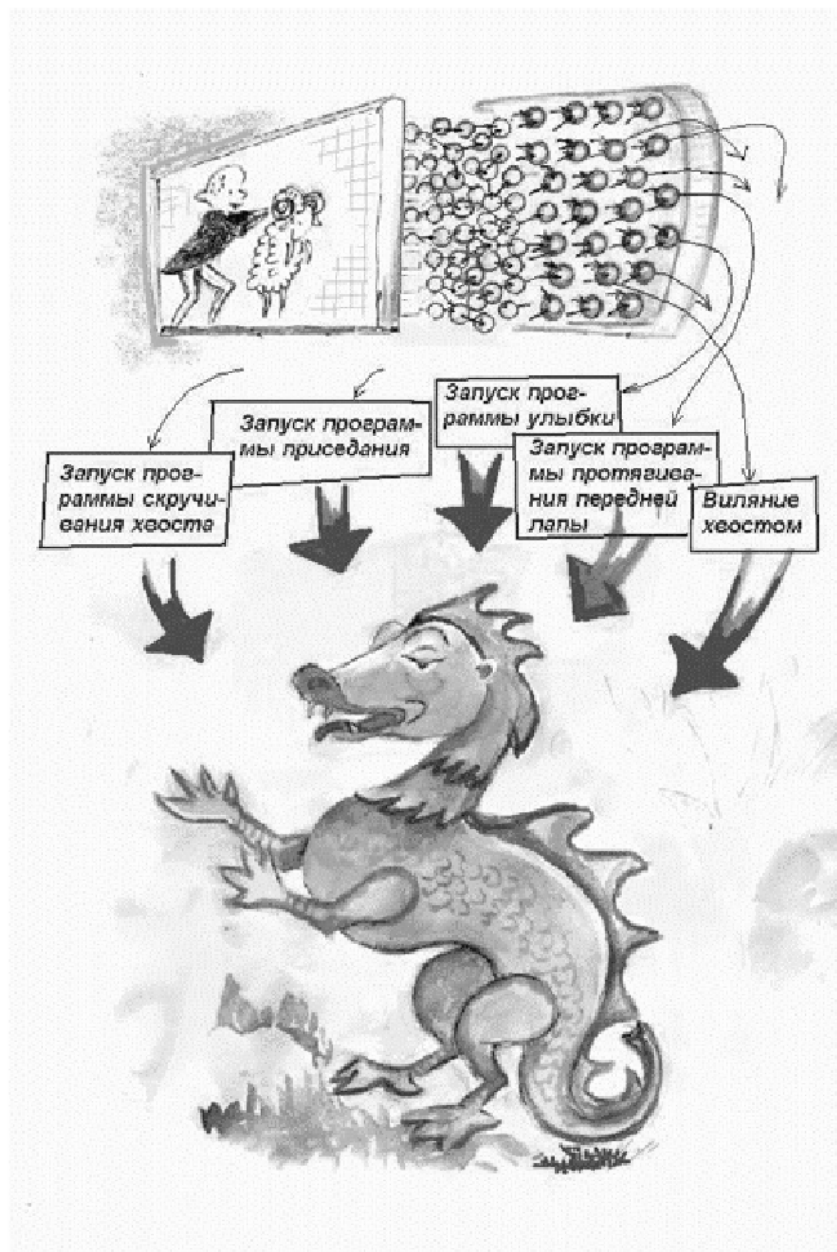


Рис. 2.11. Реакция на поощрение

2.3. Ввод и "разглядывание" эталонов и образов

Устройства ввода информации - *эталон*ов, входных векторов, исходных ситуаций - имеют определяющее значение для нейросети. С их помощью формируются и поддерживаются возбуждения входного слоя. Однако связь модели живого организма с внешней средой естественно представляет собой сложную проблему - конгломерат ряда частных технических и алгоритмических проблем. Среди них - успешно решаемая проблема видеоввода. Однако ввести в компьютер "картинку" - это лишь часть дела. Картинку надо обработать - в целом и по частям, чтобы по максимуму интересующей информации получить полные и достоверные выводы. Здесь мы ищем аналоги нашего восприятия действительности.

Мы совершаем обзор представляемой нам картины тремя способами:

- сканированием сектора обзора, разбитого на элементарные сегменты (рис. 2.12);
- сканированием сектора обзора со "своим окном просмотра" (рис. 2.13);
- спонтанным обзором, обусловленным привлечением внимания к цветовому или скоростному всплеску, быстрым увеличением размера (угрожающим приближением) объекта, указанием извне (целеуказанием) и т.д. (рис. 2.14).

Третий способ также требует сканирования сектора обзора, однако со значительно меньшими энергетическими затратами.

При первом и втором способах анализ сложнее, т.к. требует согласования всего виденного по сегментам. Это, в свою очередь, требует включения высших уровней логического вывода (интеллекта).

При третьем же способе можно добиться избирательности, чрезвычайности реакции, например на резкие движения, на бег, появление яркой расцветки в одежде и т.д. Это может с успехом использоваться в развлекательных, игровых системах.

Все способы реализуются легче, если речь идет о единственном объекте единовременного *распознавания*, например буквы, хозяина квартиры,

подписи и т.д. Ибо любая сцена, например туристская группа, пришедшая полюбоваться "умным" монстром, требует не только детального, но и совместного анализа этим монстром всех (многих) ее составляющих.



Рис. 2.12. Сканирование по строкам



Рис. 2.13. Реакция на внезапность



Рис. 2.14. Беспорядочное сканирование

Впрочем, говоря о туристах, можно говорить о конечной, усредненной реакции на всю группу. Ведя обзор, сеть постепенно, по критериям обучения "это хорошо - это плохо", воспринимая "настроение" как последовательное добавление элементов радости и огорчения, приходит к некоторому окончательному состоянию, обусловленному тем, сколько того и другого она увидела. Тогда для разных групп туристов или экскурсантов это состояние будет разным. Это может стать источником веселья и шутливого "поощрения" той группы, которая привела объект в радость, и "осуждения" группы, ввергнувшей его в печаль.

Итак, в каждом такте обзора, формируется сегмент, содержимое которого необходимо распознать. Чаще всего целесообразно допущение о том, что в элементарном сегменте (или в "окне просмотра") при дискретном сканировании находится не более чем один значимый объект. Пусть это - максимальная область текста, вмещающая единственную букву, написанную с допустимой долей небрежности. Как помочь себе же разглядеть эту букву? По-видимому, следует пытаться разместить эту букву на входном слое так, чтобы она максимально соответствовала тому размещению *эталонов*, с помощью которых производилось обучение. Тогда *распознавание* заработает правильно (рис. 2.15). Такой процесс "разглядывания" может предполагать:

- поиск возможности совмещения условного центра элемента изображения и центра экрана - входного рецепторного слоя сети (фокусировка);
- поиск варианта масштабирования элемента изображения (приближение - удаление);
- поиск угла наклона и др.

В результате таких пробных действий может вдруг "запуститься" процесс *распознавания*, хотя, возможно, и ошибочного. Что ж, в жизни бывает и так.

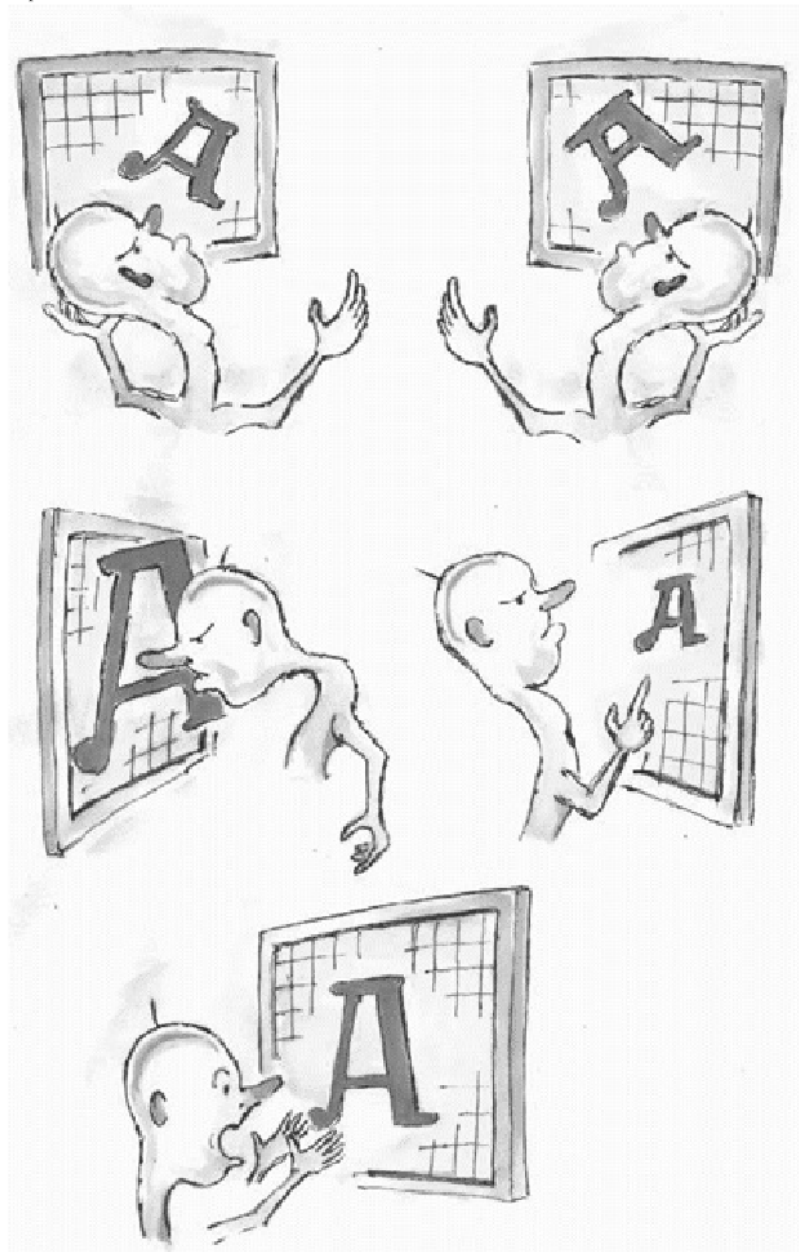


Рис. 2.15. Поиск условия узнавания

Этот процесс выделения и размещения в попытке инициировать *распознавание* мы можем сравнить с концентрацией нашего внимания и с фокусировкой, понимая, что наше зрение в каждый момент всегда

сконцентрировано на элементе изображения, держа его в фокусе, в целом производя обзор и разглядывание всего изображения [26].

Напомним, что работа нейросети тактируется. Тогда развитие сценария в увлекательной многофункциональной детской игре с обучаемым компьютерным человечком КОМПИ может быть таким, как представлено на рис. 2.16.

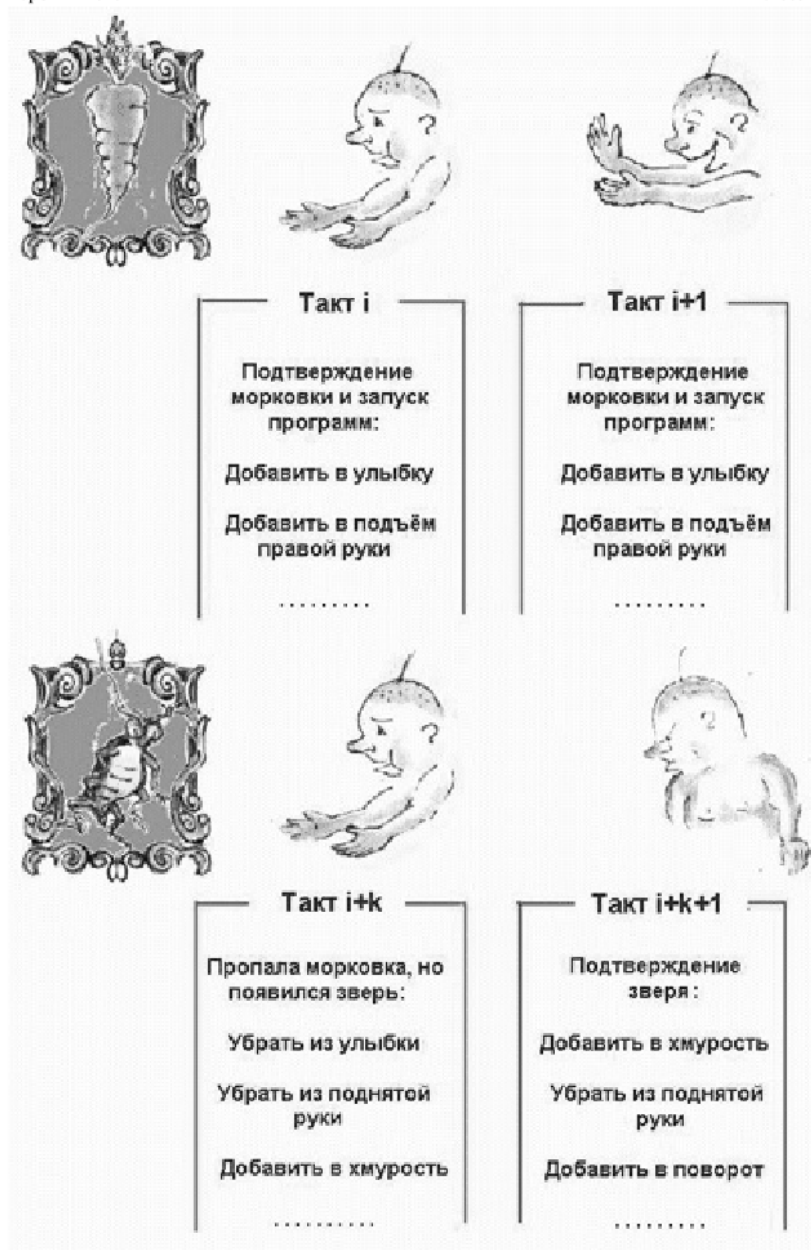


Рис. 2.16. Реакция на распознавание в реальном времени

2.4. Кора

"Крошка сын к отцу пришел,

и спросила кроха:

- Что такое хорошо

и что такое плохо?"

В. Маяковский

Мы видим, что столь примитивная оптимизация целевой функции, принимающей только два значения - "хорошо" и "плохо", может быть использована лишь на начальной стадии изучения и применения принципов нейросети. Можно построить занимательную игрушку (как весь наш мир!), веселый аттракцион . . .

Очевидно, что мы не можем оставаться на том примитивном уровне, когда по предъявляемому изображению мы получаем заключение типа "это буква х ", "это - хорошо (вкусно)", "это - плохо" и т.д. В научном приложении на таких принципах можно построить модель собачки Павлова, демонстрирующую условный рефлекс при предъявлении этикетки "Pedigree".

Но более серьезные применения должны использовать более сложные критерии логической выводимости. Ведь хронология Божественных событий Сотворения Мира закончилась созданием Человека!.. И тут начались проблемы . . . Эти проблемы обусловлены вопросом, запретным плодом райского сада: "Ну, и что же из этого следует?"

Цепочки логических выводов, производимые человеком, содержат не одно, а много звеньев. Сделанные выводы вновь входят в конфигурацию изображений (входных векторов) для новых выводов, действий и т.д.

Буква нужна, чтобы распознать слово. Слово - чтобы распознать предложение, уяснить его смысл, предпринять адекватные действия и т.д. (вспомним исходную посылку Иоанна Евангелиста: "В начале было Слово").

Тогда возникает вопрос о целесообразной структуре сети, где одни выводы, собираясь в различные конфигурации, как бы множась, вновь участвуют в формировании новых выводов. Здесь можно предположить

как прямые, так и обратные связи, когда попутно сделанные выводы уточняют правильность всего умозаключения.

Технически возникает вопрос о такой концентрации выводов, когда совмещены понятия входного и выходного слоев. Это наводит на мысль о целесообразности существования такого универсального слоя - коры со связями между отдельными *нейронами* и целыми областями. На коре концентрируются выводы для немедленного участия в формировании других выводов.

По-видимому, таков ход мыслей Господа Бога, создавшего кору головного мозга не подумав о греховных последствиях. Так чисто технически можно доказать ее целесообразность.

2.5. Устойчивость, помехозащищенность и локализация максимального возбуждения

" ... Цинциннат брал себя в руки и, прижав к груди, относил в безопасное место".

В. Набоков. "Приглашение на казнь"

На рассмотренном маленьком примере, где мы предположили жесткое закрепление *нейронов* выходного слоя между образами, принцип ассоциации "на что более всего это похоже" пока не виден. Как же он появляется?

Уместны ассоциации, предположения о том, как это реализовано в природе.

Представим себе отдельно выходной слой ([рис. 2.17](#)).

Это более всего похоже на букву "А"

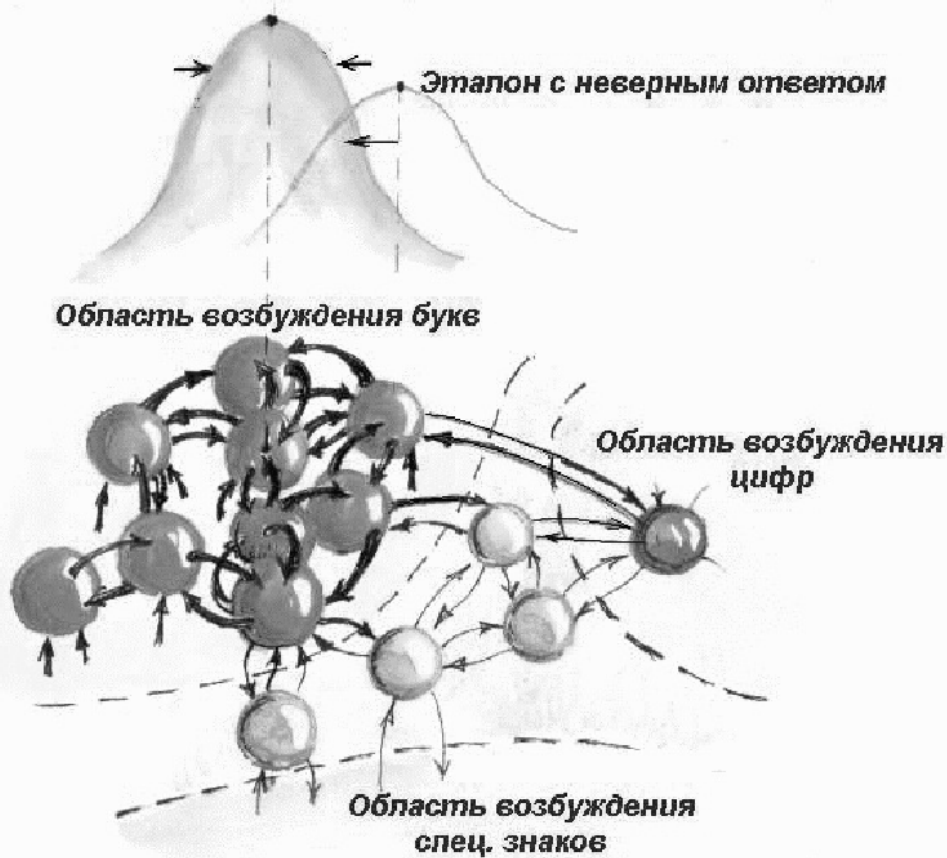


Рис. 2.17. Области возбуждения

Рассматривая прохождение возбуждения, например, при подаче того же изображения буквы А, в модели нейросети, ближе к реальному мы можем предположить, что не один *нейрон*, строго соответствующий этой букве, придет в возбужденное состояние, как это предполагается в логической модели, в его окрестности возбудятся и другие *нейроны*. Эта величина возбуждения будет угасать с ростом удаления. Нам же будет нужно, чтобы максимальной величиной возбуждения в этой окрестности обладал именно отмеченный *нейрон*. Более того, желательно, чтобы этот и только этот *нейрон* обладал высокой, существенно различимой величиной возбуждения. Это обеспечит определенность и однозначность при дальнейшем использовании

полученного вывода для принятия решения и для построения других логических цепочек, использующих это решение. Такая локализация сигнала возбуждения позволяет ответить на вопрос: "на что более всего похож предъявляемый образ, несмотря на случайные отличия и оказываемые помехи?"

Кстати, такое предположение указывает на другое, фактически реализуемое предположение о непрерывности перерастания свойства похожести одних образов в другие. Следование этому принципу при принудительном закреплении *нейронов* выходного слоя способствует построению адекватных моделей. А точнее, подавая первый раз букву А, не следует указывать системе, какой *нейрон* выходного слоя должен возбудиться. Лучше подать достаточно "правильный" эталон и посмотреть, какой *нейрон* возбудится. Этот *нейрон* и будет впредь соответствовать нашей букве. Возбуждения именно этого *нейрона* мы будем добиваться при предъявлении других эталонов.

(Хотя, мы же далее предполагаем, на основе уже рассмотренных примеров, что первоначально сеть "пуста", т.е. все веса - нулевые. Путь же возбуждения прокладывается, "трассируется".)

Способы максимизации и локализации уровня возбуждения основаны на нахождении экстремума функции возбуждения, построенной на области выходного слоя. Здесь нельзя обойтись без обмена тормозящими воздействиями между *нейронами* выходного слоя. Простейшая идея состоит в следующем. Все *нейроны* области выходного слоя имеют между собой синаптические связи, так что каждый *нейрон* связан с ближайшими *нейронами* тормозящими связями, по модулю пропорциональными величине собственного возбуждения. Тогда в итоге взаимодействия двух *нейронов* (один из алгоритмов такого взаимодействия *нейронов* выходного слоя будет рассмотрен) более "сильный" *нейрон* сохранит положительный (хотя и меньший) потенциал, сообщив более "слабому" *нейрону* тормозящее воздействие. Такое пошаговое "голосование" на фоне приходящего подтверждения от эталона и выделит сильнейшего.

Итак, мы видим, что обучение сети неразрывно связано с локализацией возбуждения на выходном слое.

Анализируя все сказанное выше, мы можем попытаться собрать

некоторую универсальную модель нейросети. В ней будут присутствовать входной и выходной слои. Картина возбуждений выходного слоя при подаче изображения на входной слой будет представлять собой (после аппроксимации в непрерывную область определения из дискретной) непрерывную функцию, максимум которой должен определить нам необходимое заключение. Однако на этапе *обучения* (да и в рабочем режиме - с удовлетворяющей нас достаточно малой вероятностью) по ряду предъявляемых *эталонов* этот максимум не совпадает с желаемым ответом. Следовательно, путь распространения возбуждений внутри сети необходимо скорректировать изменением синаптических весов *нейронов*, оказавшихся задействованными в этом процессе.

Было сказано, что предварительная *локализация максимума величины сигнала* производится в результате взаимодействия *нейронов* в области или некоторой окрестности выходного слоя или коры. Получается так, что области возбуждений оказываются "закрепленными" за типами объектов - за буквами, цифрами, продуктами питания и т.д. В развитой сети, где становится актуальным понятие коры, *нейроны* отдельных ее областей через *нейроны* внутренних слоев вновь порождают пути прохождения возбуждений в другие области коры и т.д.

Пытаясь разгадать и воспроизвести универсальную нейросеть, мы вновь и вновь пытаемся "заглянуть в зеркало". Мы видим, что хотя отдельный *нейрон* обладает возможностями взаимодействия с огромным числом *нейронов* (*нейрон* имеет до 10 тысяч *дендритов*), это взаимодействие характеризуется локальностью. А именно, несмотря на случайность связей, вероятность связи с "близким" *нейроном* значительно выше вероятности связи с *нейроном* "далеким". Об этом говорят даже исследованные длины *дендритов* и *аксонов*. (Длина *дендрита* достигает одного миллиметра; однако длина *аксона* достигает сотен миллиметров. При этом применяется, по-видимому, усредненная характеристика, вряд ли принимающая во внимание *нейроны* только лишь головного мозга.)

Такой *принцип локальности*, пронизывающий всю структуру сети, в сочетании с принципом иерархии - возможностью построения новых выводов на основе сделанных - позволяет на деле реализовать связи "каждый с каждым". Никакой вывод не может оставаться недоступным и

неиспользованным при построении сложных умозаключений.

Принцип локальности обеспечивает минимизацию входной информации, существенное влияние лишь значимых признаков на заключение, контролируемое и диагностируемое по функциональное разбиение областей нейросети, определение и выделение той области нейросети, в синаптические веса которой необходимо внести поправки в процессе обучения (например, организовать прилив крови).

Принцип локальности не отвергает существования маловероятных связей "каждый с каждым". Эта вероятность может быть высокой вследствие аномалий генетического характера. Например, человек, которого мы относим к уникамам, может видеть кожей вследствие того, что *нейроны*, воспринимающие кожные ощущения, сильно связаны с *нейронами* выходного слоя, "отвечающими" за зрение. И вместо того чтобы возбуждение в сторону последних при слабых связях угасло, оно становится результативным. Ведь в целом все *нейроны* устроены одинаково!

Большое число связей способствует высокой надежности мозга. Ведь ежедневная гибель огромного числа *нейронов*, подхлестанная алкоголем и наркотиками, а также травмы, компенсируются другими путями прохождения возбуждений, иногда даже связанными с необходимостью *переобучения*. Впрочем, ограниченный ресурс возможного не спасает, в конце концов, от деградации.

Итак, рассмотрим более подробно процесс *локализации максимальной величины возбуждения* на выходном слое, заключающейся в выделении того *нейрона* некоторой малой области, величина возбуждения которого максимальна. Он основан на подавлении тех сигналов возбуждения *нейронов*, которые не соответствуют *нейрону* с максимальным возбуждением. Т.е. если необходимо сконцентрировать сигнал и выделить *нейрон* с максимальной величиной возбуждения, это достигается с помощью подавляющих связей, с которыми действуют друг на друга "соседние" *нейроны* выходного слоя.

Пусть в целом над *нейронами* выходного слоя, условно расположенного на плоскости (x, y) , можно построить непрерывную функцию их возбуждения $P(x, y)$ (рис. 2.18), обусловленную прохождением

сигналов возбуждений в сети на основе предъявленного эталона. Будем считать, что эта функция имеет один или более максимумов. Пусть P_{ij} - значение величины возбуждения нейрона с координатами (i, j) .

Каждый нейрон (i, j) , действуя в своей окрестности, рассылает соседним нейронам, на их дендриты с отрицательными, не обязательно регулируемыми, весами, свою величину возбуждения P_{ij} , первоначально полученную из сети.

Представим взаимодействие двух "близких" нейронов, например (i, j) и $(i, j+1)$, получивших первоначально разные значения величин возбуждения из сети. Пусть $P_{ij} > P_{i, j+1}$. Тогда в очередном такте времени на входе нейрона (i, j) появится подавляющий сигнал $P_{i, j+1}$, а на входе нейрона $(i, j+1)$, и так имеющего меньшее значение величины возбуждения, - подавляющий, больший сигнал P_{ij} .

.

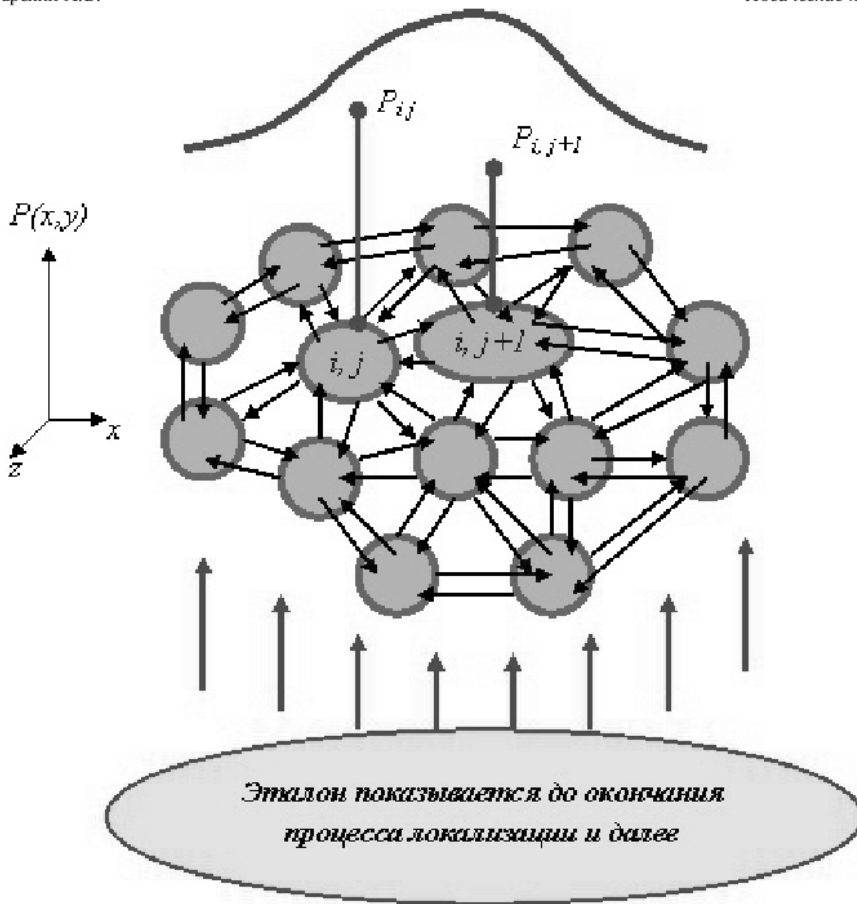


Рис. 2.18. Проблема локализации возбуждения

При подтверждаемом на входном слое эталоне, т.е. при существовании на некотором отрезке времени характера и величины возбуждений, обуславливающих возбуждение выходного слоя, на выходном слое, в частности между нейронами (i, j) и $(i, j+1)$, происходит перераспределение величины возбуждения.

А именно: несомненно, уменьшится значение P_{ij} , но в еще большей степени уменьшится значение $P_{i,j+1}$. В следующем такте более "сильный" нейрон еще более "ослабит" более "слабый" нейрон, который, в свою очередь, сможет еще в меньшей степени "ослабить" более "сильный" нейрон и т.д. Более того, "слабый" нейрон может "слабеть" до тех пор, пока взвешенная сумма подаваемых ему сигналов не станет

меньше его порога.

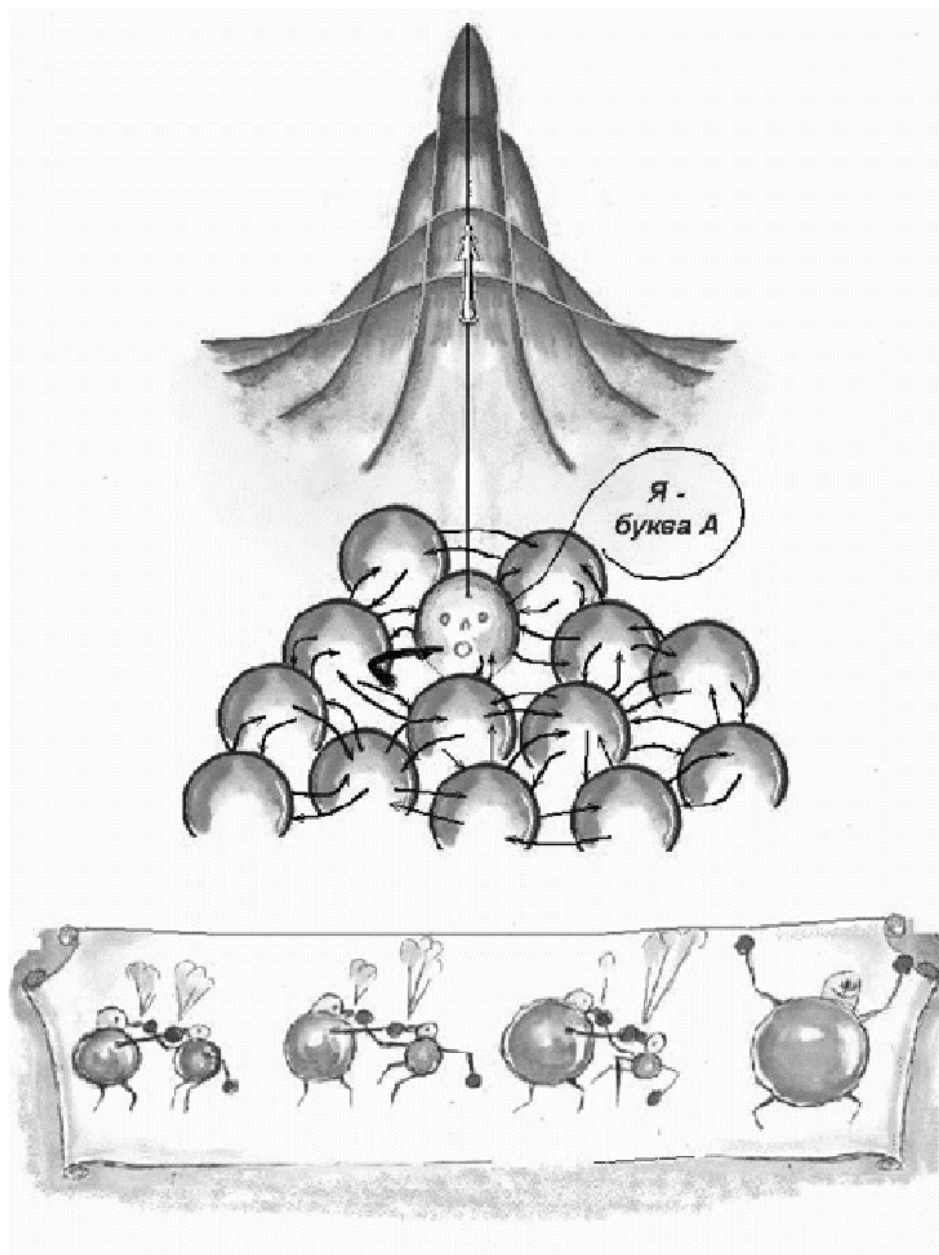


Рис. 2.19. Локализация возбуждения

Рассматривая этот процесс в рамках взаимодействия всех *нейронов* области выходного слоя, можно сделать вывод о постепенной

концентрации высокого уровня возбуждения, присущего одному или нескольким *нейронам* и определяющего один или несколько *локальных максимумов*.

При таком взаимодействии *нейронов* области возбуждения выходного слоя происходит лишь усиление сигнала наиболее возбужденного *нейрона*. Полное подавление возможно лишь на границе этой области. Если где-то внутри области возбуждение некоторого *нейрона* окажется подавленным полностью (сигнал не преодолевает *порога*), то в следующем такте этот *нейрон* не сможет подавить сигнал того *нейрона*, который до того имел более слабый сигнал возбуждения. Тогда возможно появление в такой области возбуждения нового локального максимума. Таким образом, веса отрицательных связей должны способствовать максимальному усилению возбуждения того *нейрона*, который первоначально продемонстрировал максимальное возбуждение, при затухании возбуждения *нейронов* в сторону периферии.

На человеческом языке это означает: "Любое угадывание не исключает сомнений".

Однако всегда ли необходимо на выходном слое локализовать величину возбуждения?

По-видимому, такое усиление уровня возбуждения необходимо для того, чтобы единственный *нейрон* выходного слоя преодолел некоторую "планку" (изменяющуюся?), чтобы объявить себя ответственным за сделанный вывод или решение.

Однако следует отметить, что *локализация и максимизация возбуждения* на выходном слое особенно важны тогда, когда действительно необходима высокая степень определенности. Это важно в том случае, если получаемый вывод (решение) немедленно участвует в цепочке последующих, использующих его выводов.

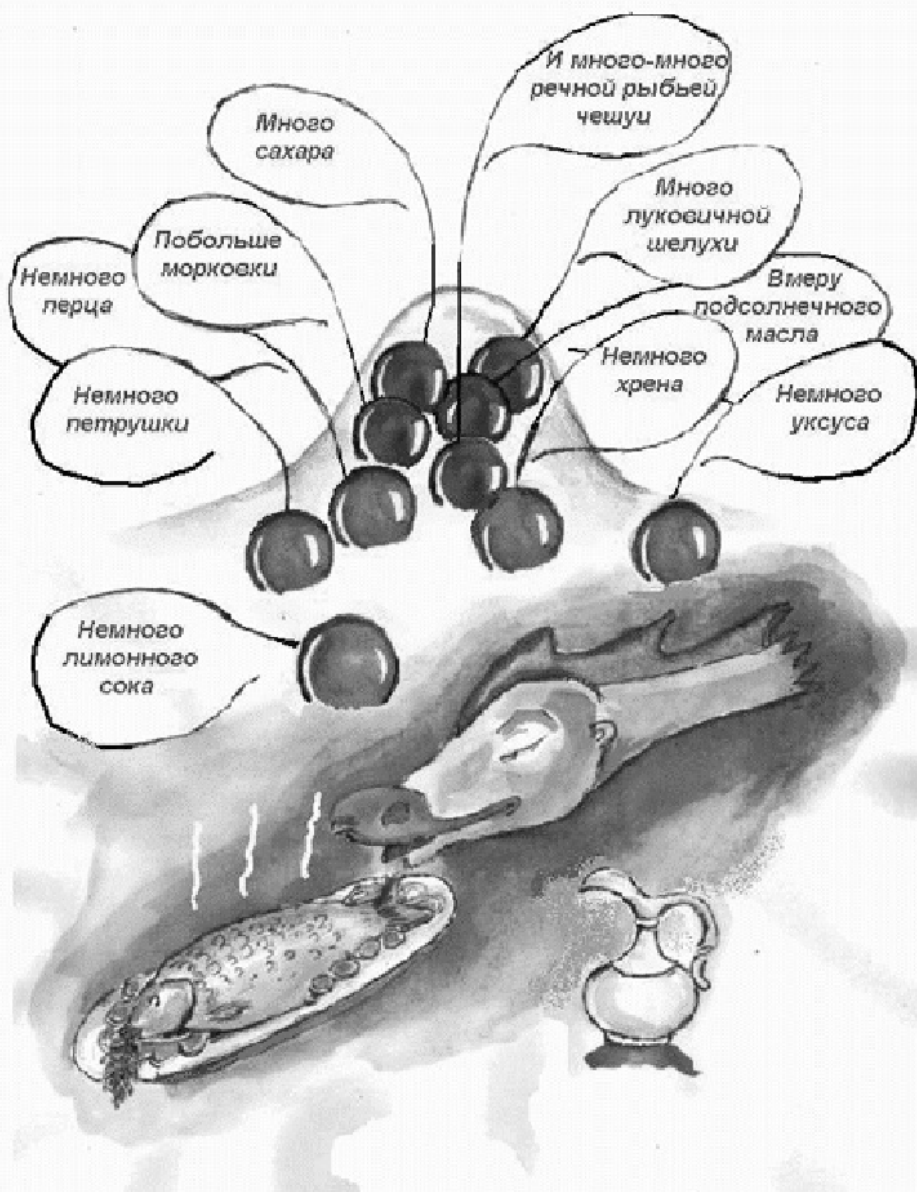


Рис. 2.20. Нейродегустатор

В конце концов, все обусловлено назначением сети, решаемой задачей. Можно представить себе возможный аттракцион - реакцию фантастического чудовища на изображение, как это представлено на

рис. 2.10 и рис. 2.11. По виду изображения инициируются те или иные программы действий: радости, гнева, поднятия лап, виляния хвостом и т.д. Возбуждение определенных *нейронов* выходного слоя связывается с запуском соответствующих программ. Величина возбуждения может являться основным параметром для этих программ. Программы не исключают друг друга, и в одном такте могут запускаться несколько программ.

Можно представить радостную модель гурмана-дегустатора (рис. 2.20), по аромату блюда определяющего состав использованных ингредиентов и приходящего в восторг или в уныние от представленного букета.

В большинстве частных задач, где нейросеть обучается с помощью "учителя", т.е. на основе действий извне при ее настройке, присутствует элемент принудительного закрепления *нейронов* выходного слоя за выводами. В процессе последующего *обучения* преимущественно с помощью весов *синаптических связей* добиваются адекватной реакции сети.

2.6. Пространство признаков

Выше неявно проявилась лишь следующая модель. *Образ* буквы, например а, налагается на входной слой нейросети - *рецепторы*. Конфигурация возбужденных *рецепторов*, порождая прохождение возбуждения через внутренние слои нейросети (образуя путь возбуждения), определяла возбуждение (если не гасло по пути) одного из *нейронов* выходного слоя, говорящего "это буква а". Интуитивно ясно, что устойчивость такой схемы по отношению к огромному множеству конфигураций возбуждений *рецепторов*, соответствующих одной только букве а, вряд ли высока. Вглядываясь в себя, мы видим, что такое непосредственное *распознавание* осуществляется далеко не всегда, особенно на этапе получения школьного образования, ибо наше образование получается на основе признаков и определений (правил вывода).

Как мы характеризуем строчную букву а? Это кружочек, справа примыкает палочка с хвостиком вправо.

А прописная А? Две палочки с перекладиной. Но ведь буква Н тоже

соответствует этому определению. Тогда мы добавляем: две палочки, соединенные вверху. (Кстати, соединение вверху может быть в результате небрежности лишь обозначено. Тогда о намеке на него мы можем судить по наклону палочек. Дальнейшая небрежность может привести к неразличимости букв А и Н.)

Значит, существует ряд признаков, лежащих в основе определений. И мы на интуитивном уровне понимаем, что такой способ *распознавания* гораздо более устойчив к искажениям и особенностям почерка, однозначен и надежен. Ведь при изображении буквы А можно допустить не только небрежность в верхнем соединении палочек, но и значительную разницу в общем наклоне буквы, в длинах боковых палочек, в месте расположения перекладины, в ее наклоне и длине и т.д. Искажение может привести к сомнениям лишь при крайней схожести на цифру 4, на телеграфный столб или на греческую Δ . Но даже в этом случае окончательный вывод может быть сделан на основе контекста, т.е. по использованию дополнительных признаков "по умолчанию".

Значит, в нашем случае необходимо ввести такие признаки, как наличие кружка, палочек, хвостиков, их взаимного расположения и т.д. То есть необходимо построить *пространство признаков*, преобразовать наши входные изображения в это пространство, и тогда появится возможность получения более определенного и устойчивого к искажениям заключения.

Перевод входного изображения в *пространство признаков* значительно расширяет возможности "разглядывания" - масштабирования, размещения, поиска угла наклона и т.д., т.е. позволяет с более высокой достоверностью производить *распознавание*.

Например, изображение танка может в разных ракурсах ложиться на входной слой *рецепторов*. Конечно, можно запомнить, что "и это - танк", "и это - тоже танк" и т.д. Но если ввести хотя бы такое определение, достаточное для соседки - тети Маши, как "массивный корпус на гусеничном ходу (тоже нуждается в определении!), а сверху башня с дулом пушки, и все такое зелененькое", то это научит хотя бы принимать меры предосторожности.

Построение современной нейросетевой технологии

На основе подхода, освещенного в лекции 1, подробно исследуется пример построения системы принятия решений. Экспериментально устанавливается корректность построений, приводящих к использованию логической нейронной сети. Обсуждаются проблемы формализованного представления и обработки нейронной сети.

"В огороде бузина, а в Киеве дядька".

Поговорка

Действительно, как связать между собой вещи несовместимые, вызывающие насмешку, но несомненно влияющие на степень мрачных раздумий о смысле жизни?..

И здесь мы сталкиваемся с действительно *трудно формализуемой задачей*, требующей мобилизации наших способностей *ассоциативного мышления!*

... Бытует теоретико-практическая предпосылка, объявляющая, что *нейросеть* формируется "под задачу". Однако в природе есть идеальная, универсальная, "унифицированная" *нейросеть* - наш мозг!.. И каждый из нас не испытывает серьезных технических трудностей, если, совершая жизненную ошибку, становится не коммерсантом-коробейником, а, скажем, блестящим специалистом по реликтовым захоронениям в мезозойский период. Более того, говорят, что если человек талантлив в одном, то он талантлив во всем.

В книге "От кого мы произошли?" (М.: ООО "АиФ Принт", 2000), в главе "Почему мы смотрим друг другу в глаза?" Э. Мулдашев говорит об известных ему 22 параметрах, отражающихся на глазной области лица и динамически описывающих морально-психологическое состояние собеседника-партнера. Однако обсчитать на компьютере удалось лишь два. Автор удивлен той колоссальной скоростью, с которой ведет такую обработку человеческий мозг: "Воистину велик Бог, создавший такое компьютерное совершенство мозга!"

Мы можем добавить, что кошка тоже смотрит в глаза своему

благодетелю, оценивая его расположение: даст еду или не даст? А вот волк не смотрит в глаза. Он бегло, с холодным равнодушием взглядывает из клетки на движущееся пока еще мясо: палач избегает взгляда своей жертвы.

Мы уже не удивляемся высокой скорости мозга. Мы знаем, что это - распараллеливание обрабатываемой информации. И мы понимаем, что без параллельных вычислительных средств заниматься моделированием работы мозга, т.е. "больших" нейросетей, - бессмысленно.

Именно возможности параллельной обработки информации, наряду с допустимым отсутствием формальных расчетов - только на основе *ассоциативного мышления*, - привлекли к себе внимание проектировщиков супер-ЭВМ в период т.н. "Японского вызова" в начале 1980-х годов.

3.1. Начинаем решать пример

"В России революция - дрогнула мать
сыра земля, замутился белый свет..."

Артем Веселый, "Россия, кровью умытая"

Нейросетевые технологии основаны на моделировании деятельности мозга. А как сказал Р. Шеннон, "Моделирование есть искусство" [29]. И как любое искусство, модель нейросети может базироваться на примерах, опыте и общих рекомендациях.

Рассмотрим пример, который навеян славным временем революционной перестройки, предлагающей нам эталоны актуальности, культурного ориентира и предприимчивости.

Вася и Петя - друзья. Нет, не в том смысле: они почти нормальной сексуальной ориентации. Скорее в смысле вечной святой мужской дружбы, без смущения применяющей слово "друг". Обозначим A - множество друзей,

$$A = \{\text{Вася, Петя}\}$$

Вася и Петя - крутые парни. Они плохо учились в школе, и это хорошо! Они создали "крышу", под которой успешно трудится ряд палаток

$$C = \left\{ \begin{array}{l} \text{”Оксана”Роксана} \\ \text{”Марина”Регина} \\ \text{”Св.Аполлинария”} \end{array} \right\}$$

заботливо опекаемые хозяйками, соответственно, Оксаной и Роксаной, Мариной и Региной, а также Аполлинарией. Палатки реализуют продукцию фирм

$$B = \{\text{КрасныйКиллер, Пират, НочнаяБабочка}\}$$

Фирма Красный Киллер в секретных подвалах славных подразделений бойцов холодной войны на основе бабушкиного самогона и контрабандного синтетического спирта гонит всемирно известную вино-водочную продукцию отличного качества. Фирма Пират производит аудио- и видеопродукцию и другие культурные ценности. Фирма Ночная Бабочка стряпает французскую косметику из мосластых московских дворян.

Ситуацию контролирует дядя Рамзай из налогового ведомства, который имеет свой маленький частный бизнес. С каждой сложившейся ситуацией, определяемой тем, кто из друзей какие палатки "накрыл" и чья продукция находилась на реализации, дядя Рамзай связывает свою долю прибыли, основанную, мягко говоря, на шантаже. Дядя Рамзай имеет свой штат осведомителей: пару бомжей - жертв предыдущей амнистии, и пару-тройку голопузых апологетов трудного детства, которые с некоторой долей достоверности, за небольшую мзду и мелкое попустительство, доставляют ему информацию.

Дядя Рамзай - прогрессивный бизнесмен, и оценки прибыли решает проводить на высоком математическом уровне, обратившись за помощью к нам (рис. 3.1). Мы хорошо учились в школе, и это - плохо! Мы, как истинные альтруисты и ученые-бессеребренники, с радостью поможем ему, - бесплатно.

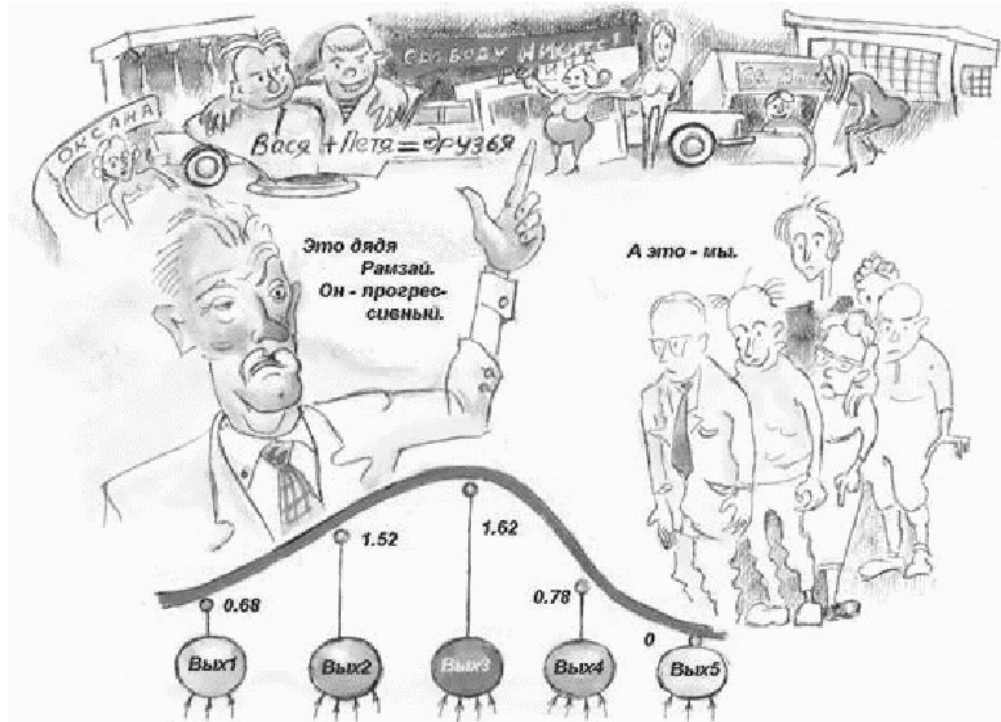


Рис. 3.1. Предмет исследования

А информации приходится обрабатывать дяде Рамзаю много. Он, прямо скажем, работает в условиях неопределенности и усиленных помех. Судите сами. Оксана делит любовь между Васей и Петей. Роксана - пока нет. Марина и Регина, жалея, подкармливают юных следопытов. Аполлинария вообще закадрила хахаля из местной мэрии и разъезжает в длиннющем "линкольне". Тщетно пытаюсь разрушить узы бескорыстной дружбы, фирма Пират напрямую подмазала Васю, снизив нагрузку вымогательства на свою продукцию. Петя, кажется, пошел на нарушение Конвенции и вторгается в область, контролируемую конкурентами. (Ох, не избежать благородной разборки, со стрельбой и окровавленными трупами!) Скоро отмотает свой срок Никита, и предприятие расширится и т.д., и т.д., и т.д. . . .

Все такие обстоятельства прямо или косвенно влияют на долю прибыли дяди Рамзая.

Однако, разбираясь в столь сложной ситуации - для демонстрации

действительно очень трудно формализуемой задачи, - мы чувствуем, как чем-то липким покрываются наши честные ладошки. А потому мы решительно отталкиваемся от ... и со сладким упоением возносимся на уровень милого сердцу абстрактного, математического, формально-логического мышления.

Но, прежде всего, принимая столь ответственный заказ, мы хотим четко уяснить, что хочет дядя Рамзай, - чтобы все же максимально формализовать задачу. И после долгих согласований мы устанавливаем:

1. Он хочет, задавая исходную информацию на входе той системы, которую мы для него создадим, на основе, возможно, не полной или недостоверной информации своих агентов, все-таки распознать с наибольшей определенностью, что это за ситуация (на какую ситуацию в наибольшей степени указывают сложившиеся обстоятельства), чтобы знать, на какой навар можно рассчитывать.
2. Он хочет, задавая исходную ситуацию на входе системы, установить среднюю величину прибыли (учитывая, что в разной степени речь идет о нескольких возможных ситуациях).
3. Он хочет сделать вывод о частоте появления различных ситуаций, чтобы перераспределить тарифные ставки за умолчание о шалостях Васи и Пети.

Итак, ступим на путь абстрагирования.

Пусть по стечению обстоятельств, которые мы будем называть событиями, принимаются решения. Решения образуют конечное множество. Каждое решение соответствует некоторой, в общем случае не единственной, комбинации событий. Предположим наличие нескольких вариантов одного события.

Введем "удобную", не обязательно единственно возможную, иерархию событий на основе их совместимости и алгоритмически привычного представления.

Считая, что варианты каждого вида событий образуют исчерпывающее множество, алгоритм работы системы можно записать как

if A1 then
if B1 then R1 else
if (C1 ∨ C2 ∨ C3)
then R2 else R3
else
if B3 then R4 else R5.

Здесь R1 - R5 - принимаемые решения.

Одно решение соответствует некоторой, в общем случае не единственной, комбинации событий. Для изображения таких комбинаций воспользуемся записями, например, вида

$A1 \wedge (B2 \vee B3) \wedge (C4 \vee C5)$. Это означает, что Вася отправился в палатку то ли к Регине, то ли к Аполлинару, торгующей продукцией то ли фирмы Пират, то ли фирмы Ночная Бабочка. Однако эта же запись означает, что все составляющие ее конъюнкции $A1 \wedge B2 \wedge C4$, $A1 \wedge B3 \wedge C4$ и др. приводят к одному и тому же решению.

Проанализировав и перебрав все возможные ситуации, с учетом одинакового принимаемого решения, получим систему логических высказываний - *предикатов*, как основу формализации задачи при построении нейросети:

$$\begin{aligned}
 & \textit{if } A1 \wedge B1 \wedge (C1 \vee C2 \vee C3 \vee C4 \vee C5) \textit{ then } R1; \\
 & \quad \textit{if } A1 \wedge (B2 \vee B3) \wedge (C1 \vee C2 \vee C3) \textit{ then } R2; \\
 & \quad \quad \textit{if } A1 \wedge (B2 \vee B3) \wedge (C4 \vee C5) \textit{ then } R3; \\
 & \quad \textit{if } A2 \wedge B3 \wedge (C1 \vee C2 \vee C3 \vee C4 \vee C5) \textit{ then } R4; \\
 & \textit{if } A2 \wedge (B1 \vee B2) \wedge (C1 \vee C2 \vee C3 \vee C4 \vee C5) \textit{ then } R5.
 \end{aligned} \tag{3.1}$$

Тогда, например, первое логическое высказывание означает "Если Вася отправился в одну из пяти палаток, и все они торгуют сегодня продукцией фирмы Красный Киллер, то следует принять решение R1 (например, заказать туристическую путевку)".

Второе логическое высказывание означает: "Если Вася посетил одну из палаток C1, C2 или C3, торгующих сегодня продукцией фирм B2 и (или) B3, то следует принять решение R2 " и т.д.

3.2. Возбуждение входного слоя

Определим теперь возбуждение *входного слоя*, учитывая то, что по нейросетевым технологиям решаются сложные, чаще всего *трудно формализуемые задачи*. Исходная информация этих задач может быть настолько несовместима по смыслу, типам данных и единицам измерения, что приведение ее к некоторому количественному воплощению - величине возбуждения нейронов *входного слоя* - представляет серьезную проблему.

Например, как объединить величину превышаемой водителем скорости и тип автомобиля иностранного производства со вчерашним неудачным выступлением любимой автоинспектором футбольной команды, - при нахождении величины штрафа? Ведь каждый из перечисленных факторов должен определить некоторые общие, приведенные значения возбуждения.

Такое приведение также зависит от задачи. Поскольку нейроны - нечто стандартное для данной задачи или класса задач, то каждая характеристика нейрона - величина возбуждения, веса его *синаптических связей*, порог, *передаточная функция* - должны быть одинаковы или принадлежать общему (по каждой характеристике) для всех нейронов диапазону возможных значений.

Дадим рекомендации, основанные на "событийном" принципе.

В нашем новом примере о превышении скорости разобьем скорость на диапазоны штрафования, например, [90-100), [100-110), [110-120), [120-200]. За каждым диапазоном скорости закрепим нейрон *входного слоя* - рецептор. Определим его среднее значение возбуждения, равное 1. Пусть отклонение к границам диапазона скорости пропорционально уменьшает эту величину возбуждения, увеличивая величину возбуждения рецептора "соседнего" диапазона. Другой вариант задания исходных данных основан на решении вопроса принадлежности скорости некоторому диапазону. Тогда величина возбуждения превратится в булеву переменную, фиксируя событие.

Однако более универсальный подход основан на связывании величины возбуждения рецептора с достоверностью - вероятностью того, что

величина скорости принадлежит одному или нескольким диапазонам. Такой подход мы и намерены воплотить в дальнейшем.

А именно, хотя бы интуитивно (а интуиция основана на опыте) определим достоверность того, что интересующая нас величина принадлежит данному диапазону. С какой достоверностью она принадлежит второму диапазону? А третьему?

Здесь явно просматривается концепция нечетких множеств [30].

Множество называется нечетким, если элементы входят в него с некоторой вероятностью.

Но можно даже уйти от понятия достоверности как вероятностной категории. Все ли мы знакомы с понятием *исчерпывающего множества событий*, связанным с условием нормировки, т.е. с условием равенства единице суммы их вероятностей? Ведь часто можно услышать: "Даю голову на отсечение, что это так, хотя и допускаю, что все наоборот..." Главное, чтобы исходные оценки информации были относительными, отражающими принцип "больше - меньше". Это расширит популярность *нейротехнологий*, исключит необходимость специальных знаний. Ведь какие-то начальные возбуждения рецепторов, при их относительном различии, распространятся по нейросети, определяя предпочтительность принимаемого решения!

Тогда, на этапе обучения нейросети, получим возможность формирования аналога некой таблицы, в соответствии с которой будет действовать инспектор. (Однако здесь пока не рассматривается процесс обучения нейросети - формирование этой таблицы, - а только принцип формирования данных для входа в нее.)

Выделим нейроны, "отвечающие" за типы автомобилей: отечественного производства, "мерседес", "вольво", "джип" и т.д. Величину возбуждения этих нейронов будем полагать равной 1 - на этапе обучения, или равной достоверности события - в рабочем режиме. Аналогично выделим рецепторы, "отвечающие" за другие возможные события: степень интеллигентности водителя (так же по диапазонам изменения), выигрыш или проигрыш любимой команды и т.д.

Следовательно, на входном слое будут формироваться приведенные

значения возбуждения.

В рабочем режиме мы, таким образом, получили возможность использования неопределенной, недостоверной информации. Например, инспектор не смог из-за высокой скорости отличить "ниву" от "чероки". Тогда он решает ввести значение 0,5 в графу "нива" (величина возбуждения рецептора, "отвечающего" за "ниву", станет равной 0,5) и 0,5 - в графу "джип-чероки" (такой же станет величина возбуждения соответствующего рецептора). Однако, подумав, он на всякий случай вводит величину 0,2 в графу ВАЗ 2104, что также во власти его сомнений. Так же инспектор поступает и с другими характеристиками сложившейся ситуации в поисках наиболее достоверного решения по принципу наибольшей похожести.

3.3. "Схемотехнический" подход к построению нейросети "под задачу"

Реализуем подход, используемый при построении схем устройств компьютера и другой электронной техники. Выделим функционально полную, для данного применения, систему булевых функций - дизъюнкцию \vee и конъюнкцию \wedge .

Отрицание нам не понадобится, мы пока не рассматриваем тормозящие связи.

Нарушив принятые обозначения, построим (рис. 3.2) схему, реализующую алгоритм счета значения выражения (3.1). Предполагаем, что на вход будут подаваться значения булевых переменных, обозначающих события.

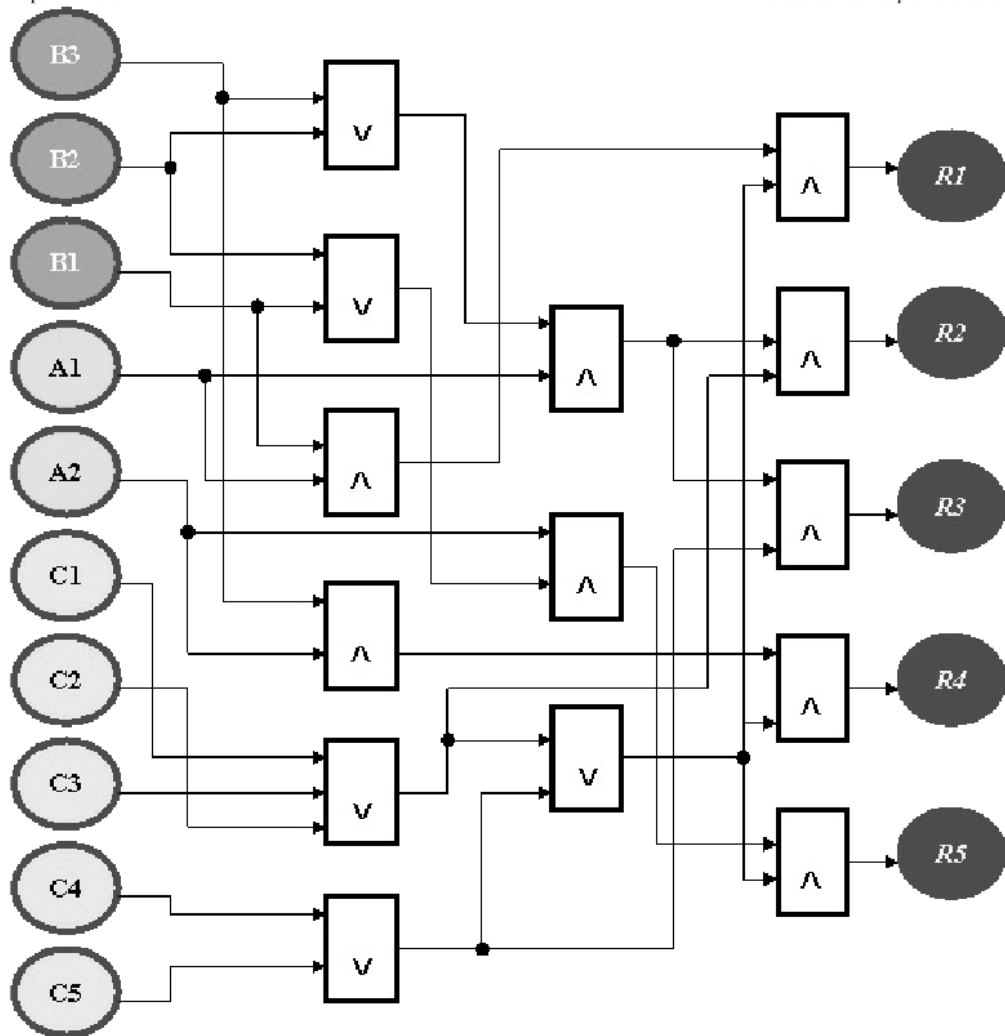


Рис. 3.2. "Электронная" схема системы принятия решений

Такая электронная схема могла бы нам верно служить, способствуя быстрому определению необходимой реакции на сложившуюся ситуацию, если мы предусмотрели все возможные ситуации, знали, какое решение соответствует каждой из них, и всегда обладали полной и точной информацией о происходящих событиях. Но ведь не зря мы обращаем внимание на те помехи и неопределенность, в условиях которых приходится жить и работать. Мы должны оперировать только достоверностями либо другими оценками событий, пытаюсь определить, какой ситуации более всего соответствуют сложившиеся

обстоятельства.

Значит, мы должны из точного, детерминированного представления перейти в область ассоциативного, неточного, приблизительного мышления! Но степень (частота) угадывания должна быть достаточно высока.

Именно здесь должна помочь *нейросеть*.

Прежде всего надо перейти от типа булевых переменных к типу действительных, введя в обращение не непреложность событий, а лишь вероятности или другие весовые оценки их наступления (электронной технике это не свойственно). Затем необходимо реализовать аналоги булевых функций над этим новым типом данных, т.е. заставить нейроны с помощью весов, порогов и самой *передаточной функции* выполнять дизъюнкции и конъюнкции с учетом вариации и неопределенности данных. При этом абсолютно достоверные данные, несомненно, приведут к известным решениям, а по неточным данным можно определить лишь вес каждого из возможных решений. Тогда по максимальному весу определим, на что более всего похожа данная неопределенная ситуация.

Выберем *передаточную функцию* произвольного (i -го) нейрона, с числом m входов-дендритов:

$$V := \xi\left(\sum_{j=1}^m V_j \omega_{ij} - h_i\right); \quad V_i := \text{if } V > 1 \text{ then } 1 \text{ else } V$$

где $\xi(x) = x$ при $x \geq 0$, 0 - в противном случае.

Здесь V_j , как всегда, - величина возбуждения (другого нейрона), поступающая на j -й вход.

Тогда нейрон-конъюнктор может быть реализован с помощью существенно высокого порога (рис. 3.3), где значение δ обусловлено некой поправкой, достаточной, чтобы для преодоления порога сигналы возбуждения поступали обязательно по всем входам.

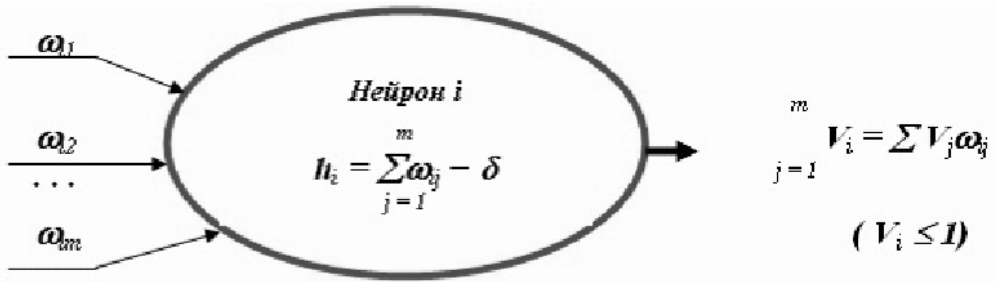


Рис. 3.3. Модель нейрона-конъюнктора

При обучении предполагается, что входные сигналы - булевы переменные, принимающие значения 0, 1. Положим $\omega_{ij} = 1/m$ и выберем $\delta < 1/m$. Тогда для того, чтобы преодолеть порог, на всех входах должны быть "1"; недостаток хотя бы одной "1" приведет к тому, что взвешенная сумма будет более чем на $1/m$ меньше указанной суммы весов.

При переходе к действительным переменным, когда вместо событий рассматриваются, например, лишь предполагаемые вероятности их наступления, экспериментальный выбор значения δ может обусловить ту границу, когда считается с возможностью данной комбинации событий нецелесообразно.

Нейрон-дизъюнктор реализуется, наоборот, при низком значении порога, но при высоких значениях весов. Порог выбирается так, чтобы уже при возбуждении на одном входе возникал сигнал возбуждения на выходе. При этом сигнал на выходе не превышает "1" (рис. 3.4).

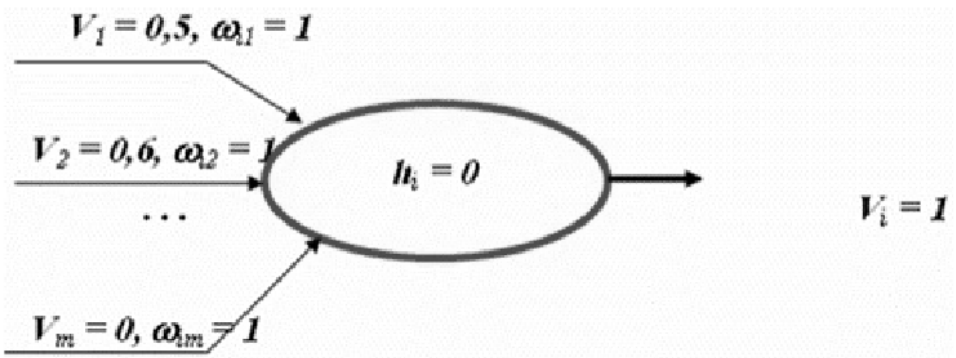


Рис. 3.4. Модель нейрона-дизъюнктора

Понятно, что при полной определенности в режиме обучения возбуждение поступает по единственному входу (нейрон реализует функцию "ИСКЛЮЧАЮЩЕЕ ИЛИ"). В условиях неопределенности предполагается, что нейрон имитирует выполнение функции "ИЛИ", допуская возбуждение более чем на одном входе.

Итак, поменяем тип данных и заменим нейронами все элементы на схеме рис. 3.2. Получим *нейросеть* на рис. 3.5, где нейроны-конъюнкторы заштрихованы.

Теперь позволим дяде Рамзаю поучиться, поэкспериментировать, задавая различные достоверности событий, - возможных или невозможных.

Например, зададим "правильную" и абсолютно достоверную ситуацию $V3 = 1, A1 = 1, C4 = 1$ (Вася отправился к Регине, торгующей ямайским ромом). Легко проследить, что в первом такте возбудятся нейроны 1 и 6, реализующие дизъюнкцию. Величина их возбуждения равна "1". В следующем такте возбуждение нейронов 1, 6 и A1 приведет к возбуждению (с величиной, равной "1") нейронов 7 и 9, а в следующем такте - сигналы возбуждения нейронов 6 и 7 поступят на вход нейрона-конъюнктора Вых3. Никакой другой нейрон выходного слоя не возбуждается.

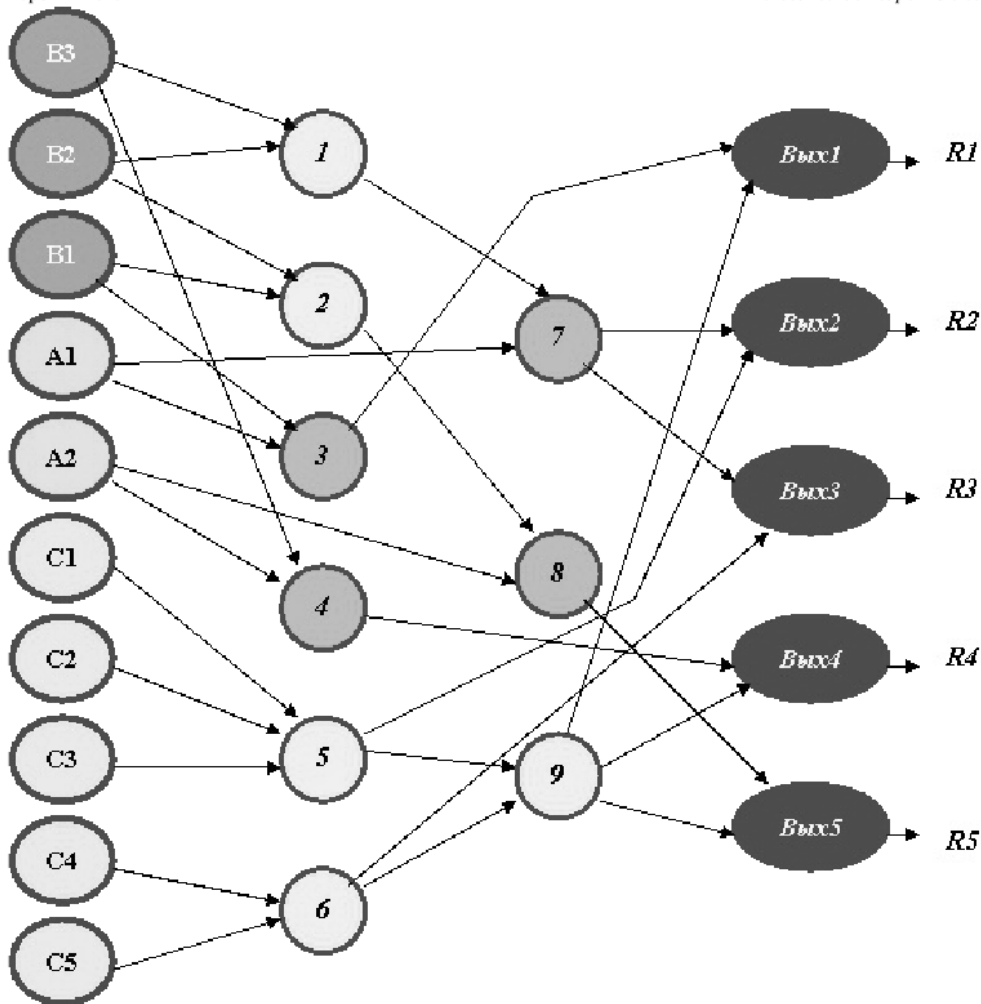


Рис. 3.5. Нейросеть с "конъюнкторами" и "дизъюнкторами"

Рассмотрим другую ситуацию, неопределенную и недостоверную.

Пусть то ли Вася, то ли Петя - "разведчик" не установил точно - направился то ли к Оксане, то ли к Аполлинару, торгующим в этот день то ли тройным одеколоном, то ли золотым диском группы "Та-ра-рам".

Дядя Рамзай, по выданной нами инструкции, решает использовать интуитивные оценки веса или, на нашем языке, оценить достоверность каждой компоненты возникшей ситуации. Поскольку прогулки как Васи,

так и Пети одинаково достоверны, то дядя Рамзай полагает величину возбуждения нейронов A1 и A2 равной 0,5 ($V_{A1} = V_{A2} = 0,5$). После долгих раздумий он по наитию полагает $V_{B1} = 0,8$, $V_{B2} = 0,8$, $V_{C1} = 0,7$, $V_{C5} = 0,8$.

Замечание. Напоминаем еще раз, что требовать *исчерпывающего множества событий* и неперемного выполнения нормировочного условия не обязательно. Достоверность может выбираться по наитию, на уровне чувств. Именно эти качества неопределенности, субъективности, наличия жизненного опыта и интуиции присущи механизмам *ассоциативного мышления*.

Сдавая *нейросеть* "в эксплуатацию", мы установили веса всех конъюнкторов равными 0,5, а дизъюнкторов - равными 1. Пороги конъюнкторов определяются значением $\delta = 0,4$. Пороги дизъюнкторов имеют нулевое значение.

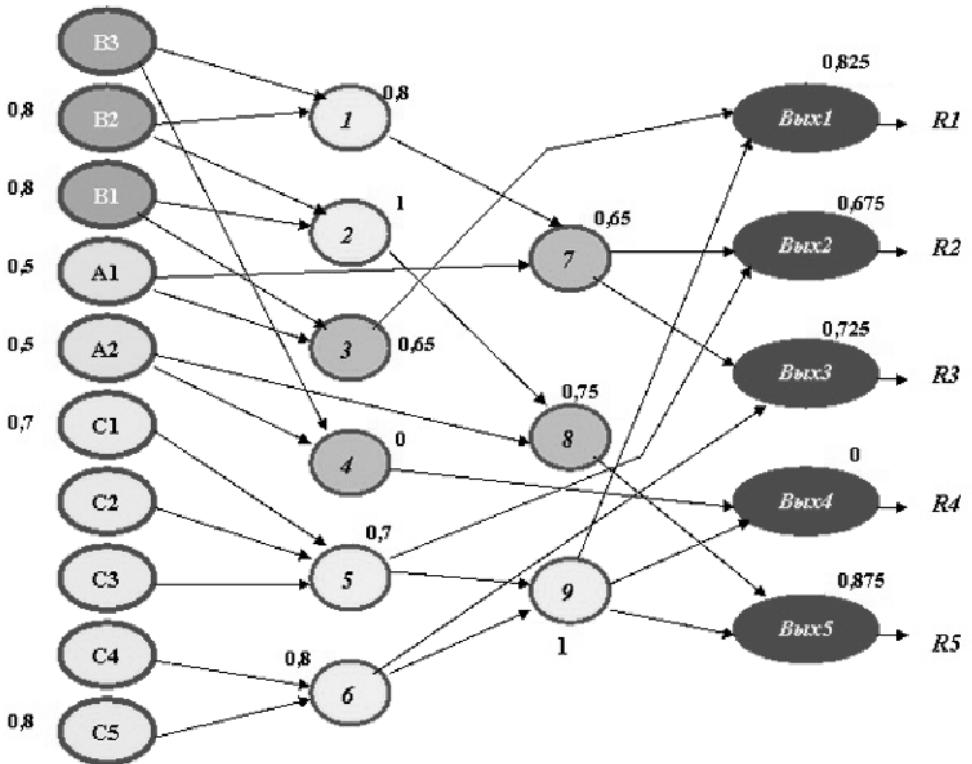


Рис. 3.6. Расчет примера

Важность данного примера требует повторения рисунка нейросети (рис. 3.6) с проставленными возле нейронов значениями сигналов возбуждения.

В итоге ситуация скорее всего имеет решение R5, и уж никак не R4. Однако ситуация, соответствующая решению R1, требует внимания.

Пусть при вполне определенной ситуации (все достоверности принимают значение "1") каждое решение R_i приносит прибыль M_i . Тогда средняя величина ожидаемой прибыли для нашей неопределенной ситуации рассчитывается так:

$$M = \frac{\sum_i M_i V_{\text{Вых}i}}{\sum_i V_{\text{Вых}i}} = \frac{M_1 \cdot 0.825 + M_2 \cdot 0.675 + M_3 \cdot 0.725 + M_5 \cdot 0.875}{3.1}$$

Конечно, полученное решение столь же неопределенно, как и тот карточный расклад, что предвещает трогательную встречу в казенном доме, поэтому мы погружаемся в дальнейший поиск.

3.4. Построение нейросети "под задачу"

Мы построили *нейросеть* - с экзотическими (с точки зрения невропатолога) конъюнкторами и дизъюнкторами.

Предположим теперь (рис. 3.5, 3.6), что все нейроны одинаковы, реализуют одну *передаточную функцию*, а веса и пороги реализуют равные и общие возможности.

Введем ту же, но без ограничения по величине возбуждения, *передаточную функцию*

$$V := \xi \left(\sum_{j=1}^m V_j \omega_{ij} - h_i \right)$$

Положим $\omega_{ij} = 0,8$, $h = 0,2$. Сеть представлена на рис. 3.7.

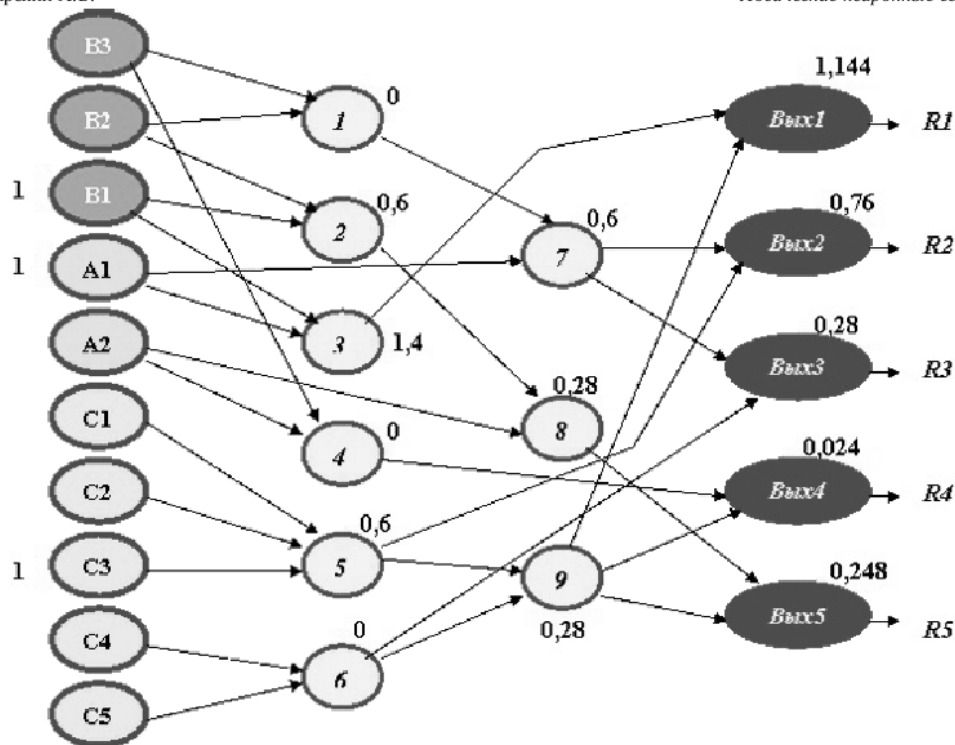


Рис. 3.7. Расчет примера на нейросети

Подадим на вход, например, ситуацию {A1, B1, C3}, требующую решения R1. Величины возбуждений нейронов показаны на рисунке.

На основе расчетов по полученной сети составим табл. 3.1, отображающую правильную (!) работу сети при получении различных решений. При этом связи, предыстория которых определена дизъюнкторами, требуют проверки не более чем одного "представителя": в рассмотренном примере получаем тот же результат, если вместо C3 положим C1 или C2.

Таблица 3.1. Примеры расчета принимаемых решений

Ситуация	Требуемое решение	$V_{\text{Вых1}}$	$V_{\text{Вых2}}$	$V_{\text{Вых3}}$	$V_{\text{Вых4}}$	$V_{\text{Вых5}}$
{A1, B1, C3}	R1	1,144	0,76	0,28	0,024	0,248
{A1, B2, C2}	R2	0,504	1,144	0,664	0,024	0,248
{A1, B3, C3}	R2	0,504	1,144	0,664	0,504	0,024

{A1, B2, C4}	R3	0,504	0,664	1,144	0,024	0,224
{A1, B3, C5}	R3	0,504	0,664	1,144	0,504	0,024
{A2, B3, C1}	R4	0,024	0,504	0,024	1,144	0,504
{A2, B1, C3}	R5	0,504	0,28	0	0,504	0,888
{A2, B2, C4}	R5	0,024	0,024	0,504	0,504	0,888
$V_{B1} = V_{B2} = 0.8, V_{A1} = V_{A2} = 0.5, V_{C1} = 0.7, V_{C5} = 0.8$?	0,824	0,529	0,593	0,312	1,003

Анализируя первые восемь строк таблицы, соответствующие достоверным ситуациям, видим, что по крайней мере максимум возбуждения определяется устойчиво верно.

Рассмотрим ту же неопределенную ситуацию. Она отражена в последней строке таблицы. Близка ли она более всего ситуации, когда Петя направился к Аполлинарии и надо принимать решение R5? Ситуация с Васей, устремившимся туда же, дает примерно тот же ответ.

Отметим, что по убыванию величин возбуждения нейронов выходного слоя, вновь полученный результат полностью совпадает с полученным по "схемотехнической" сети (рис. 3.5), так что и величина средней прибыли, по-видимому, будет близка найденной ранее.

Однако не проще было бы применять способ построения нейросети, близкий к табличному? Что, если каждую ситуацию непосредственно "замкнуть" на соответствующее решение, избежав сложной путаницы промежуточных слоев нейронов и не рассчитывая множества вариантов для нахождения максимального возбуждения и распределения возбуждения на выходном слое?

Очень часто на практике так и поступают. Поэтому широкое распространение получили так называемые *однослойные сети*. Построим такую сеть и для нашего примера (рис. 3.8).

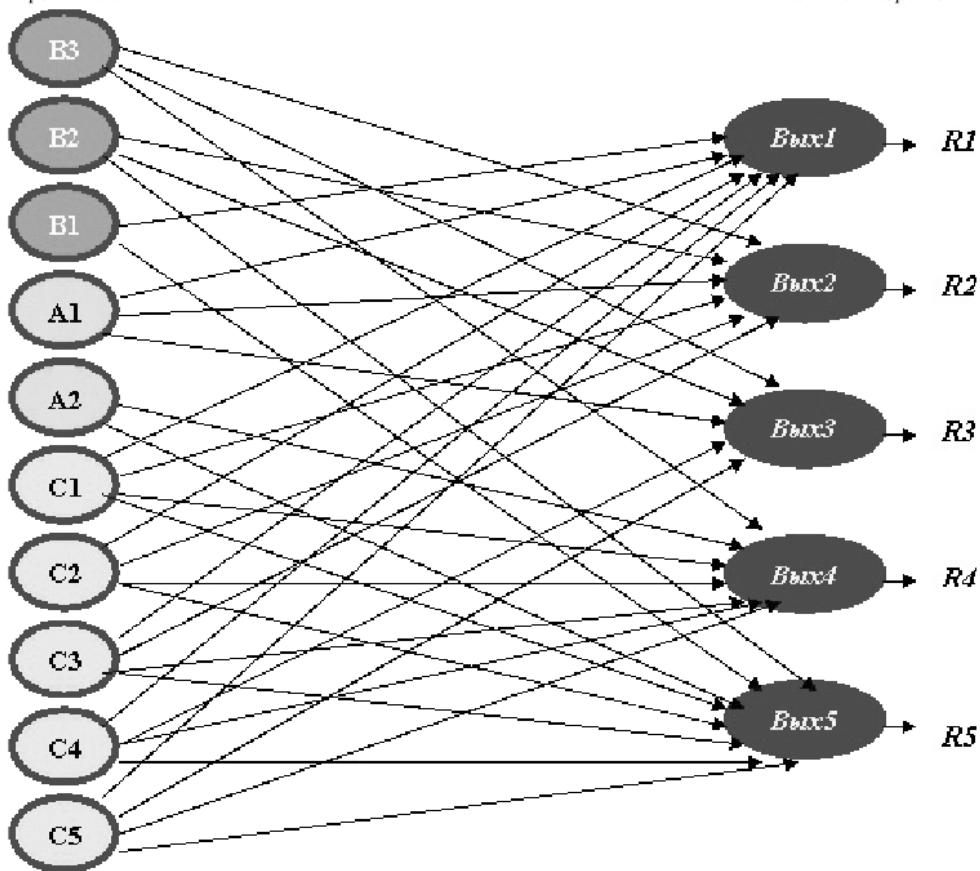


Рис. 3.8. Однослойная нейросеть

Возьмем ту же *передаточную функцию*, с теми же параметрами и рассчитаем те же примеры, отображенные в табл. 3.1. Составим для них табл. 3.2.

Данная *нейросеть* также оказывает предпочтение решению R5, хотя порядок убывания величин возбуждения выходного слоя отличен от ранее полученного. Предпочтительность решений R2 и R3 меняется местами.

Таблица 3.2. Примеры расчета решений по однослойной нейросети

Ситуация	Требуемое решение	$V_{\text{Вых1}}$	$V_{\text{Вых2}}$	$V_{\text{Вых3}}$	$V_{\text{Вых4}}$	$V_{\text{Вых5}}$
{A1, B1, C3}	R1	2,2	1,4	0,6	0,6	1,4

{A1, B2, C2}	R2	1,4	2,2	1,4	0,6	1,4
{A1, B3, C3}	R2	1,4	2,2	1,4	1,4	0,6
{A1, B2, C4}	R3	1,4	1,4	2,2	0,6	1,4
{A1, B3, C5}	R3	1,4	1,4	2,2	1,4	0,6
{A2, B3, C1}	R4	1,4	1,4	0,6	2,2	1,4
{A2, B1, C3}	R5	1,4	0,6	0	1,4	2,2
{A2, B2, C4}	R5	0,6	0,6	1,4	1,4	2,2
$V_{B1} = V_{B2} = 0.8, V_{A1} = V_{A2} = 0.5, V_{C1} = 0.7, V_{C5} = 0.8$?	2,04	1,4	0,84	1,4	2,68

3.5. Формализация нейросети

Тактирование работы сети, столь характерное для каждой управляющей системы, которая отслеживает состояние сети в дискретные моменты времени, определяет потактовое продвижение по ней волны возбуждений от входного слоя к выходному. Волна за волной, возбуждения имитируют *систолическую схему вычислений* - параллельный конвейер обработки отдельных кадров, соответствующих конфигурации возбуждений *входного слоя* в одном такте.

На практике широко исследуются многослойные сети типа *персептрон*, где отсутствуют обратные связи и связи возможны между нейронами только смежных слоев. В данном разделе мы также не рассматриваем обратные связи, но снимаем ограничение на "*слоистость*" нейросети, обеспечивая более общий подход. Именно такая *нейросеть*, допускающая связи "через слой", была построена в приведенном выше примере. Для таких сетей значительно упрощаются следующие построения.

Нейронную сеть можно изучать статически, исследуя ее структуру, и динамически, исследуя в ней прохождение возбуждений.

Статические исследования нейросети показывают, что она представляет собой ориентированный граф G без контуров. Вершины его соответствуют нейронам, дуги - *синаптическим связям*.

Целесообразно такой, высший уровень представления отделить от более глубокого описания каждого нейрона и связей между ними, отображающего динамику проходящих процессов, т.е. расчет значения возбуждения нейронов в зависимости от *весов синаптических связей* и порогов.

Граф малопригоден для формальных исследований и компьютерных алгоритмов. Удобнее пользоваться матричным отображением нейросети. Таким способом представления можно отобразить как структуру, конфигурацию, топологию графа, так и численные значения характеристик его *синаптических связей*.

Составим *матрицу следования* S (рис. 3.9), число строк (и столбцов) которой равно числу нейронов сети, включая нейроны входного и выходного слоя. Каждая строка (и столбец с тем же номером) соответствуют одному нейрону.

Элемент (i, j) этой матрицы - суть непустой объект, содержащий вес ω_{ij} синаптической связи $j \rightarrow i$, если такая связь существует. В то же время элементы матрицы S следует интерпретировать как булевы переменные, равные "1" в случае ненулевого значения указанного веса. Это разрешает выполнение логических операций над строками и столбцами матрицы S , рассматривая ее как аналогичную *матрицу следования*, используемую при описании частично упорядоченных множеств работ в параллельном программировании [6].

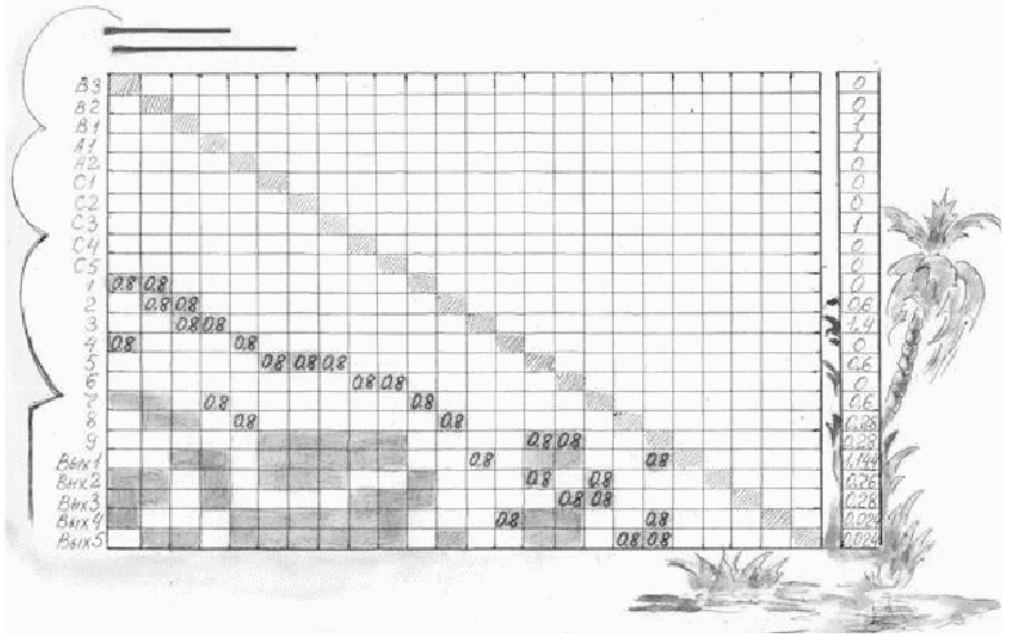


Рис. 3.9. Матрица следования с транзитивными связями

Матрицу S можно изучать в статическом режиме, исследуя и корректируя возможные пути прохождения возбуждений. По этой же матрице в динамическом режиме (моделирования) можно исследовать действительные пути прохождения возбуждений. Такое исследование связано с потактовым расчетом величин возбуждения нейронов.

Нулевые строки (входы) матрицы S соответствуют нейронам входного слоя - рецепторам, нулевые столбцы (выходы) - нейронам выходного слоя.

Нейроны образуют статические цепочки $\alpha_1 \rightarrow \alpha_2 \rightarrow \dots \rightarrow \alpha_m$, длины m , если существуют синапсические связи (впредь будем опускать слово "синапсические") вида $\alpha_k \rightarrow \alpha_{k+1}$. Составляя все статические цепочки по графу G , можно выделить одну или более цепочек максимальной длины. Такие цепочки назовем критическими. (В традиционных многослойных моделях длина критической цепочки совпадает с числом слоев сети.)

Пусть по статической цепочке $i \rightarrow j \rightarrow k$ мы обнаружили отсутствие непосредственной связи $i \rightarrow k$, т.е. элемент матрицы S , стоящий на

пересечении k -й строки и i -го столбца, равен "0". Однако очевидно опосредствованное влияние возбуждения нейрона i на величину возбуждения нейрона k , что указывает на наличие *транзитивной связи* $i \rightarrow k$. Нахождение *транзитивных связей* позволяет выявлять все нейроны, потенциально возбуждающие определенный нейрон выходного слоя.

Алгоритм дополнения матрицы S *транзитивными связями* весьма прост.

Для всех $i = 1, 2, \dots, N$:

Формируем новое значение строки i логическим сложением этой строки со строками, соответствующими непустым элементам в ней. "Новые" непустые элементы заменяем на не пустые элементы, обозначающие *транзитивные связи*.

В нашем примере матрица S , дополненная *транзитивными связями* (непустые клеточки), и представлена на рис. 3.9.

Зафиксируем некоторое подмножество R нейронов *входного слоя* и единственный нейрон r *выходного*. Построим множество цепочек, ведущих из выделенного подмножества нейронов *входного слоя* в данный нейрон *выходного слоя*. Выделенное таким образом множество цепочек назовем *статическим путем возбуждения* данного нейрона *выходного слоя* по множеству нейронов *входного слоя*. Обозначим его $R \rightarrow r$.

Представим алгоритм формирования *статического пути возбуждения* $R \rightarrow r$.

1. Вычеркиваем из *матрицы следования* S строки и столбцы, которые соответствуют нейронам, не принадлежащим R .
2. Вычеркиваем из *матрицы следования* S все строки и столбцы, отображающие нейроны *выходного слоя*, кроме нейрона r .
3. Вычеркиваем из *матрицы следования* S строки и столбцы, отображающие нулевые позиции строки нейрона r .
4. Вычеркиваем из *матрицы следования* S строки и столбцы, отображающие нейроны *внутренних слоев* нейросети, в том

случае, если в результате предыдущего вычеркивания эти строки оказались нулевыми.

5. Выполняем пункт 4, пока не прекратится образование нулевых строк.

На рис. 3.10 матрицей $S[B1, A1, C1 \rightarrow \text{Вых1}]$ отображено множество цепочек, ведущих из нейронов $B1, A1, C1$ в нейрон Вых1 , т.е. статический путь возбуждения $[B1, A1, C1] \rightarrow \text{Вых1}$.

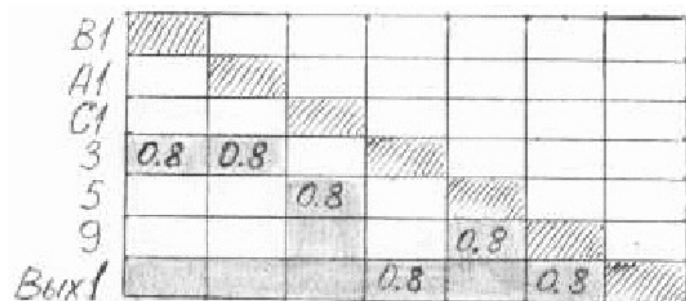


Рис. 3.10. Статический путь возбуждения

Теперь рассмотрим фактор возбуждения и проанализируем возможные динамические пути прохождения возбуждений.

Динамической цепочкой возбуждений $\beta_1 \rightarrow \beta_2 \rightarrow \dots \rightarrow \beta_m$ длины m назовем статическую цепочку, составленную из нейронов, первые $m-1$ которых обладают отличной от нуля величиной возбуждения.

Динамическая цепочка возбуждений является вырожденной, если ее последний элемент обладает нулевой величиной возбуждения. Вырожденная динамическая цепочка возбуждений характеризует затухание сигнала.

Предъявим сети некоторый образ или эталон (не будет большой ошибки, если и здесь при предъявлении мы будем пользоваться словом "образ", ибо все есть образ чего-то), приводящий к возбуждению нейроны входного слоя, - в некоторой конфигурации. Эти нейроны образуют множество R . В результате возбуждения сети оказался возбужденным некоторый нейрон r выходного слоя. Сформировался, таким образом,

динамический путь возбуждения $R \rightarrow r$; очевидно, его образуют нейроны, входящие в состав *статического пути возбуждения* $R \rightarrow r$, так как некоторые нейроны могут остаться невозбужденными.

В общем случае не все нейроны, составляющие *статический путь возбуждения*, "работают" на возбуждение нейрона выходного слоя, поскольку возможны вырожденные *динамические цепочки* возбуждения, обусловленные значениями *порогов передаточной функции*.

3.6. Модель механизма запоминания

Главный механизм запоминания, реализованный в природе, можно представить так. Импульс возбуждения, проходя через синапс, "нагревает" и уменьшает его сопротивление, увеличивая синаптический вес. В последующих тактах, при последующих предъявлениях эталона, импульс возбуждения увереннее одолевает путь возбуждения, с большей определенностью указывая соответствующий образ (как говорится, "Сложилась связь"), а используемые при этом *синапсы*, "подогреваясь", сохраняют и, возможно, увеличивают вес.

Здесь работает известное правило Хебба [19, 20]: Синаптический вес связи двух возбужденных нейронов увеличивается.

Таким способом даже достигается эффект локализации и максимизации возбуждения на выходном слое, дублирующий, а возможно, исключаящий необходимость взаимодействия соседних нейронов.

По-видимому, *синапсы* обладают свойством "остывания" со временем, если нет подтверждения их использования. Такое предположение адекватно свойству нашей памяти: ненужная, не подтверждаемая и не используемая периодически информация стирается ("Связи рвутся!"). Стирается до такой степени, что приходится учиться заново.

Отметим и важную роль воображения: эталоны на входном слое поддерживаются достаточно долго, возобновляются или моделируются. По-видимому, здесь большое значение имеет эпифиз, "третий глаз" - орган воображения и медитации, память и генератор видений.

При создании искусственных механизмов обучения нейросети

возникают вопросы:

- Увеличивать ли веса всем нейронам, образующим *статический путь возбуждения* - для запоминания эталона?
- Увеличивать ли веса только вдоль некоторых (опорных) цепочек *статического пути возбуждения*?
- Увеличивать ли веса только нейронов, образующих *динамический путь возбуждений* ; ведь выше предполагалось, что уже само возбуждение увеличивает вес связи? Зачем возбуждать дополнительные нейроны, если и этих достаточно?

По-видимому, третий аспект в большей степени соответствует самообучению, самонастройке. Вмешательство в наш мозг на этом уровне исключено. Однако система искусственного интеллекта - в более выгодном положении. Ведь она находится под нашим неусыпным контролем, реализуя обучение "с учителем", и допускает любое вторжение, корректирующее вынужденные недостатки естественного интеллекта. Поэтому, рассматривая пример (и пытаюсь накопить хоть какой-то опыт), мы будем увеличивать *веса синаптических связей* нейронов, составляющих выделенные цепочки *статического пути возбуждения* от эталона к образу. Назовем такой метод методом опорных путей, который выработан в процессе проведения многочисленных экспериментов. В частности, увеличение синаптических весов большого числа нейронов приводило к неудачам, связанным с корреляцией динамических цепочек возбуждения для разных эталонов и с быстрым насыщением нейросети (недостаток многих известных алгоритмов обучения). В процессе обучения многим эталонам рано или поздно все веса сети оказывались повышенными, и она прекращала что-либо различать. Начиналась путаница согласно выражению "Ум за разум заходит!"

Что же касается величины изменения синаптических весов, то вряд ли необходимо относительно каждого нейрона решать системы дифференциальных уравнений в частных производных. (Речь идет о методе обратного распространения ошибки, подробно изложенном в [20].) Природа больше рассчитывает на авось, "на глазок" и другие мудрые и практические приблизительные ориентиры, малопривлекательные теоретически. В данном случае нас более обнадеживает "прилив крови" в нужном направлении, стимулируемый

информационным раздражителем или легким подзатыльником. В связи с этим введем некую переменную $\delta\omega$, выверяемую практически (а то и просто единицу), уменьшаемую или увеличиваемую при необходимости.

Таким образом, нам удалось перейти от схемотехники к *нейротехнологии*, от точного, определенного к приближительному, неопределенному. Действительно, не могла эволюция, основанная на принципе "делай все, что можно", методом "проб и ошибок" привести к производству точных электронных схем, упрятанных в черепную коробку. Да и нет в природе точной, абсолютно достоверной информации. Но общие логические принципы и зависимости должны быть воплощены и в той и в другой технологии.

3.7. Применение "готовых" нейросетей

Справедливо желание построения универсальных моделей нейросетей в составе программного обеспечения компьютера, снабженных механизмами приспособления под задачу пользователя. Еще более справедливо желание построить набор аппаратных средств - нейросетей (*нейрокомпьютеров*, НК), сопряженных с компьютером и, по выбору пользователя, участвующих в решении сложных задач. Такие аппаратно реализованные нейросети, как приставки или внешние устройства компьютера, например, определяют специальное направление использования ПЛИС - интегральных схем с программируемой логикой.

3.7.1. Однослойная нейросеть

Для наглядности воспользуемся как матричным, удобным алгоритмически, так и графическим представлением нейросети.

Однослойная *нейросеть*, составленная по принципу "каждый с каждым", представлена на рис. 3.11.

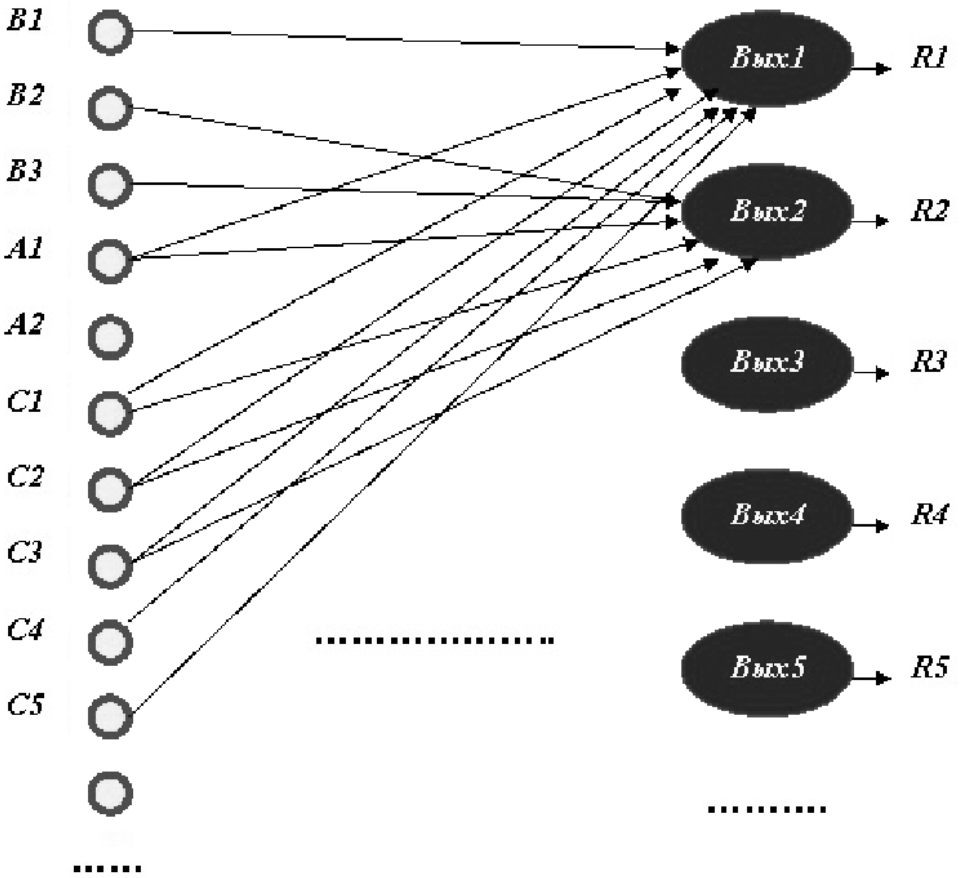


Рис. 3.11. Формирование однослойной нейросети

Пусть используется предложенная выше *передаточная функция*

$$V := \xi\left(\sum_{j=1}^m V_j \omega_{ij} - h_i\right)$$

Значения ω_{ij} предстоит подобрать, а значения h_i положим равными нулю.

Закрепим 10 нейронов *входного слоя* за исходными данными, 5 выходов - за решениями. Этим мы выделим интересующую нас подсеть, которой соответствует *матрица следования* на рис. 3.12.

<i>B1</i>	■													
<i>B2</i>		■												
<i>B3</i>			■											
<i>A1</i>				■										
<i>A2</i>					■									
<i>C1</i>						■								
<i>C2</i>							■							
<i>C3</i>								■						
<i>C4</i>									■					
<i>C5</i>										■				
<i>Вых1</i>	1			1		1	1	1	1	1	1	1	■	
<i>Вых2</i>		1	1	1		1	1	1					■	
<i>Вых3</i>		1	1	1					1	1			■	
<i>Вых4</i>			1		1	1	1	1	1	1				■
<i>Вых5</i>	1	1			1	1	1	1	1	1				■

Рис. 3.12. Матрица следования для однослойной нейросети

Здесь отображен ее окончательный вид, т.к. сначала все веса связей принимаются равными нулю.

Для того чтобы сформировать решение $R1$ на нейроне $Вых1$, надо значительно увеличить веса связей этого нейрона с нейронами $B1$, $A1$, $C1$, $C2$, $C3$, $C4$, $C5$, т.е. построить *статический путь возбуждения* $[B1, A1, C1, C2, C3, C4, C5] \rightarrow Вых1$ и вдоль него по некоторой дисциплине увеличить веса связей.

В данном случае результат очевиден, поэтому обратим внимание на некоторый общий подход.

Установим веса связей между нейронами $B1$, $A1$, $C1$, $C2$, $C3$, $C4$, $C5$ с одной стороны, и нейроном $Вых1$ - с другой равными единице, оставив нулевыми веса связей этого нейрона с другими

нейронами входного слоя. Таким образом полностью исключается влияние других нейронов входного слоя на данный выходной нейрон. Конкретная задача может потребовать корректировки, учета взаимного влияния всех входных ситуаций в результате тщательного экспериментального исследования задачи.

Здесь вновь прослеживается преимущество нейросети, способной элементарно просто учитывать наблюдаемые или интуитивно предполагаемые поправки, требующие огромных исследований и расчетов.

Поступив так же со всеми выделенными нейронами выходного слоя, получим окончательный вид матрицы следования (рис. 3.12). Построенная нейросеть полностью соответствует специальной сети "под задачу", представленной на рис. 3.7.

...Так какую же сеть предложить дяде Рамзаю? Ведь надо и подходчивее, и попрacticalнее, но и так, чтобы не казалось уж слишком просто.

3.7.2. Нейросеть произвольной структуры

Предположим, мы располагаем некоторым банком "красиво" изображенных графических схем, которые можно положить в основу структуры нейросети. Понравившуюся структуру мы решаем интерпретировать как нейросеть, дополнив ее передаточной функцией и обучив решению задачи, поставленной дядей Рамзаем.

Пусть выбранная нами нейросеть имеет 12 входов (более чем достаточно), 5 выходов и реализует ту же передаточную функцию с начальными значениями весов $\omega_{ij} = 0$ и порога $h = 0$.

Однако сеть обладает специфической топологией, затрудняющей ее обучение. Сеть многослойная, что исключает связи "через слой", присутствующие, например, на рис. 3.7 как результат построения нейросети "под задачу". Да и связь между слоями скорее соответствует известному предупреждению "шаг влево, шаг вправо ...".

Будем использовать метод опорных путей, или трассировки, расширяющий использованный выше "схемотехнический" подход. Его можно изобразить схемой рис. 3.2 → рис. 3.5 → рис. 3.7 → рис. 3.8. Суть метода - в трассировке соединений, в назначении высоких весов некоторых связей, превращающих *нейросеть* в законченное функциональное устройство. При такой трассировке обучение производится на эталонах в полном смысле этого слова, на вполне определенных (достоверных) ситуациях, например, на отсутствии (0) события или на его наступлении (1). После обучения сеть должна выдавать наиболее близкое решение при недостоверной информации, т.е. согласно вероятности наступления того или иного события. Хотя, как показано на примерах, можно выйти и за рамки теории вероятности, не требуя полноты множества событий и условий нормировки, но взвешивая события на основе каких-то других принципов.

Итак, для успешной, наглядной и легко рассчитываемой трассировки решим вопрос кардинально: какие веса полагать равными нулю, а какие - единице? Все прочие возможности, например $\min \omega_{ij} = 0, 1$, введение порога $h = 0, 5$ и т.д., будут способствовать более плавной работе сети, непрерывности перехода из состояния в состояние.

Применив принципы *комбинаторики* и эвристики, выполним трассировку нейросети (рис. 3.13). На рисунке наглядно показано, какая *нейросеть* была предоставлена, насколько она "неповоротлива", скажем, по сравнению с однослойной "каждый с каждым".

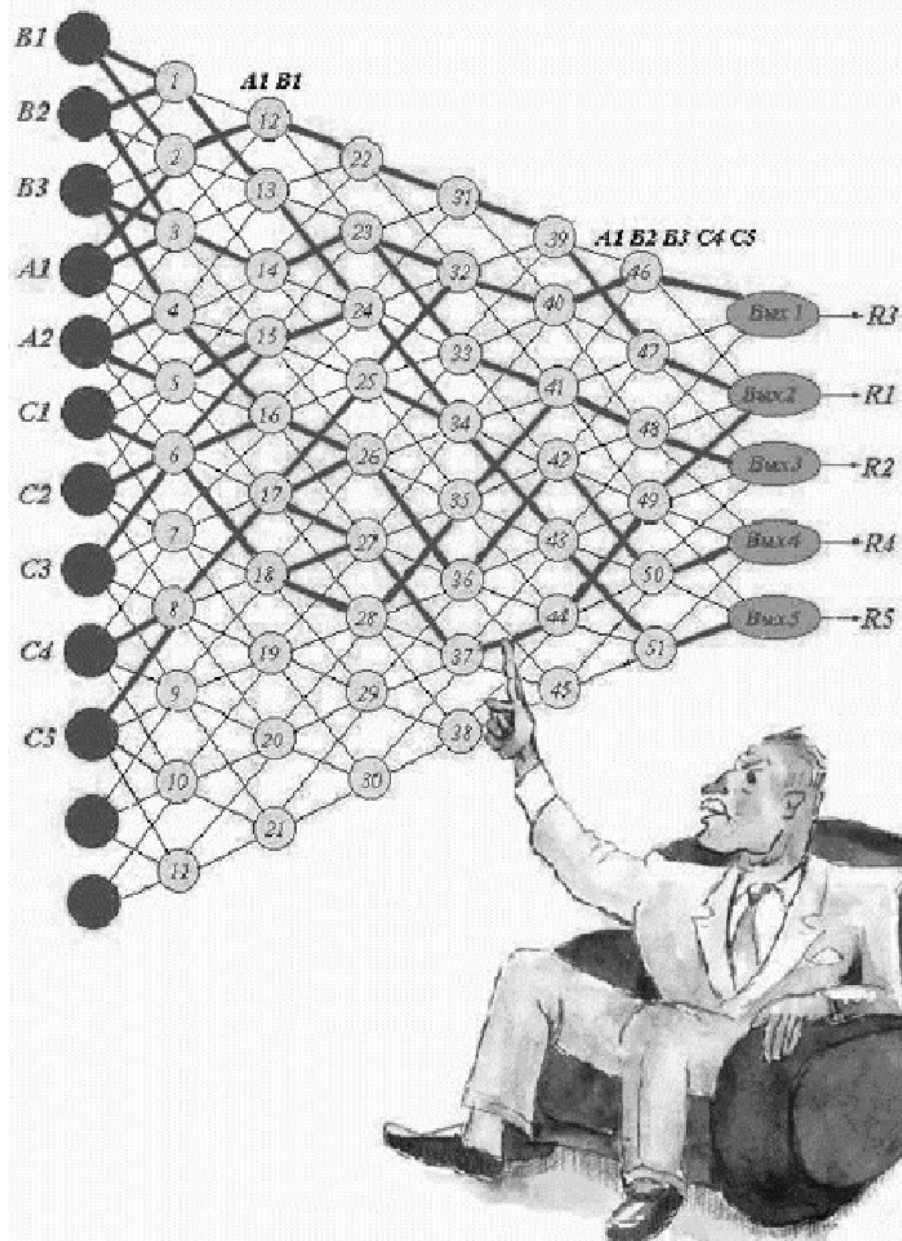


Рис. 3.13. Результат трассировки многослойной нейросети

Далее будет представлен формальный алгоритм трассировки. Однако чтобы настроиться на его понимание, следует проанализировать свои действия и сделать следующие выводы:

1. Мы анализировали слой за слоем, постоянно помня цель - пять комбинаций ситуаций, каждая из которых должна возбудить один из нейронов выходного слоя, не вводя пока жесткое закрепление решений за этими нейронами.
2. В каждом слое мы собирали частную комбинацию - терм, который можно использовать в последующем, - из доступных термов предыдущего слоя.
3. Термы, которые пока не могут быть использованы при конструировании из-за их взаимной удаленности, мы запоминали без изменения на анализируемом слое, пытаясь их "подтянуть" в направлении возможного дальнейшего объединения.
4. Мы старались не "тянуть" термы "поперек" всей сети. В противном случае мы сталкивались бы с проблемой: как избежать пересечений и искажения уже сформированных термов. Все это заставило нас долго не закреплять нейроны выходного слоя за решениями, что в конце концов привело к нарушению естественного порядка следования решений.

Для автоматизации трассировки необходимо матричное представление, только и доступное компьютеру.

Матрица следования, отражающая трассировку нейросети, получается на основе [рис. 3.13](#), если отметить элементы, соответствующие "тонким" линиям, нулевыми весами, а элементы, соответствующие "жирным" линиям, - весами, равными единице.

На [рис. 3.14](#) отражен *динамический путь возбуждения*, приводящий к решению R1. Он строится по алгоритму, изложенному в [разд. 3.5](#). В данном случае *динамические пути возбуждения* совпадают со статическими. В общем случае из *статического пути возбуждения* необходимо исключить нейроны, которые не входят во *входной слой* и которым соответствуют нулевые строки сформированной *матрицы следования*.

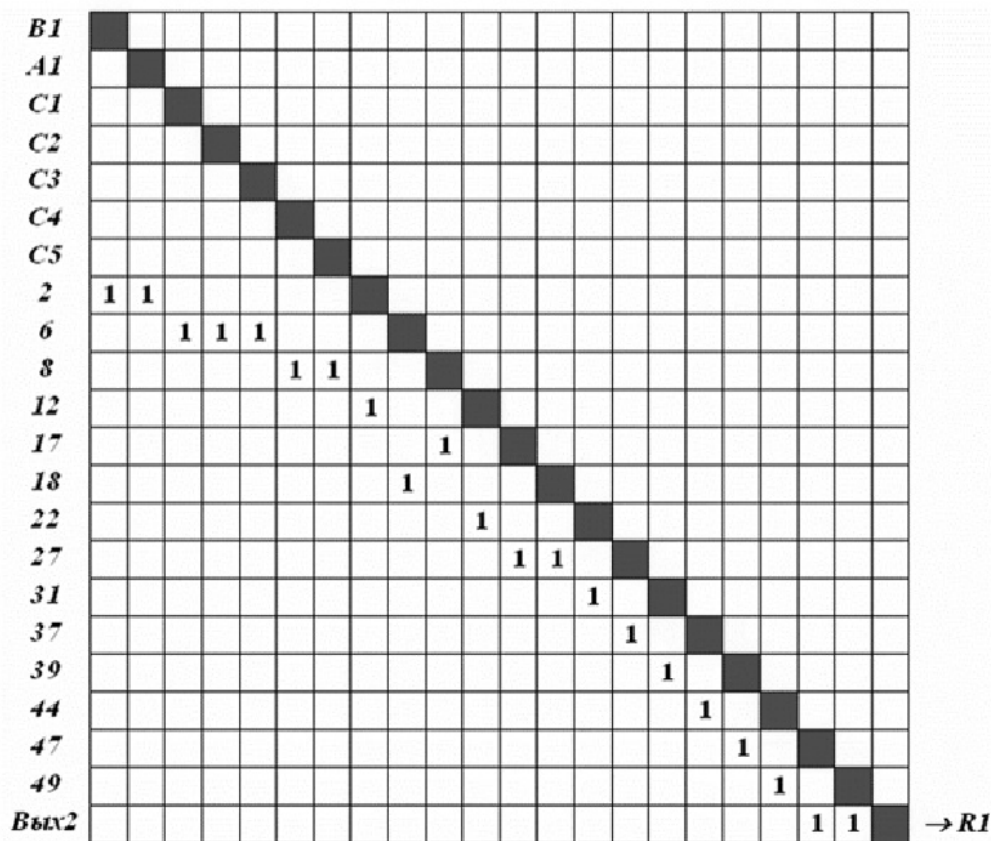


Рис. 3.14. Динамический путь возбуждения

Аналогично получают *динамические пути возбуждения*, приводящие к другим решениям.

Поставим теперь задачу дальнейших исследований: как построить все необходимые *пути возбуждения* так, чтобы они, возможно, пересекались, - но только для формирования общих термов? А способна ли выбранная нами "готовая" *нейросеть* вообще справиться с поставленной задачей или предпочтительнее принцип "*нейросеть* под задачу"?

... Вот теперь-то мы довольны! Мы снабдили дядю Рамзя универсальной обучаемой *нейросетью*. Теперь, если он, раскаявшись, вновь возлюбит свою благороднейшую профессию, он найдет ей (сети) достойное применение в водворении Васи, Пети & К0 на то спальное

место, которого они действительно заслуживают.

3.8. Нейросетевые технологии и нейрокомпьютеры

Итак, располагая знаниями о ситуациях на входе управляющей системы и реакциях на каждую из них, получаем таблицу. В одном столбце, так называемом запросном поле, - вектор-ситуация, в другом, *ответном поле* - принимаемое решение. Такая идея *ситуационного управления* высказана Д.А. Поспеловым [23] еще в 1970-х годах.

Для ее воплощения можно применить технологии построения ассоциативной памяти, столь широко используемой в современных вычислительных системах. Если между ситуациями ввести операции отношения, то можно построить операции вида "найти ближайшую величину слева (справа)", "найти ближайшие включающие границы" и т.д. Идея ассоциативной ЭВМ давно реализована, например в STARAN [15] (1977 г.).

На основе вышесказанного строится *самообучающаяся система*, в режиме обучения дополняющая базу знаний (ту самую таблицу в ассоциативной памяти) с помощью моделирования ситуаций и принимаемых решений и выдающая решение в рабочем режиме согласно таблице на основе интерполяции.

Однако необходимость выхода на большие размерности ограничивает возможность построения "большой" ассоциативной памяти по существующей технологии. Ведь такая память, даже при реализации единственной операции, основана на одновременном сравнении входного вектора с запросными частями всех ее регистров и с выдачей содержимого ответного поля в случае совпадения.

Выходит, что мозг не может непосредственно хранить таблицы, а моделирует их с помощью нейросетевых механизмов? Можно вспомнить, что исходные предложения (например, Кохонена) по применению нейросетей касались именно ассоциативной памяти.

Зачем же нам следовать столь неловкому воплощению?

Ответ может быть лишь таким: все хорошо к месту. Просто более

тщательно следует определять области разумного применения каждого из различных средств решения задач искусственного интеллекта. И как можно раньше уйти от опасных для здоровья вопросов философски-мистического характера.

Хотя и следует вновь заглянуть "в зеркало".

Где мы используем принцип нейросети непосредственно, а не косвенно, - посредством расчета, анализа баз знаний и т.д.? По-видимому, обязательно там, где мы получаем первичную информацию для последующих выводов: органами зрения, слуха, обоняния, осязания. На этом же уровне мы способны провести первичную классификацию и принять оперативное решение: убежать от стремительно приближающегося автомобиля, надеть противогаз и т.д.

Значит, в нашей жизнедеятельности, требующей разнообразного проявления, существует такая ниша, эффективная на самом низком уровне животного состояния, где решение должно быть сверхоперативным, скорее - рефлекторным, не допускающим анализа. Этому способствует высокий параллелизм сети. Именно высокий параллелизм, наряду с исключением сложных расчетов, обусловил взрыв интереса к системам искусственного интеллекта в начале 1980-х годов, когда остро встала задача разработки вычислительных средств сверхвысокой производительности.

И этим мы вновь затронули важный вопрос актуальности аппаратной реализации нейросети или *нейрокомпьютеров*, т.к. программная модель на непараллельном компьютере лишена свойства высокого параллелизма мозга и ограничивает выход на "большие" нейросети.

Этот параллелизм выражается в том, что одновременно обрабатывается большое число цепочек нейронов. При этом каждый нейрон обрабатывается хотя и по одному алгоритму, но - по разным его ветвям: один, в конце концов, возбудится, другой нет; связи нейрона индивидуальны и изменяются не идентично связям других нейронов и т.д.

Тогда, ставя задачу разработки параллельного вычислительного устройства - нейрокомпьютера, способного имитировать работу нейросети с учетом ее достоинств по реализации высокой

производительности, следует учесть, что:

- необходимо распределять нейроны (точнее, нейроподобные элементы или соответствующие им программные процедуры) между процессорами НК, синхронизируя обработку нейронов в соответствии с *матрицей следования*. Т.е. необходимо реализовать способ распараллеливания по информации;
- одинаковые на всех процессорах программы одновременно обрабатываемых нейронов в общем случае должны выполняться по разным ветвям.

При программной реализации нейросети перечисленные требования соответствуют SPMD-технологии ("одна программа - много потоков данных") [9], привлекательность которой обоснована для многих приложений параллельного решения задач высокой сложности.

Привлекательна реализация "большой" нейросети на основе сетевых технологий.

При аппаратной реализации НК (или его аппаратной поддержке) также необходимо учесть следующее требование: один нейроподобный элемент должен делить время между имитацией многих нейронов. Жесткая аппаратная имитация нейросети, соответствующая связи "один нейроподобный элемент - один нейрон", неэффективна, т.к. ограничивает возможную размерность моделируемой сети.

Учитывая специализацию нейрокомпьютера при применении сетевых технологий в рамках построения более сложных управляющих систем, целесообразно, чтобы НК использовался как сопроцессор под управлением мощного и универсального компьютера-монитора. Это же обусловлено разнообразными функциями "учителя" по формированию, заданию и регулировке параметров, по обучению и по дальнейшему применению результатов. В рамках современных компьютерных технологий НК должен дополнять персональный компьютер как его внешнее устройство и "врезаться" в существующие ОС.

Итак, мы знаем языки логического вывода, экспертные системы,

самообучающиеся системы управления. Но в природе первично воплощены лишь нейросети!

Почему же мы, располагая столь богатым арсеналом средств искусственного интеллекта, в основе своей (как нам кажется!) опирающихся далеко не на одни только нейросети, самодовольно недоумеваем, отчего они не были первоначально заданы Природой, а явились плодом нашей творческой гениальности? И мы кощунственно вопрошаем:

- Ты что, Господи, не мог додуматься до этого Сам?

И посылается нам догадка с Неба: о Великой Целесообразности, о непрерывности и преемственности Развития, о предлагаемом Базисе, требующем Надстройки, о вечной Причинно-Следственной Вытекаемости... Мозг - конструктивно целесообразен, развиваем, универсален, самодостаточен. Мозг - основа, на которой еще не то можно построить!

Здесь можно привести пример колеса, не существующего в живых организмах. Какова должна быть длина нерва, чувствующего обод колеса? Значит, колесо должно быть отделено от живого существа! Но не рожать же матери отдельно ребенка, отдельно - колесо к нему. И сказал Господь Человеку:

- Я дам тебе разум, а колесо ты сделаешь сам . . .

Нет, додумался Он, Высший Разум, - посредством средства, вложенного в нас . . . Мы сыграли роль слепых исполнителей предназначенного, предопределенного, предсказанного . . . Словно микробы в желудочно-кишечном тракте - в заблуждении о личном счастье и сытости, - в действительности мы работаем на Волю Создателя, помогая Ему в главной и глобальной установке на Развитие.

Трассировка нейронной сети

Приводится простейший алгоритм трассировки логической нейронной сети – построения логических цепочек связей, реализующих описание системы принятия решений. На сети произвольной структуры воссоздается однослойная нейронная сеть, приводящая возбуждения рецепторов, соответствующих заданию обобщенных эталонов, к возбуждению нейронов выходного слоя, соответствующих заданным решениям.

" - Аннушка . . . Аннушка?.. - забормотал поэт, тревожно озираясь, -
позвольте, позвольте . . .

К слову "Аннушка" привязались слова "подсолнечное масло", а затем почему-то "Понтий Пилат". Пилата поэт отринул и стал вязать цепочку, начиная со слова "Аннушка". И цепочка эта связалась очень быстро и тотчас привела к сумасшедшему профессору".

М.А. Булгаков. "Мастер и Маргарита"

4.1. Пример - опыт - предпосылки обобщения

Объединим множество эталонов, требующих возбуждения одного нейрона выходного слоя, понятием "*обобщенный эталон*". Например, эталоны $A1 \wedge B1 \wedge C1$, $A1 \wedge B1 \wedge C2$, $A1 \wedge B1 \wedge C3$, $A1 \wedge B1 \wedge C4$, $A1 \wedge B1 \wedge C5$ образуют один *обобщенный эталон* $A1 \wedge B1 \wedge (C1 \vee C2 \vee C3 \vee C4 \vee C5)$, ведущий к R1. Тогда для *обобщенного эталона* введем понятие обобщенной операции &, которая объединяет все посылки, ведущие к одному выводу, $A1 \& B1 \& C1 \& C2 \& C3 \& C4 \& C5 \rightarrow R1$.

Выберем нейросеть, отражающую, как нам кажется, все особенности, которые необходимо учесть при построении алгоритма трассировки, и, как мы уже начали, возьмем тот же пример, который выглядит теперь следующим образом:

$$\begin{aligned}A1 \& B1 \& C1 \& C2 \& C3 \& C4 \& C5 \rightarrow R1 \\A1 \& B2 \& B3 \& C1 \& C2 \& C3 \rightarrow R2 \\A1 \& B2 \& B3 \& C4 \& C5 \rightarrow R3 \\A2 \& B3 \& C1 \& C2 \& C3 \& C4 \& C5 \rightarrow R4 \\A2 \& B1 \& B2 \& C1 \& C2 \& C3 \& C4 \& C5 \rightarrow R5\end{aligned}\tag{4.1}$$

Мы хотим, задавая *обобщенный эталон* на входе, например из тех, что приведены выше, экономным путем построить, возможно, пересекающиеся *пути возбуждения*, ведущие от каждого возбужденного *входного нейрона* в соответствующий нейрон выходного слоя. Способ должен быть экономным в том смысле, что пути должны быть максимально объединены, совмещены, реализуя стратегию слияния и выхода на общую, единую дорогу, ведущую к заветной цели.

При этом мы строим эвристический алгоритм, который крайне экономно расходует ресурсы сети - ее нейроны и связи, но и так же экономно вводит изменения, развивает сеть, если связей оказывается недостаточно.

Ибо мы видели, с каким трудом удается приспособить "готовую" нейросеть под конкретную задачу. Мы постоянно склонялись к вопросу: "А не лучше ли сразу строить сеть "под задачу", а не наоборот?" Нам кажется, мы нашли диалектический компромисс: беря что-то близкое, подходящее, мы минимально переделываем, словно костюм с жмуцим гольфиком, купленный в универсальном магазине. Кроме того, нам очень важно сделать алгоритм доступным широким массам студенческой молодежи, овладевшей основами таинства параллельного программирования!

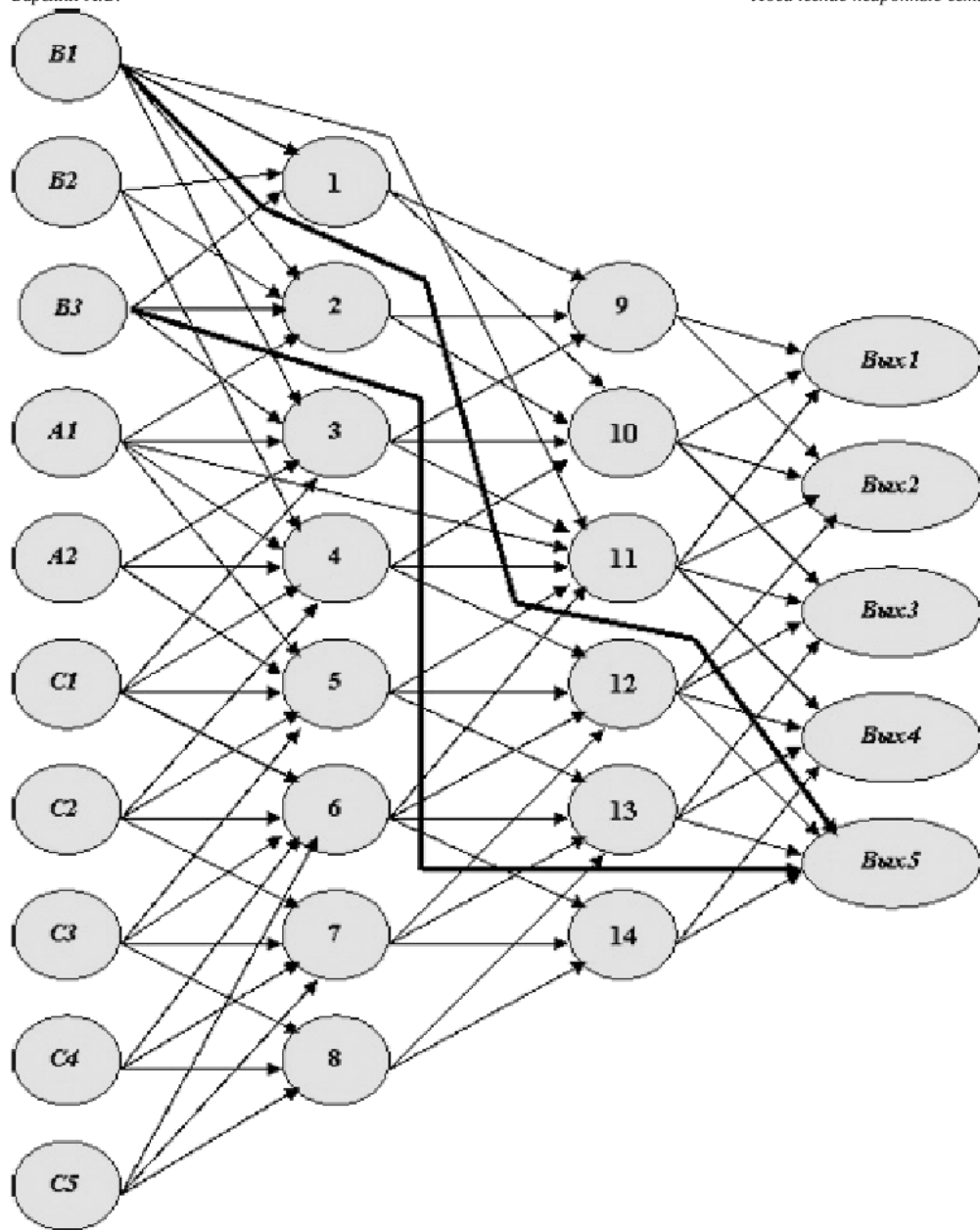


Рис. 4.1. Нейросеть, подлежащая обучению

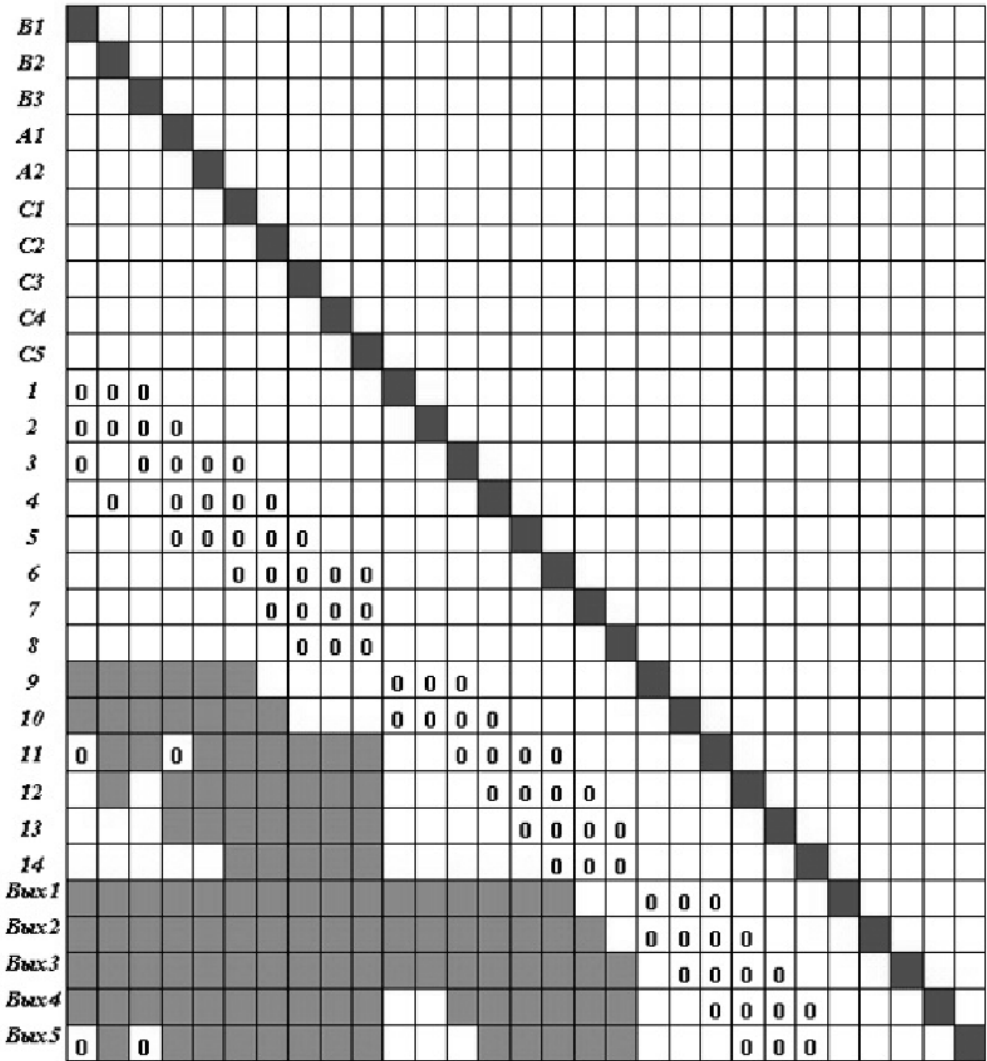


Рис. 4.2. Матрица следования с транзитивными связями

Пусть выбранная нейросеть представлена на рис. 4.1, где первоначально заданные связи обозначены тонкими стрелками. Строим соответствующую *матрицу следования* S и в ней - *транзитивные связи* (рис. 4.2). Проверяем, все ли нейроны выходного слоя достижимы из любого нейрона входного слоя. Если нет, считаем, что сеть составлена не корректно.

В данном случае из нейрона $B1$ не исходит ни одной *статической*

цепочки, заканчивающейся нейроном Вых5. Это следует из того, что в строке, соответствующей нейрону Вых5, нет даже транзитивной связи в столбце, соответствующем нейрону В1. То же касается нейрона В3.

Введем синаптические связи В1 -> Вых5 и В3 -> Вых5, что отмечено на рис. 4.1 жирными стрелками, а на рис. 4.2 – нулевыми весами введенных связей.

Сформируем статический путь возбуждения {В1, А1, С1, С2, С3, С4, С5} -> Вых1 и отобразим его матрицей S₁[В1, А1, С1, С2, С3, С4, С5 -> Вых1] на рис. 4.3.

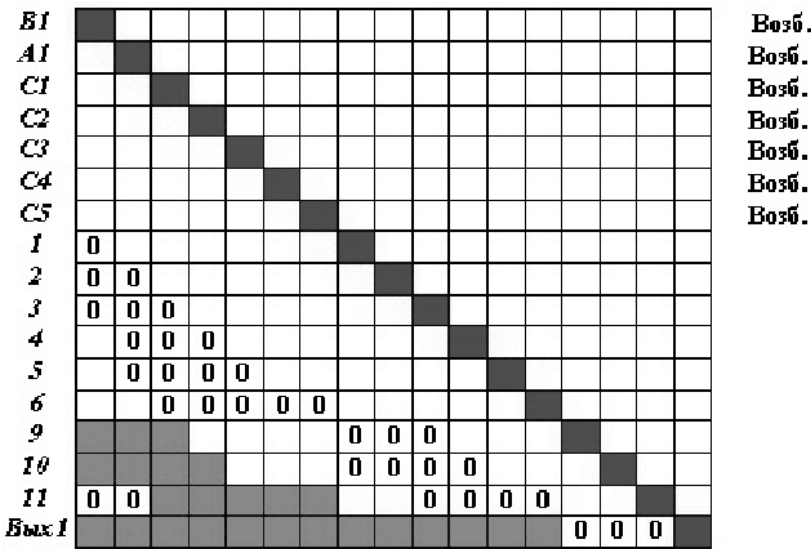


Рис. 4.3. Матрица следования при обучении первому эталону

Пытаясь симитировать прокладываемые пути возбуждений, снабдим все нейроны входного слоя, отраженные в этой матрице, признаком "возбужден".

Найдем входы матрицы S₁ (пустые строки), обладающие признаком "возбужден". Выделим в одну совокупность соответствующие им столбцы, в данном случае столбцы 1-7, и найдем в ней строку, содержащую максимальное число "нулей", - строку, соответствующую нейрону 6. Заменяем в этой строке (в выделенной совокупности

столбцов) "нули" "единицами", т.е. максимально увеличиваем соответствующие веса. Присваиваем нейрону 6 признак "возбужден", а также значение $m_6 = 5$, характеризующее то количество нейронов, "возбуждение" которых мы "собрали" на входе этого нейрона. Впредь мы можем только повторно использовать этот нейрон, но менять веса связей на его входе мы не должны. Изменение весов отображаем в матрице S .

Исключаем из матрицы S_1 строки и столбцы, соответствующие нейронам $C1, C2, C3, C4, C5$. Новый вид матрицы S_1 представлен на рис. 4.4.

B1																					Возб.
A1																					Возб.
1	0																				
2	0	0																			
3	0	0																			
4		0																			
5		0																			
6																					Возб.
9																					
10																					
11	0	0																			
Вых 1																					

Рис. 4.4. Шаг преобразования матрицы следования

Вновь выделяем столбцы, соответствующие "возбужденным" входам матрицы. Это столбцы, соответствующие нейронам $B1, A1, 6$. Находим строку в выделенной совокупности столбцов, имеющую максимальное число "нулей". Такая строка соответствует нейрону 11. Заменяем в этой строке (в выделенной совокупности столбцов) "нули" "единицами", т.е. максимально увеличиваем веса. Присваиваем нейрону 11 признак "возбужден" и значение $m_{11} = 3$. Отражаем изменение весов в матрице S .

Исключаем из матрицы S_1 строки и столбцы, соответствующие нейронам $B1, A1, 6$. Матрица принимает вид, изображенный на рис. 4.5.

1	■								
2		■							
3			■						
4				■					
5					■				
9	0	0	0			■			
10	0	0	0	0			■		
11			0	0	0			■	
Вых1	■	■	■	■	■	0	0	0	■

Возб.

Рис. 4.5. Шаг преобразования матрицы следования

Исключаем из матрицы все входы, которые соответствуют нейронам 1, 2, 3, 4, 5, не обладающим признаком "возбужден". Ведь строящийся нами *путь возбуждения* уже миновал эти нейроны! Исключение этих нейронов породит новые нейроны, нуждающиеся в исключении по той же причине, - нейроны 9 и 10.

Последовательное исключение сверху вниз динамически учитывает появление таких нейронов и гарантирует полное исключение за один проход. Матрица S_1 примет вид на рис. 4.6.

11	■	■	Возб.
Вых1	0	■	

Рис. 4.6. Шаг преобразования матрицы следования

Теперь в первом столбце имеется единственная строка с "нулем" - соответствующая нейрону Вых1. Меняем "нуль" на "единицу" - получаем окончательный возможный *динамический путь возбуждения* по заданному эталону. Изменения отражаем в матрице S .

Мы не будем приводить новый вид *матрицы следования* S , а на изображении самой сети (рис. 4.7) выделим темным построенную трассу (она же - опорный путь, а также *динамический путь возбуждения* по предлагаемому эталону).

Реакции на эталон одной комбинации мы сеть научили.

Составим *обобщенный эталон* ситуации, требующий решения R2, - {A1, B2, B3, C1, C2, C3}. Свяжем это решение с нейроном Вых2.

По сети на рис. 4.7 или по матрице на рис. 4.2 (с учетом частичного обучения!) построим (рис. 4.8) матрицу *статического пути возбуждения* $S_2[B2, B3, A1, C1, C2, C3 \rightarrow \text{Вых}2]$.

Нейрон 6 ранее "объединил" возбуждение пяти нейронов: C1, C2, C3, C4, C5, т.е. в матрице S в соответствующей строке содержится пять единиц и $m_6 = 5$. Однако в составленной матрице S_2 в строке, соответствующей этому нейрону, присутствуют лишь три единицы. Значит, этот нейрон не может быть использован для трассировки *необходимого пути возбуждения*, т.к. комбинация C1&C2&C3&C4&C5 нам здесь не нужна.

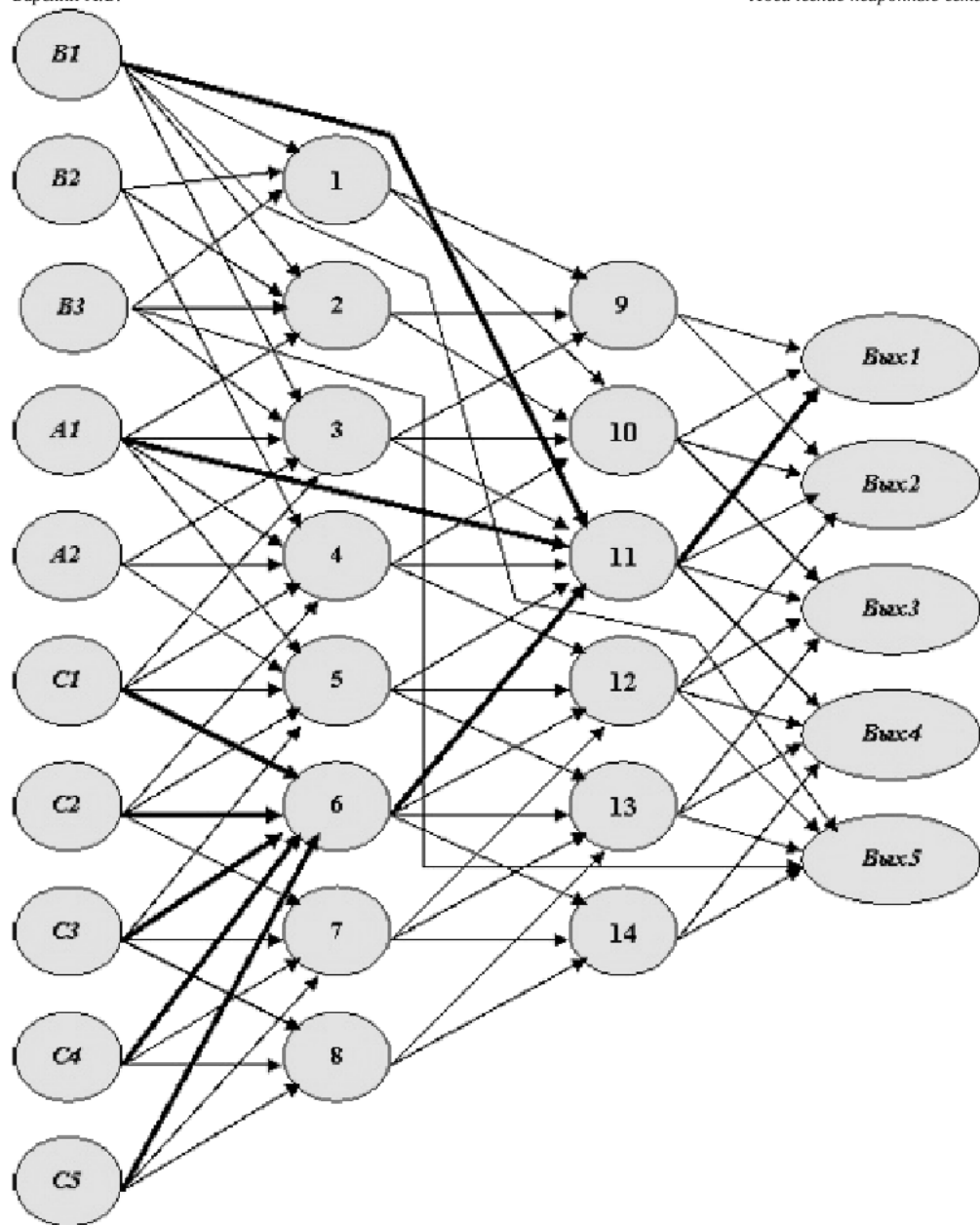


Рис. 4.7. Нейросеть, обученная первому эталону

<i>B2</i>	■																				Возб.
<i>B3</i>		■																			Возб.
<i>A1</i>			■																		Возб.
<i>C1</i>				■																	Возб.
<i>C2</i>					■																Возб.
<i>C3</i>						■															Возб.
<i>1</i>	0	0					■														
<i>2</i>	0	0	0					■													
<i>3</i>		0	0	0					■												
<i>4</i>	0		0	0	0					■											
<i>5</i>			0	0	0	0					■										
<i>6</i>				1	1	1						■									$m_6 = 5$
<i>7</i>				0	0								■								
<i>9</i>	■	■	■	■				0	0	0									■		
<i>10</i>	■	■	■	■				0	0	0	0								■		
<i>11</i>	■		1	■	■	■				0	0	0	1						■		$m_{11} = 3$
<i>12</i>	■		■	■	■	■				0	0	0	0						■		
<i>Вых2</i>	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	0	0	■		

Рис. 4.8. Матрица следования для обучения второму эталону

То же следует сказать и о нейроне 11.

Исключим из данной матрицы строки и столбцы, соответствующие нейронам 6 и 11. Матрица примет вид, показанный на [рис. 4.9](#).

B2	■																			Возб.		
B3		■																		Возб.		
A1			■																	Возб.		
C1				■																Возб.		
C2					■															Возб.		
C3						■														Возб.		
1	0	0				■																
2	0	0	0				■															
3		0	0	0				■														
4	0		0	0	0				■													
5			0	0	0	0				■												
7					0	0					■											
9	■	■	■	■			0	0	0													
10	■	■	■	■			0	0	0	0												
12	■		■	■	■						0	0	0									
Вых2	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	0

Рис. 4.9. Шаг преобразования матрицы следования

Объединяем столбцы множества входов матрицы, обладающих признаком "возбужден". На данном шаге - это первые шесть столбцов матрицы. Находим в них первую строку, содержащую максимальное число нулей. Это строка, соответствующая нейрону 4. Меняем в ней все нули на единицы, полагаем $m_4 = 4$. Присваиваем нейрону 4 признак "возбужден". Отражаем внесенные изменения весов в матрице S .

Исключаем из матрицы S_2 строки и столбцы, соответствующие нейронам B_2, A_1, C_1, C_2 , "передавшим" свое возбуждение нейрону 4.

Вновь объединяем столбцы множества входов, обладающих признаком "возбужден". Это столбцы, соответствующие нейронам $B_3, C_3, 4$. В совокупности этих столбцов выбираем первую из строк, содержащую максимальное число нулей. В данном случае это строка, соответствующая нейрону 1. Меняем нуль на единицу, полагаем $m_1 = 1$, нейрону 1 присваиваем признак "возбужден".

Исключаем из матрицы строку и столбец, соответствующие нейрону B_3 ,

"передавшему" возбуждение нейрону 1. Исключаем из матрицы строки и столбцы, соответствующие образовавшимся входам - "не возбужденным" нейронам 2 и 3.

Выделяем столбцы, которые соответствуют нейронам - входам, обладающим признаком "возбужден", и в их совокупности находим строку, обладающую наибольшим числом нулей. Такая строка соответствует нейрону 10. Меняем нули на единицы, присваиваем нейрону признак "возбужден", полагаем $m_{10} = 2$.

Исключаем из матрицы S_2 строки и столбцы, которые соответствуют нейронам 1 и 4, передавшим возбуждение нейрону 10.

Нейрон 9, не обладающий признаком "возбужден", образует вход матрицы. Исключаем соответствующие ему строку и столбец.

Выделяем столбцы, соответствующие нейронам 3 и 10, и в них находим строку с максимальным числом нулей. Первая из таких строк соответствует нейрону 5. Заменяем в ней единственный нуль единицей, присваиваем нейрону признак "возбужден", полагаем $m_5 = 1$. Отражаем изменение веса в матрице S .

Исключаем из матрицы S_2 строку и столбец, которые соответствуют нейрону 3, передавшему возбуждение нейрону 5.

Исключаем строку и столбец, соответствующие нейрону 7, как порождающему вход матрицы и не имеющему признака "возбужден".

В совокупности выделенных столбцов, соответствующих нейронам 5 и 10, строка, соответствующая нейрону 12, имеет единственный нуль. Меняем его на единицу, присваиваем нейрону 12 признак "возбужден", полагаем $m_{12} = 1$. Отражаем внесенное изменение веса в матрице S . Исключаем из матрицы S_1 строку и столбец, соответствующие нейрону 5.

И, наконец, на последнем шаге, заменяя нули в строке, соответствующей нейрону $V_{вых2}$, единицами, мы получим окончательный искомый путь возбуждения. Отразим его на рис. 4.10 нейросети.

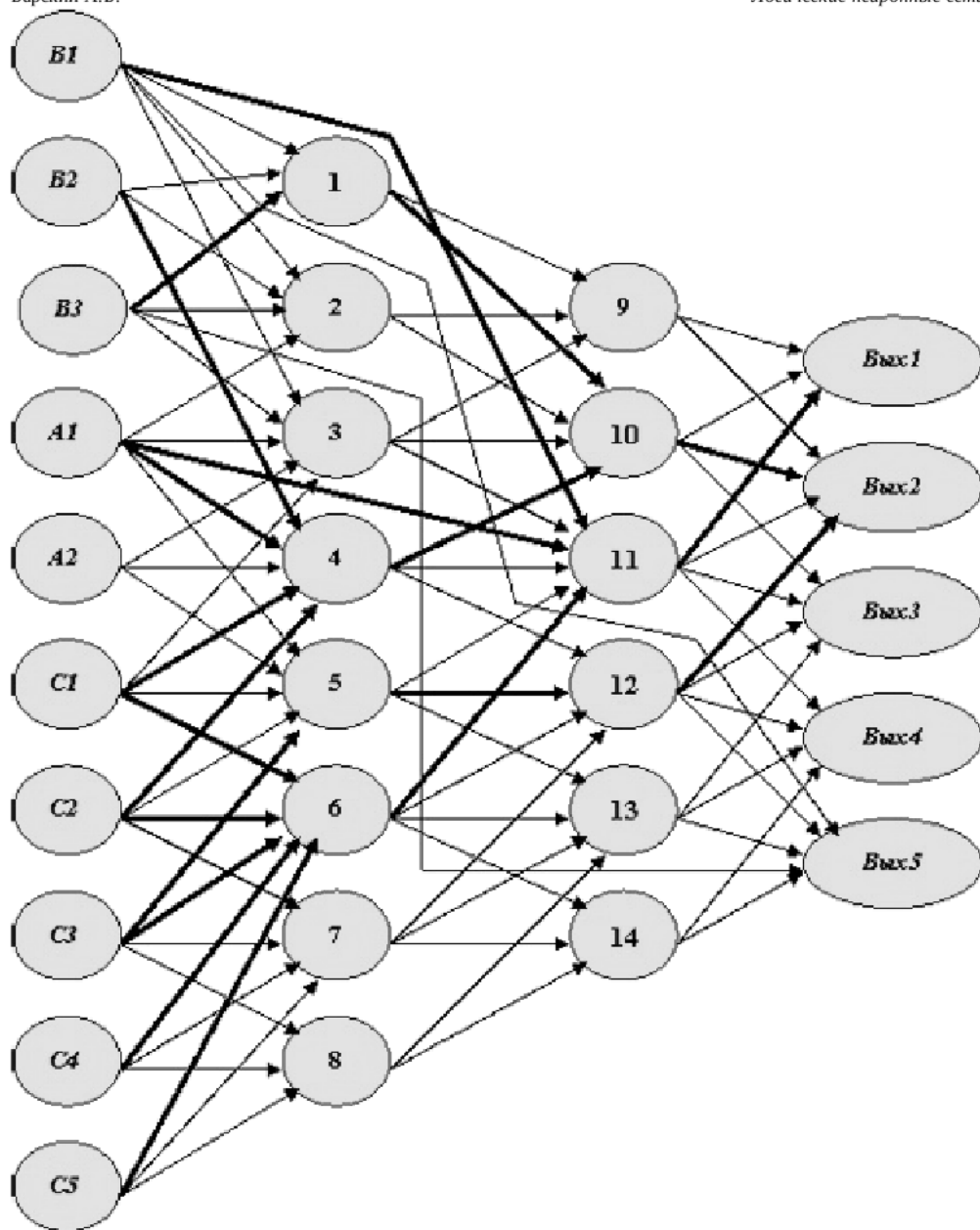


Рис. 4.10. Нейросеть после обучения двум эталонам

Обучим сеть третьей ситуации, требующей решения R3. Закрепим за этим решением нейрон Вых3. Матрица $S_3[B2, B3, A1, C4, C5 \rightarrow \text{Вых3}]$ представлена на [рис. 4.11](#).

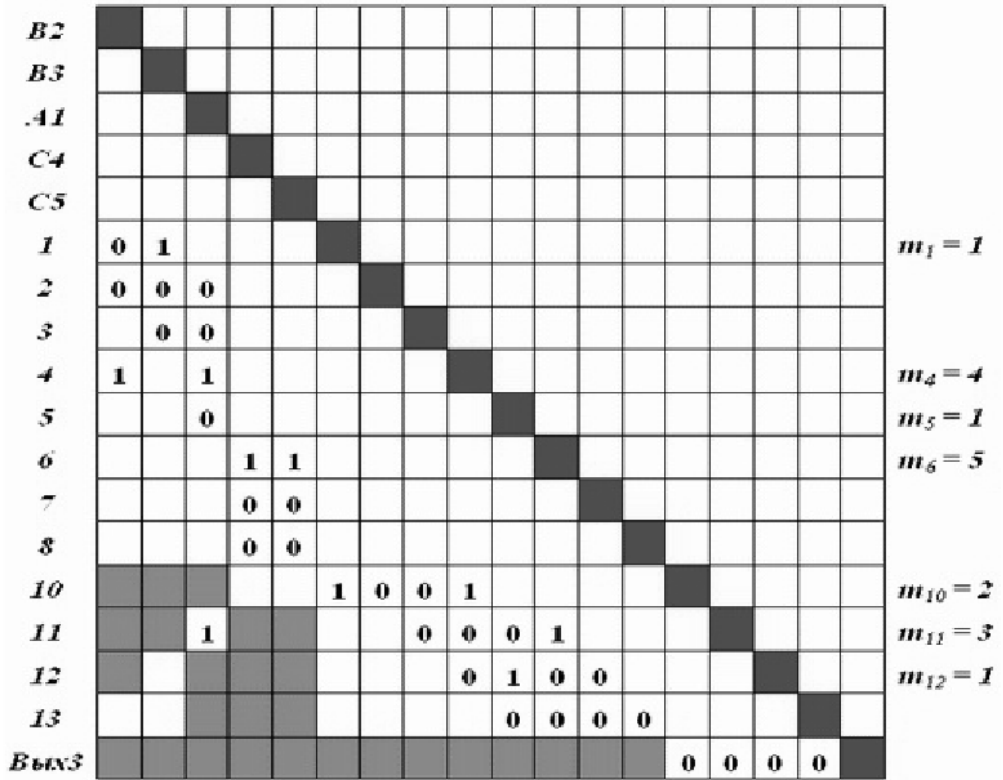


Рис. 4.11. Матрица следования для обучения третьему эталону

Исключаем из матрицы S_3 представительство тех ранее использованных ("возбужденных") нейронов, для которых значение m (указано в дополнительном столбце матрицы) превышает количество единиц в соответствующей строке. Это значит, что эти нейроны "собрали" уже возбуждение нейронов, которые в данном пути возбуждения, т.е. в матрице S_3 , не используются. Это нейроны 4, 5, 6, 11. Переиспользование их невозможно.

Повторяем попытки исключения из матрицы S_3 представительства тех нейронов, для которых значение m стало превышать количество единиц в соответствующей строке. И так далее - до исчерпания этой возможности.

В данном случае нам придется исключить из рассмотрения (вследствие исключения нейронов 4 и 5) нейроны 10 и 12. Матрица S_3 примет

окончательный вид на [рис. 4.12](#).

<i>B2</i>															<i>Возб.</i>
<i>B3</i>															<i>Возб.</i>
<i>A1</i>															<i>Возб.</i>
<i>C4</i>															<i>Возб.</i>
<i>C5</i>															<i>Возб.</i>
<i>1</i>	0	1													$m_1 = 1$
<i>2</i>	0	0	0												
<i>3</i>		0	0												
<i>7</i>				0	0										
<i>8</i>				0	0										
<i>13</i>										0	0				
<i>Вых3</i>												0			

Рис. 4.12. Шаг преобразования матрицы следования

Присваиваем всем нейронам, образующим входы матрицы, признак "возбужден".

Выделяем и объединяем столбцы, соответствующие входам матрицы S_3 .

Разовьем предшествующие аналогичные действия следующим образом.

В выделенных столбцах находим строку, содержащую хотя бы одну единицу. Она (строка) представляет тот нейрон, который "собрал" (при анализе предыдущих эталонов) возбуждение одного или более нейронов, образующих входы. Эти нейроны, "передавшие" свое возбуждение, могут быть исключены из рассмотрения. Среди таких строк выбираем строку с максимальным числом единиц.

Так, нейрон 1 возбужден единственным нейроном, принадлежащим текущему множеству входов матрицы, - нейроном *B3*. Впредь он представляет этот нейрон, который может быть исключен из рассмотрения. Матрица S_3 принимает вид на [рис. 4.13](#).

<i>B2</i>	■											<i>Возб.</i>
<i>A1</i>		■										<i>Возб.</i>
<i>C4</i>			■									<i>Возб.</i>
<i>C5</i>				■								<i>Возб.</i>
<i>1</i>	0				■							<i>Возб.</i>
<i>2</i>	0	0				■						
<i>3</i>		0					■					
<i>7</i>			0	0				■				
<i>8</i>			0	0					■			
<i>13</i>		■	■	■					0	0	■	
<i>Вых3</i>	■	■	■	■	■	■	■	■	■	0	■	

Рис. 4.13. Шаг преобразования матрицы следования

Подтверждаем нейрону 1 признак "возбужден" и превращаем его во вход матрицы S_3 , уничтожив веса в его строке.

Вновь находим множество входов матрицы и выделяем соответствующие им столбцы. Пытаемся найти строку, содержащую единицы в этих и только этих столбцах, но таких больше нет. Тогда находим строку, содержащую максимальное число нулей. Первая такая строка соответствует нейрону 2. "Объединяем" на нем возбуждение нейронов $B2$ и $A1$, заменяя нули единицами. Полагаем $m_2 = 2$, исключаем нейроны $B2$ и $A1$ из рассмотрения, присваиваем нейрону 2 признак "возбужден". Отражаем сделанные изменения весов в матрице S .

Исключаем входы, не обладающие признаком "возбужден", - нейроны 3 и 5.

Выделяем столбцы, соответствующие входам. В их совокупности не находим строк, содержащих единицы. Тогда находим строку, содержащую максимальное число нулей. Такая строка соответствует нейрону 7. Меняем нули (в выделенной совокупности столбцов!) на единицы, полагаем $m_7 = 2$, присваиваем нейрону 7 признак "возбужден", отражаем изменение весов в матрице S . Исключаем

нейроны С4 и С5 из рассмотрения. Сразу замечаем, что сформируется "невозбужденный" вход, соответствующий нейрону 8. Исключаем из рассмотрения и этот нейрон (рис. 4.14).

1						Возб.
2						Возб.
7						Возб.
13			0			
Вых3				0		

Рис. 4.14. Шаг преобразования матрицы следования

Выделяем столбцы, соответствующие "возбужденным" входам, и не находим строк, содержащих единицы. Тогда находим первую строку, содержащую максимальное число нулей. Это строка, соответствующая нейрону 13. Меняем нуль на единицу, нейрону 13 присваиваем признак "возбужден", полагаем $m_{13} = 1$. Исключаем нейрон 7 из рассмотрения.

Выделяем столбцы, соответствующие множеству "возбужденных" входов. В них не находим строку с единицами, но единственная строка с максимальным числом нулей соответствует нейрону Вых3. Меняем нуль на единицу. Однако статус этого нейрона особый, и значение m , как и признак "возбужден", ему не присваиваем.

Исключаем нейрон 13 из рассмотрения.

Выделяем множество столбцов, соответствующих "возбужденным" входам. Это столбцы, соответствующие нейронам 1 и 2. Пытаемся в этих столбцах найти строку с единицами, затем с максимальным числом нулей, - но таковых нет! Значит, необходимо введение дополнительных связей. Тогда в каждом столбце при просмотре сверху вниз находим первую *транзитивную связь* и меняем ее на единицу. Это означает, что данная сеть дополняется *динамическими цепочками* возбуждения $1 \rightarrow \text{Вых3}$ и $2 \rightarrow \text{Вых3}$.

Построение трассы решения R3 закончено. После этого этапа обучения сеть имеет вид как на рис. 4.15.

Приступим к обучению следующему *обобщенному эталону* - решению $A_2 \& B_3 \& C_1 \& C_2 \& C_3 \& C_4 \& C_5 \rightarrow R_4$. Матрица следования $S_4 [B_3, A_2, C_1, C_2, C_3, C_4, C_5 \rightarrow \text{Вых}4]$ представлена на рис. 4.16.

Исключим из рассмотрения те нейроны, для которых количество единиц в строках меньше значения m . Такими нейронами являются 4 и 11. Матрица S_4 имеет вид как на рис. 4.17.

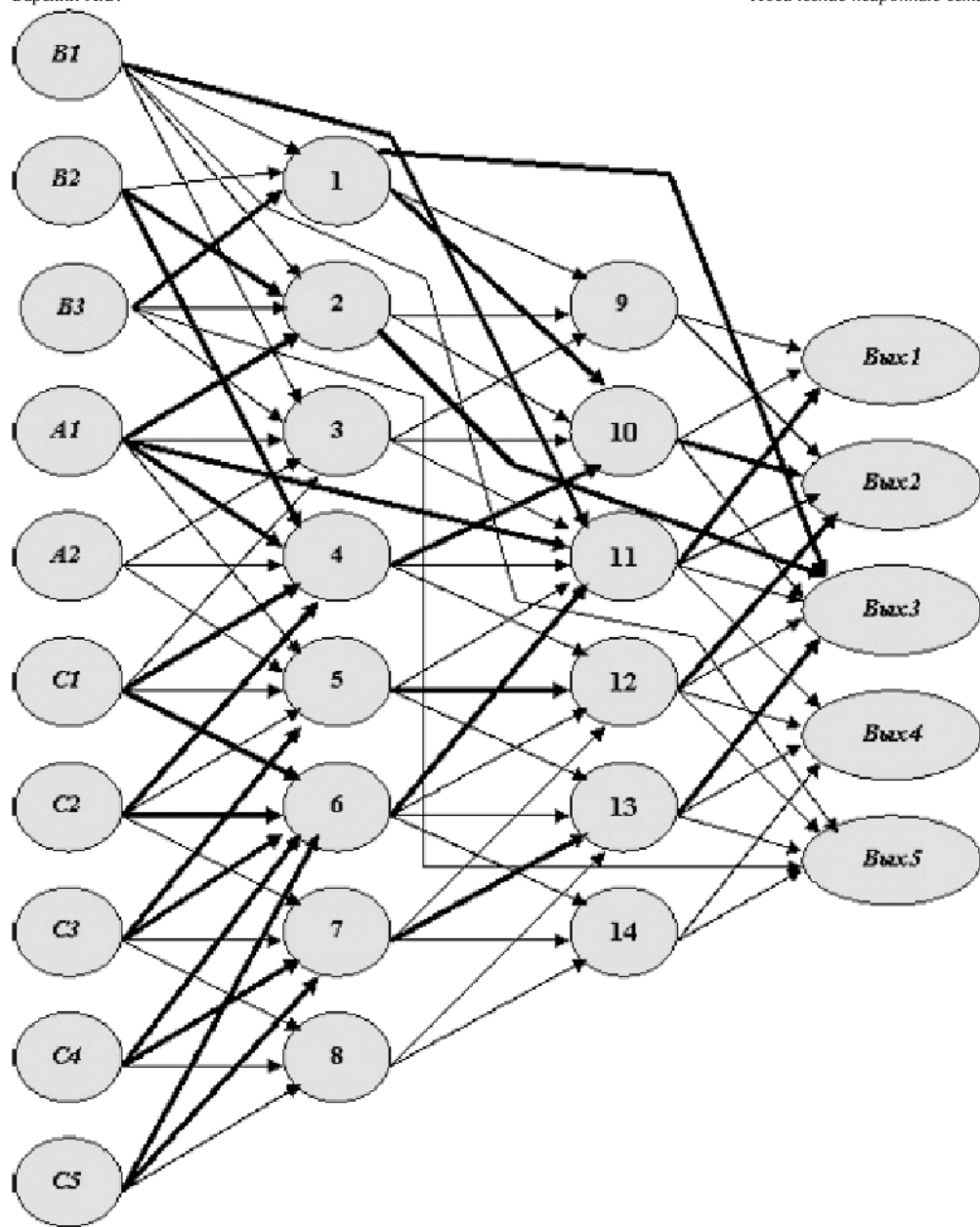


Рис. 4.15. Нейросеть после обучения третьему эталону

<i>B3</i>	■																
<i>A2</i>		■															
<i>C1</i>			■														
<i>C2</i>				■													
<i>C3</i>					■												
<i>C4</i>						■											
<i>C5</i>							■										
3	0	0	0					■									
4		0	1	1					■								$m_4 = 4$
5		0	0	0	1					■							$m_5 = 1$
6			1	1	1	1	1				■						$m_6 = 5$
7				0	0	1	1					■					$m_7 = 2$
8					0	0	0						■				
11	■	■	■	■	■	■	■	0	0	0	1				■		$m_{11} = 3$
12		■	■	■	■	■	■		0	1	0	0				■	$m_{12} = 1$
13		■	■	■	■	■	■			0	0	1	0			■	$m_{13} = 1$
14			■	■	■	■	■				0	0	0				■
<i>Выход</i>	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	0	0

Рис. 4.16. Матрица следования для обучения четвертому эталону

<i>B3</i>	■																			
<i>A2</i>		■																		
<i>C1</i>			■																	
<i>C2</i>				■																
<i>C3</i>					■															
<i>C4</i>						■														
<i>C5</i>							■													
<i>3</i>	0	0	0					■												
<i>5</i>		0	0	0	1				■											$m_5 = 1$
<i>6</i>			1	1	1	1	1			■										$m_6 = 5$
<i>7</i>				0	0	1	1				■									$m_7 = 2$
<i>8</i>					0	0	0					■								
<i>12</i>		■	■	■	■	■	■		1	0	0			■						$m_{12} = 1$
<i>13</i>		■	■	■	■	■	■		0	0	1	0			■					$m_{13} = 1$
<i>14</i>			■	■	■	■	■			0	0	0					■			
<i>Вых4</i>	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	0	■

Рис. 4.17. Шаг преобразования матрицы следования

Присваиваем всем нейронам-входам значение "возбужден". Выделяем столбцы, соответствующие входам - первым строкам "возбужденных" нейронов. Это первые семь столбцов. Среди строк этой совокупности столбцов находим строку с максимальным числом единиц в этих столбцах, при отсутствии единиц в других столбцах. Это строка, соответствующая нейрону 6. Теперь нейрон 6 будет представлять нейроны C_1 , C_2 , C_3 , C_4 , C_5 , "передавшие" ему свое возбуждение. Присваиваем ему признак "возбужден", исключаем из рассмотрения перечисленные нейроны, матрица S_4 принимает вид как на рис. 4.18.

<i>B3</i>	■										<i>Возб.</i>
<i>A2</i>		■									<i>Возб.</i>
<i>3</i>	0	0	■								
<i>5</i>		0		■							$m_5 = 1$
<i>6</i>					■						<i>Возб.</i>
<i>7</i>						■					$m_7 = 2$
<i>8</i>							■				
<i>12</i>		■		1	0	0		■			$m_{12} = 1$
<i>13</i>		■		0	0	1	0		■		$m_{13} = 1$
<i>14</i>				0	0	0				■	
<i>Выход</i>	■	■	■	■	■	■	■	0	0	0	■

Рис. 4.18. Шаг преобразования матрицы следования

Для "не возбужденных" нейронов исключаем из матрицы S_4 строки (и столбцы), число единиц в которых стало меньше соответствующего значения m . Они соответствуют нейронам 5 и 7. Повторяем этот шаг до полного исключения таких нейронов, - исключаются нейроны 12 и 13. Матрица имеет вид как на [рис. 4.19](#).

<i>B3</i>	■										<i>Возб.</i>
<i>A2</i>		■									<i>Возб.</i>
<i>3</i>	0	0	■								
<i>6</i>				■							<i>Возб.</i>
<i>8</i>					■						
<i>14</i>				0	0	■					
<i>Выход</i>	■	■	■	■	■	0	■				

Рис. 4.19. Шаг преобразования матрицы следования

Исключаем из рассмотрения множество "невозбужденных" входов. К таким относится нейрон 8. Выделяем множество столбцов, соответствующих входам матрицы. В их совокупности находим первую строку, обладающую максимальным числом нулей (единиц нет во всей матрице!). Такая строка соответствует нейрону 3. Нули в ней соответствуют нейронам *B3* и *A2*. Исключаем эти нейроны из

рассмотрения, полагаем $m_3 = 2$, присваиваем нейрону 3 признак "возбужден".

Вновь выделяем множество столбцов, соответствующих входам матрицы, и так как в их совокупности нет строк, содержащих единицы, находим строку с максимальным количеством нулей. Эта строка соответствует нейрону 14. Заменяем в ней нуль на единицу, полагаем $m_{14} = 1$, присваиваем нейрону 14 признак "возбужден". Исключаем нейрон 6 из матрицы. Матрица принимает вид как на [рис. 4.20](#).

3				Вол.
14				Вол.
Вых4		0		

Рис. 4.20. Шаг преобразования матрицы следования

Выделяем множество столбцов, соответствующих входам матрицы, и так как в их совокупности нет строк, содержащих единицы, находим строку с максимальным количеством нулей. Эта строка соответствует нейрону выходного слоя **Вых4**. Меняем в ней нуль на единицу. Так как это - нейрон выходного слоя, не присваиваем ему признак "возбужден" и значение m . Исключаем нейрон 14 из рассмотрения. Матрица имеет вид как на [рис. 4.21](#).

3			Вол.
Вых4			

Рис. 4.21. Шаг преобразования матрицы следования

В этой матрице множество "возбужденных" входов составляет единственный нейрон 3. Однако в его столбце нет ни единичных, ни нулевых элементов. Тогда, как и прежде, вводим в сеть дополнительную связь, найдя в столбце, соответствующем нейрону 3, первую сверху *транзитивную связь* и положив ее вес равным единице. Эта связь порождает *динамический путь возбуждения* $3 \rightarrow \text{Вых4}$. Все изменения отражаем в матрице S .

Построение трассы решения R4 закончено. Не приводя промежуточного

вида матрицы S , дадим на рис. 4.22 вид сети, полученной после данного этапа трассировки.

И, наконец, приступим к трассировке последнего *пути возбуждения*. Он осуществляется по *обобщенному эталону*, требующему решения R_5 , $A_2 \& B_1 \& B_2 \& C_1 \& C_2 \& C_3 \& C_4 \& C_5 \rightarrow R_5$. За решением R_5 закреплен нейрон выходного слоя $Вых_5$.

На основе текущего вида матрицы S построим матрицу $S_5 [B_1, B_2, A_2, C_1, C_2, C_3, C_4, C_5 \rightarrow Вых_5]$, исключая из S "ненужные" строки и столбцы и сохраняя *транзитивные связи*. Затем нам придется исключить из рассмотрения нейрон 4, для которого количество единиц в строке меньше значения $m_4 = 4$. Получившаяся матрица S_5 представлена на рис. 4.23.

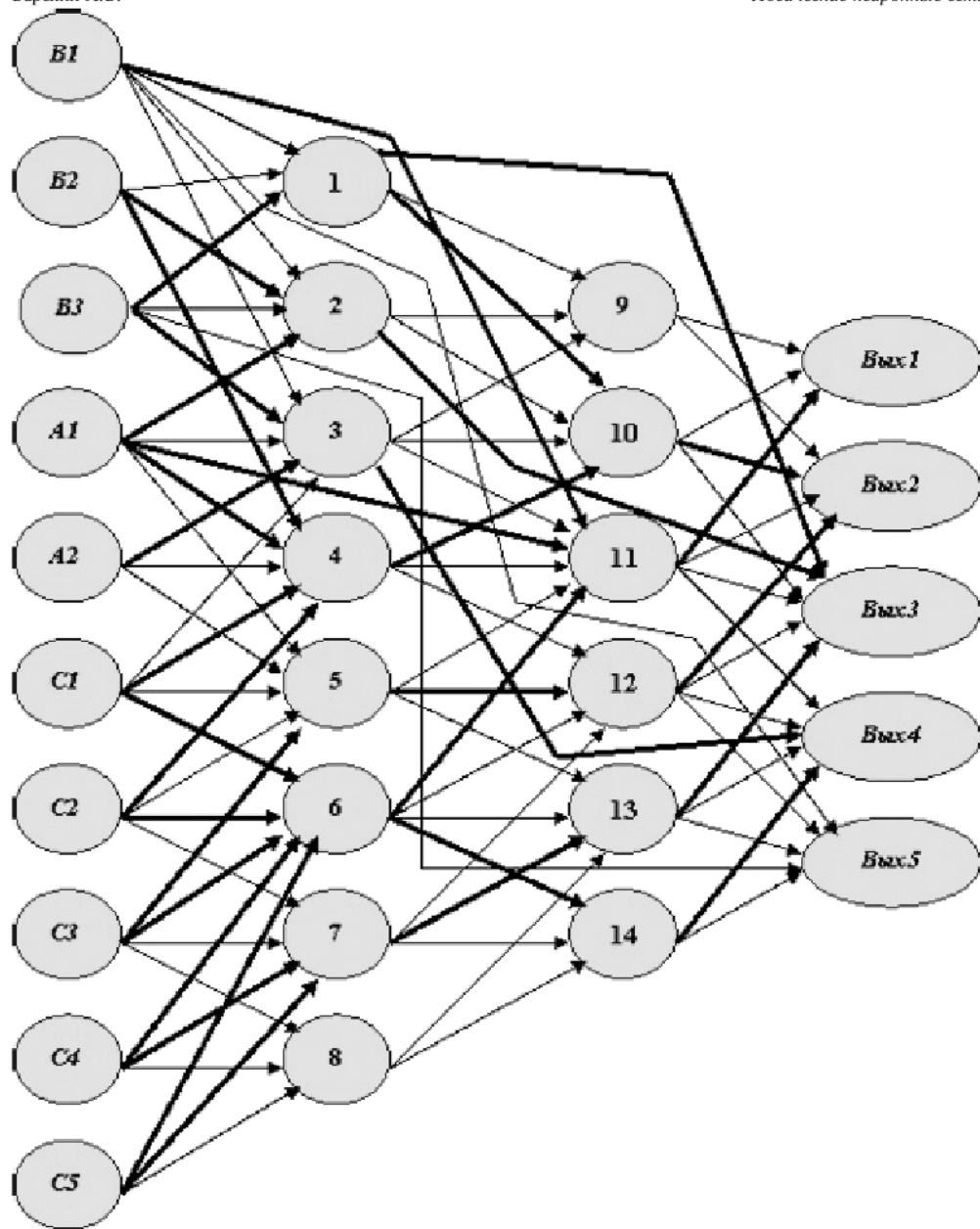


Рис. 4.22. Нейросеть после обучения четырьмя эталонами

B1	■																Волб.
B2		■															Волб.
A2			■														Волб.
C1				■													Волб.
C2					■												Волб.
C3						■											Волб.
C4							■										Волб.
C5								■									Волб.
5			0	0	0	1			■								$m_5 = 1$
6				1	1	1	1	1		■							$m_6 = 5$
7					0	0	1	1			■						$m_7 = 2$
8						0	0	0				■					
12		■	■	■	■	■	■	■	1	0	0		■				$m_{12} = 1$
13			■	■	■	■	■	■	0	0	1	0		■			$m_{13} = 1$
14					■	■	■	■		1	0	0			■		$m_{14} = 1$
Вых.5	θ	■	■	■	■	■	■	■	■	■	■	■	0	0	0	■	

Рис. 4.23. Матрица следования для обучения пятому эталону

Присваиваем всем входам признак "возбужден". Выделяем столбцы, соответствующие "возбужденным" входам. В совокупности этих столбцов находим строку, содержащую максимальное число единиц в этих и только этих столбцах, если таковые имеются. В данном случае это строка, соответствующая нейрону 6. Присваиваем нейрону 6 признак "возбужден" и исключаем из рассмотрения нейроны C1, ..., C5.

Исключаем из рассмотрения нейроны 5 и 7, так как в соответствующих им строках отсутствуют единицы при отличных от нуля значениях m . Однако видно, что после их исключения придется исключить по той же причине нейроны 12 и 13. Получившаяся матрица представлена на [рис. 4.24](#).

В1	■							Возб.
В2		■						Возб.
А2			■					Возб.
6				■				Возб.
8					■			
14				1	0	■		$m_{14} = 1$
Вых.5	0	■	■	■	■	0	■	

Рис. 4.24. Шаг преобразования матрицы следования

Исключаем из рассмотрения "невозбужденный" вход, соответствующий нейрону 8. Выделяем столбцы, соответствующие "возбужденным" входам, и в их совокупности находим строку, содержащую наибольшее число единиц при отсутствии единиц в других столбцах. Такая строка соответствует нейрону 14. Исключаем из рассмотрения нейрон 6 как передавший свое возбуждение нейрону 14, присваиваем нейрону 14 признак "возбужден". Матрица S_5 принимает вид как на [рис. 4.25](#).

В1	■							Возб.
В2		■						Возб.
А2			■					Возб.
14				■				Возб.
Вых.5	0	■	■	■	■	0	■	

Рис. 4.25. Шаг преобразования матрицы следования

Выделяем столбцы входов, соответствующие "возбужденным" нейронам, и в их совокупности находим строку, содержащую максимальное число нулей (строк с единицами больше нет). Такая строка соответствует нейрону выходного слоя **Вых.5**. Меняем нули на единицы. Исключаем нейроны В1 и 14 из рассмотрения. Отражаем введенные изменения в матрице S . Новый вид матрицы S_5 приведен на [рис. 4.26](#).

В2				Возб.
А2				Возб.
Вых5				

Рис. 4.26. Шаг преобразования матрицы следования

Выделяем множество столбцов, соответствующих входам матрицы. В этих столбцах не находим ни строк, содержащих единицы, ни строк, содержащих нули. Тогда в каждом столбце заносим единицы на места *транзитивных связей*, то есть вводим связи $V2 \rightarrow \text{Вых}5$ и $A2 \rightarrow \text{Вых}5$, а веса этих связей полагаем равными единице.

Трассировка (обучение сети по *обобщенным эталонам*) закончена. Окончательный ее вид представлен на рис. 4.27, где, напоминая, веса, принявшие значение 1, соответствуют жирным стрелкам.

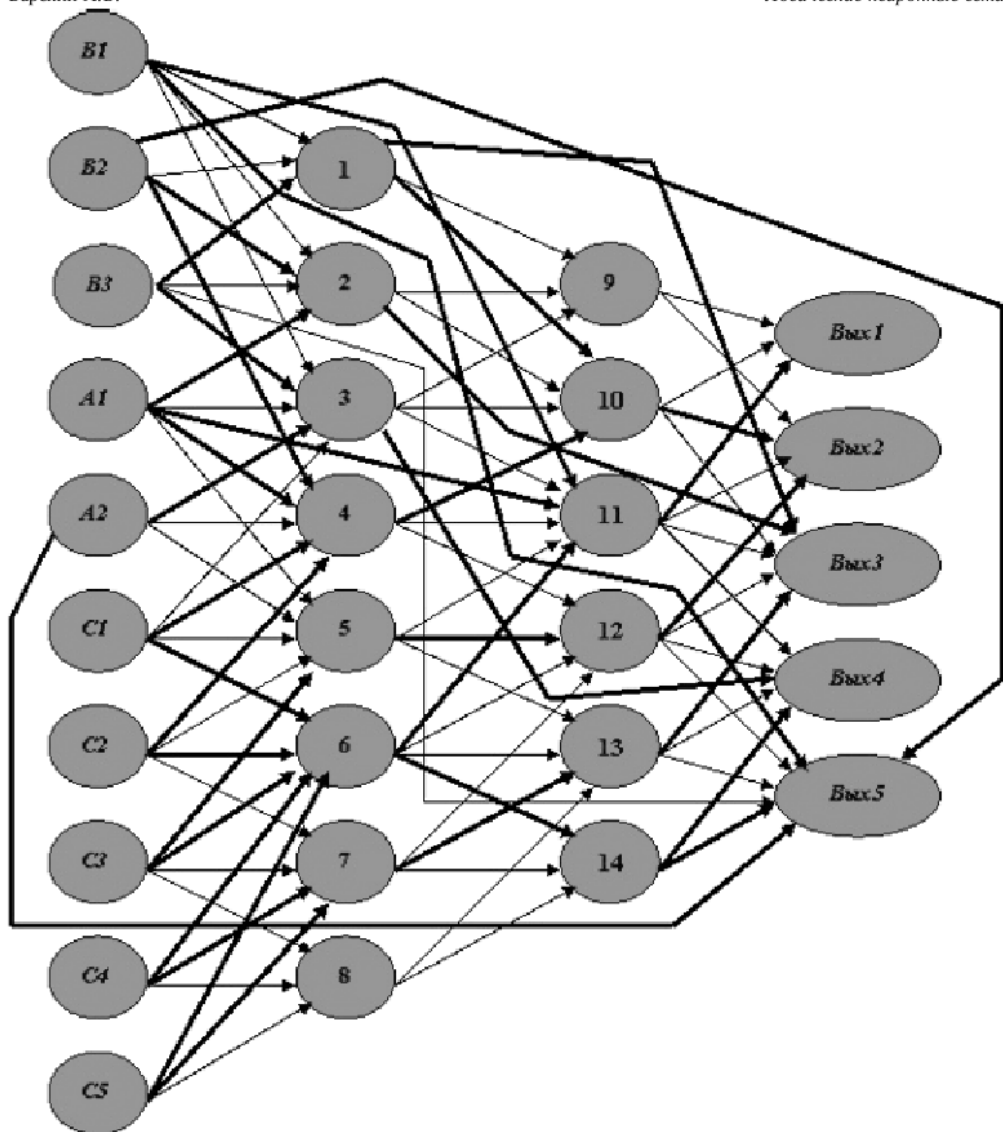


Рис. 4.27. Обученная нейросеть

4.2. Алгоритм трассировки нейросети

В результате решения примера сформировались и даже стали привычными действия, на основе которых мы можем сформулировать

Алгоритм трассировки (обучения по обобщенным эталонам)

нейросети

1. Дополняем матрицу следования S транзитивными связями по алгоритму, представленному в разделе 3.4.
2. Все нейроны выходного слоя не должны иметь в соответствующих им строках "пустых" элементов в позициях, соответствующих нейронам входного слоя. Пустой элемент указывает на отсутствие *пути возбуждения* от соответствующего нейрона входного слоя. Мы будем считать нейросеть построенной некорректно и нуждающейся во внесении дополнительных связей, например, непосредственно от нейрона входного слоя к нейрону выходного слоя.
3. Организуем перебор всех обобщенных эталонных ситуаций и соответствующих им решений, последовательно закрепляя нейроны выходного слоя $Вых_i$ за обобщенными ситуациями. Для каждой обобщенной ситуации выполняем пункты 4-15.
4. Для *обобщенного эталона* i ($i = 1, 2, \dots, m$) строим матрицу следования $S_i[V_{i1}, V_{i2}, \dots, V_{ir} \rightarrow Вых_i]$, где $V_{i1}, V_{i2}, \dots, V_{ir}$ - нейроны входного слоя, возбуждающиеся (до максимальной величины) при подаче *обобщенного эталона*, т.е. характеризующие ситуацию.
5. В матрице S_i последовательно, сверху вниз, вычеркиваем строки (и соответствующие столбцы), которые содержат количество единиц меньше значения m , указанного при строке.

Примечание. Так как в процессе такого вычеркивания могут образовываться новые подобные строки, а матрица S - треугольная, то это вычеркивание должно быть произведено последовательно, сверху вниз по одной строке. Тогда все вычеркивание выполнится за один проход. В противном случае, если сразу наметить для вычеркивания несколько строк (и столбцов), не избежать повторного, возможно, многократного анализа появления новых строк для вычеркивания.

6. Присваиваем признак "возбужден" всем нейронам входного слоя, представленным в матрице S_i .
7. Проверяем: содержит ли матрица S_i более одной строки? Если содержит, выполняется следующий пункт, в противном случае

выполняется пункт 3.

8. Исключаем из матрицы S_{\downarrow} строки и столбцы, соответствующие нейронам–входам, не обладающим признаком "возбужден".
9. Выделяем множество столбцов матрицы S_{\downarrow} , обладающих признаком "возбужден".
10. Выполняем действие, отраженное в пункте 5 (во внешнем цикле): исключаем из текущего вида матрицы S_{\downarrow} строки (и столбцы), которые содержат количество единичных элементов меньше, чем указанное при строке значение m .

Примечание. Такое действие необходимо после каждого вычеркивания строк и столбцов.

11. В совокупности выделенных столбцов находим (если таковая имеется) первую строку, содержащую максимальное число единиц и не содержащую единиц в других столбцах. (Число найденных единиц не должно быть меньше соответствующего значения m .) Соответствующий ей нейрон может быть переиспользован. Если такой строки найти не удастся, выполняем пункт 13.
12. Исключаем из рассмотрения нейроны (вычеркиваем строки и столбцы) которым соответствуют единицы в найденной строке. Присваиваем нейрону, соответствующему выделенной строке, признак "возбужден". Уничтожаем в выделенной строке все нули и символы *транзитивных связей*, если они имеются, - превращаем строку во вход матрицы S_{\downarrow} .

Примечание. В рассмотренном примере обращение строки во вход матрицы пришлось делать однажды, при трассировке третьего эталона. Однако при развитии примера в следующей лекции такое действие придется выполнять многократно.

Переходим к выполнению пункта 7.

13. В совокупности выделенных столбцов находим, если таковая имеется, первую строку, содержащую максимальное число нулевых элементов. Если такой строки найти не удастся, выполняем пункт 15.

14. Меняем значение возбуждения соответствующих связей, то есть заменяем нули единицами. Присваиваем нейрону, соответствующему выделенной строке, значение m , равное количеству единиц в строке, и признак "возбужден". Исключаем из рассмотрения нейроны (вычеркиванием строк и столбцов), "передавшие" свое возбуждение найденному нейрону.

Примечание. Значения весов связей одного нейрона могут корректироваться лишь однажды. В других ситуациях, при обучении другим эталонам, нейрон может только переиспользоваться, если в этом обучении участвуют все те нейроны, возбуждение которых он использует с весом, равным единице. При этом достаточно учитывать лишь число единиц в строке. В процессе такого обучения эталоны не мешают друг другу!

Внесенные изменения весов учитываем в матрице S . Переходим к выполнению пункта 7.

15. По каждому выделенному столбцу "спускаемся" вниз и находим первый из непустых элементов, соответствующий *транзитивной* связи. Вводим в нейросеть дополнительную связь, присваивая единичное значение найденному элементу. Исключаем из рассмотрения (вычеркиваем строки и столбцы) нейроны, соответствующие обработанным столбцам. Отражаем внесенные изменения в матрице S .

Примечание. Как мы видели в примере, а, по-видимому, это верно всегда, такая *транзитивная* связь потребуется на последней стадии выполнения алгоритма и может обнаружиться лишь в строке, соответствующей нейрону выходного слоя. Поэтому формировать значение m уже излишне, так как это может быть последним актом выполнения данного алгоритма.

Переходим к выполнению шага 7.

Описание алгоритма закончено.

Построенный алгоритм трассировки, несомненно, эвристический, то

есть дающий приблизительное, удовлетворительное решение. Точный алгоритм трассировки, минимизирующий число использованных нейронов и дополнительных связей, требует совместного анализа всех эталонов и решений, выделения и создания термов, участвующих в получении всех решений. Но само понятие точности алгоритма трассировки проблематично ввиду неопределенности критерия.

Так, в нашем случае удачно сложился терм в результате связи [C1, C2, C3, C4, C5 → 6]. Он использовался при получении трех решений по эталонным ситуациям – R1, R4, R5. По-видимому, целесообразны термы в результате объединения C1, C2, C3, а также C4 и C5, V1 и V2 и др.

Мы предлагаем читателю самому произвести трассировку сети, представленной на рис. 2.14, по предложенному алгоритму. Избежит ли он введения дополнительных связей? Мы не знаем. Но, во-первых, мы строим простой, нетрудоемкий алгоритм; во-вторых - мы же раньше говорили о нецелесообразности экономии! При такой скудости наших знаний на что нам тратить сто миллиардов нейронов с десятью тысячами дендритов каждый?!

4.3. Приведение нейросети после трассировки

Зафиксируем передаточную функцию, например

$$V := \sum_j \omega_j V_j$$

$$V_i := \text{if } V > h \text{ then } V \text{ else } 0$$

Будем задавать конкретные эталоны и наблюдать величину возбуждения соответствующего нейрона выходного слоя.

Рассмотренный выше пример нехарактерен. В нем каждый эталон приводил, при данной передаточной функции, к одинаковой величине возбуждения нейронов выходного слоя. Но, например, при распознавании знаков алфавита величины возбуждения нейронов выходного слоя, в ответ на предъявление различных эталонов, могут быть различными. Так, в первом примере лекции 2 для записи буквы О потребовалось возбуждение шести нейронов-рецепторов, а для записи буквы А – восьми рецепторов. Очевидно, эталон латинской I "засветит"

на входном слое (на клетчатке) меньшее число нейронов, чем, скажем, буква Q, но значительно большее число, чем необходимо для записи запятой. Значит, при рассматриваемых простейших видах передаточной функции величины возбуждения на выходе будут существенно различаться.

Тогда закрадывается подозрение: а не приведет ли некоторое незначительное искажение ситуации А, логично требующее все же решения R_A , в сторону ситуации В, то есть принятия решения R_B ?

Разная по возможному максимуму величина возбуждения нейронов выходного слоя утверждает неравноправие, приоритет ситуаций. И возбуждение нейронов выходного слоя следует как-то уравнивать по эталонным ситуациям, привести к одному диапазону изменения или к достаточно близким величинам.

Введем эту величину U , например, как максимальное значение возбуждения некоторого нейрона выходного слоя или понравившуюся нам оценку, не обязательно превышающую такой максимум. Введем для каждого нейрона выходного слоя коэффициент приведения

$$k_j = \frac{U_j}{V_j}.$$

Однако значение V_j находим не по *обобщенному эталону*, а по каждой конкретной ситуации, входящей в состав *обобщенного эталона*. Например, ситуация {A1, B1, C1} образует реальный эталон в составе *обобщенного эталона* {A1, B1, C1, C2, C3, C4, C5}. По нему (другие ситуации аналогичны) находится величина возбуждения $V_{\text{Вых1}}$ нейрона Вых1 . Значит,

$$k_{\text{Вых1}} = \frac{U}{V_{\text{Вых1}}}$$

Впредь, в режиме распознавания, каждое значение возбуждения нейрона выходного слоя будем умножать на его коэффициент приведения. Таким образом, мы поставим все ситуации на входе в равное положение.

Однако в лекциях 9–11 будет показано, что в общем случае проблема приведения значительно более сложна. Она требует корректировки весов связей, в частности, в зависимости от того, какая логическая операция явилась прообразом данного нейрона выходного слоя.

Стратегии обучения и самообучения

Обсуждается процесс постепенного обучения, свойственный последовательному познанию предмета исследований. При построении систем принятия решений именно постепенное познание обуславливает возможность доведения таких систем до совершенства.

"Учиться, учиться и еще раз учиться!"

В.И. Ленин

"Мы все учились понемногу..."

А.С. Пушкин

5.1. Динамизм обучения

На самом деле обучение не бывает внезапным, как мы это представили выше, рассматривая сразу *обобщенные эталоны*. Мгновенное заполнение губки памяти нашего разума из Всеобщего Информационного Пространства в момент рождения - лишь фантастическая мечта, которая опирается на опыт предшествующих земных цивилизаций, известный восточным учениям. Долгий и тернистый путь образования (слово-то какое в русском языке: образование - создание, произведение, построение, - человека!), предполагает многократность и постепенность, изменимость в совершенствовании и утверждение, трудоволический надрыв и преодоление, победный клич и разочарование.

Итак, можно учить по "чистым" эталонам, объединив их в *эталонные обобщенные*. Но "ближе к жизни" можно учить по разным "*частным*" эталонам (то, что мы называем жизненным опытом), приводящим к одному решению. Например, можно (возвращаясь к многострадальному примеру) последовательно "показывать" эталоны $B1 \& A1 \& C2$, $B1 \& A1 \& C1$ и т.д., приводящие к решению $R1$. Можно предъявлять попеременно разные эталоны, приводящие к разным решениям. И это справедливо соответствует диалектике познания на основе гипотез, экспериментов и опыта, гласящей, что мир познаваем, но всего познать

нельзя, ибо, мы добавляем, - Бесконечность!

Например, дядя Рамзай не мог ранее предвидеть некоторых комбинаций событий, предварительно имея ошибочное суждение о распределении работ своих клиентов. Или ему даже пришлось употребить свои связи для того, чтобы осуществить вожденную мечту: включить в орбиту своих действий восхитительную многоприбыльную продукцию фирмы Ночная Бабочка!

Во всех случаях на этапе обучения целесообразно определять события А, В, С с максимальной достоверностью, равной единице, рассчитывая тем самым реакцию нейросети на вполне определенные ситуации. Это следует из того, что даже возникшая при предъявлении эталона малая величина возбуждения нейрона входного слоя указывает, что событие возможно и, следовательно, нуждается в рассмотрении.

Того же требует технология *трассировки* - "прокладывания" опорных путей в сети. Она ставит вопрос ребром: "Что делать, если событие наступило?"

Как и все знания в основе нашей мудрости, сеть должна эволюционировать: пополняться и развиваться. Лишь на этой основе можно ставить вопрос не только об *обучении (с учителем)*, но и о *самообучении*.

Таким образом, мы не сразу предъявляем сети весь *обобщенный эталон*, например, $A1 \& B1 \& C1 \& C2 \& C3 \& C4 \& C5$, приводящий к решению $R1$, а последовательно развиваем его, предъявляя *частные эталоны* "в полном смысле" - как единичные значения соответствующих булевых переменных или целых. Возможно, что мы не все такие эталоны даже можем перебрать практически. Например, спустя год после успешной деятельности Компании Петя впервые направился к Марине, торгующей тибетским бальзамом. Пусть лишь глубокий анализ ситуации показал, что это требует *трассировки* решения $A2 \& B3 \& C3 \rightarrow R4$.

Конечно, мы можем поступить просто. Мы можем по вновь появившимся *частным эталонам* сформировать новые *обобщенные эталоны* и, отвергнув все ранее проведенное обучение (положив все веса связей в нейросети равными нулю), вновь произвести обучение нейросети по всему множеству *обобщенных эталонов*, включая

уточненные. Однако это может противоречить динамике всего совокупного процесса обучения и распознавания, непрерываемому участию сети в системе управления, малой трудоемкости обучения. Обучение должно быть столь же оперативным и динамичным, как распознавание.

Другой путь учета новых эталонов заключается в "обнулении" или минимизации лишь тех весов связей, которые обусловлены предыдущим значением *обобщенного эталона*, приводящего к тому же решению. Это выводит использованные ранее нейроны в ресурс. Затем *трассировка* может быть выполнена заново по уточненному *обобщенному эталону*. Этот путь может оказаться более коротким, если не учитывать затрат на ликвидацию следов присутствия устаревшего *обобщенного эталона*. (Операции "чистки" весьма распространены в практике человеческой деятельности!) Мы не оспариваем этот путь, полагаясь на творческое, критическое настроение читателя, и признаемся честно, что мы этот путь не исследовали, полагаясь на интуицию. Интуиция же нам подсказывает, что надо действовать как-то проще, тем более что мы не постесняемся ввести дополнительные связи в сеть, если понадобится.

Итак, рассмотрим проблему динамического включения новых *частных эталонов* в состав *обобщенного эталона* на фоне уже произведенного частичного обучения нейросети.

Пусть предъявление эталонов $A1 \& B1 \& C2$ и $A1 \& B1 \& C4$ ("текущий" *обобщенный эталон* $A1 \& B1 \& C2 \& C4$), требующих решения $R1$, а также предъявление текущих *обобщенных эталонов* $A1 \& B2 \& C1 \& C2 \& C3$, $A1 \& B2 \& C4 \& C5$, $A2 \& B3 \& C1 \& C2 \& C3 \& C4 \& C5$, $A2 \& B1 \& C1 \& C2 \& C3 \& C4 \& C5$, требующих соответственно решений $R2$, $R3$, $R4$, $R5$, привели к *трассировке* нейросети (к ее текущему состоянию), представленной на рис. 5.1.

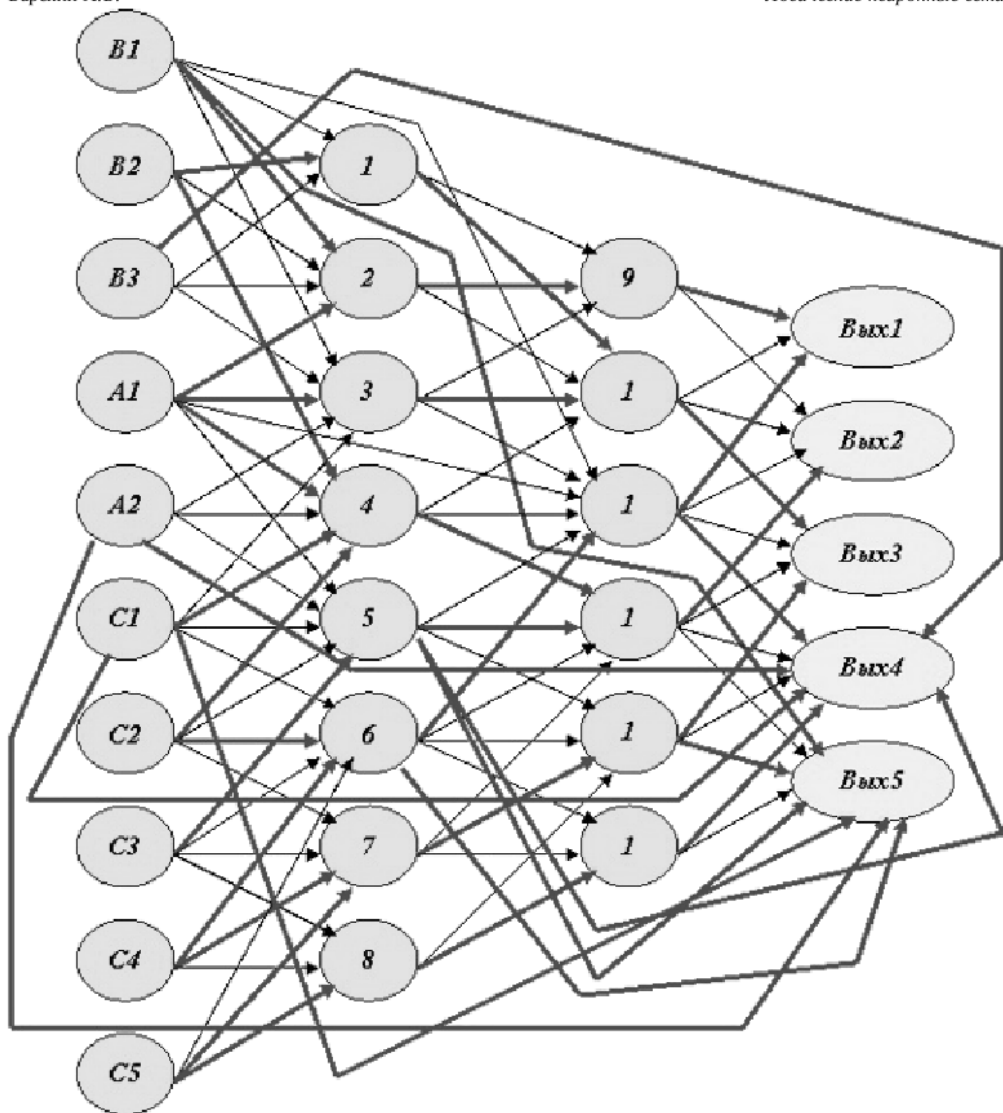


Рис. 5.1. Частично обученная нейронная сеть

Здесь "жирные" связи обладают максимальным значением веса (скорее всего - единичным). Такое состояние получено с помощью алгоритма *трассировки*, представленного в предыдущей лекции, и это предлагается проверить читателю в порядке закрепления материала.

Матрица следования S, соответствующая получившейся сети, показана на [рис. 5.2](#).

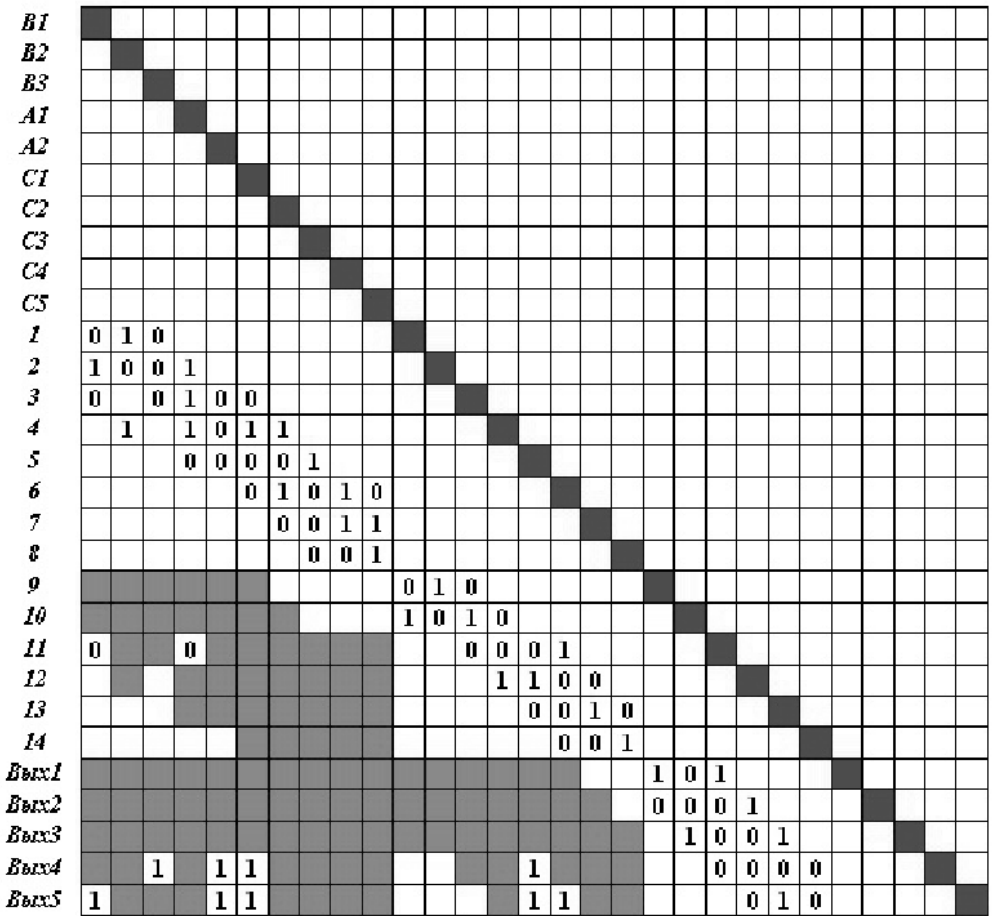


Рис. 5.2. Матрица следования

Как видим, сеть оказалась весьма "запутанной" дополнительными связями. Обучение всем *обобщенным эталонам* сразу (пример в [лекции 4](#)) выявляет термы, использующиеся при получении различных решений. Здесь же термы не складывались, нейроны почти не переиспользовались.

Это и привело к формированию большого числа дополнительных связей. (Мы не любим, когда из нас "вытягивают" одно за другим. Мы возмущаемся: "Не тяни, выкладывай все, что тебе нужно!")

Пусть в процессе эксплуатации обученной сети выяснилось, что ситуации (эталон) A1&B1&C1 и A1&B1&C3, в дополнение к ранее

предусмотренным, не только возможны, но требуют того же решения $R1$. Это означает, что *обобщенный эталон* $A1 \& B1 \& C2 \& C4$ мы должны расширить до подобного эталона $A1 \& B1 \& C1 \& C2 \& C3 \& C4$, требующего решения $R1$. То есть нам необходимо построить *путь возбуждения* (трассы) $B1, A1, C1, C2, C3, C4 \rightarrow \text{Вых}1$. Однако при его построении мы не хотим изменять уже ранее построенный *путь возбуждения* $B1, A1, C2, C4 \rightarrow \text{Вых}1$.

Значит, надо проложить трассы, дополнить сеть *путем возбуждения* $C1, C3 \rightarrow \text{Вых}1$. При этом нам бы хотелось максимально объединить, пересечь, слить этот путь с уже построенным ранее *путем возбуждения*, ведущим в $\text{Вых}1$.

Преследуя эту цель, мы не должны допустить слияния, влияния этого пути на *пути возбуждения*, ведущие в другие нейроны выходного слоя. Такая опасность кроется в естественном желании переиспользования нейронов. Ведь слияние двух *путей возбуждения* заключается в том, что у некоторых нейронов следует повысить веса синаптических связей (скажем, положить равным единице, вместо нуля, окрасить в красный цвет). То есть некоторые нейроны могут стать преемниками возбуждения большего, чем прежде, числа нейронов. А если эти нейроны использовались для получения других решений?

Здесь важно сделать Замечание в дополнение к алгоритму *трассировки*, рассмотренному в лекции 4:

целесообразно при переиспользовании нейронов, возникающем при *последовательном анализе обобщенных эталонов*, фиксировать информацию: участвует ли данный нейрон лишь в одном *пути возбуждения* - к единственному нейрону выходного слоя и к какому именно, или к более чем к одному нейрону выходного слоя. И этого будет вполне достаточно для ликвидации тревог.

Продолжим рассмотрение примера, ведущее к обобщению.

Для нового *обобщенного эталона* на основе матрицы S , так же, как мы это делали в предыдущем разделе, построим (рис. 5.3) матрицу $S^* [B1, A1, C1, C2, C3, C4 \rightarrow \text{Вых}1]$.

"Спускаемся" по столбцу, соответствующему нейрону $C1$, и в строке, не являющейся входом, находим, если таковой имеется, первый нулевой элемент. Полагаем его равным единице и отражаем введенное изменение в матрице S . Если нулевого элемента в столбце не нашлось, находим первый "пустой" элемент (или отмеченный *транзитивной связью* - все равно) и полагаем его равным единице. Этим мы вводим дополнительную связь, отраженную в матрице S . В данном случае такой связью с единичным весом является связь $C1 \rightarrow 2$.

То же проделываем с нейроном $C3$, введя дополнительную связь с единичным весом $C3 \rightarrow 2$ (рис. 5.5).

$B1$											
$A1$											
$C1$											
$C2$											
$C3$											
$C4$											
2	1	1	1	1							<i>Вых1</i>
9									1		<i>Вых1</i>
<i>Вых1</i>										1	

Рис. 5.5. Шаг преобразования матрицы следования

Отметим, что наше решение основано на эвристике, и место появления новых возбужденных связей может быть оспорено. Например, могли быть построены связи $C1 \rightarrow 2$, $C3 \rightarrow 9$, или $C1 \rightarrow \text{Вых1}$, $C3 \rightarrow 9$, или $C1 \rightarrow 9$, $C3 \rightarrow \text{Вых1}$, и т.д.

Мы не приводим еще более усложнившегося рис. 5.1 уточненной сети, предоставив читателю возможность проверить усвоение материала с помощью легкого нажатия красным карандашом прямо в книжке в направлении стрелок, которые исходят из кружочков, помеченных символами $C1$ и $C3$, и входят в кружок, помеченный цифрой 2. Однако намек на такое уточнение делаем на рис. 5.6.

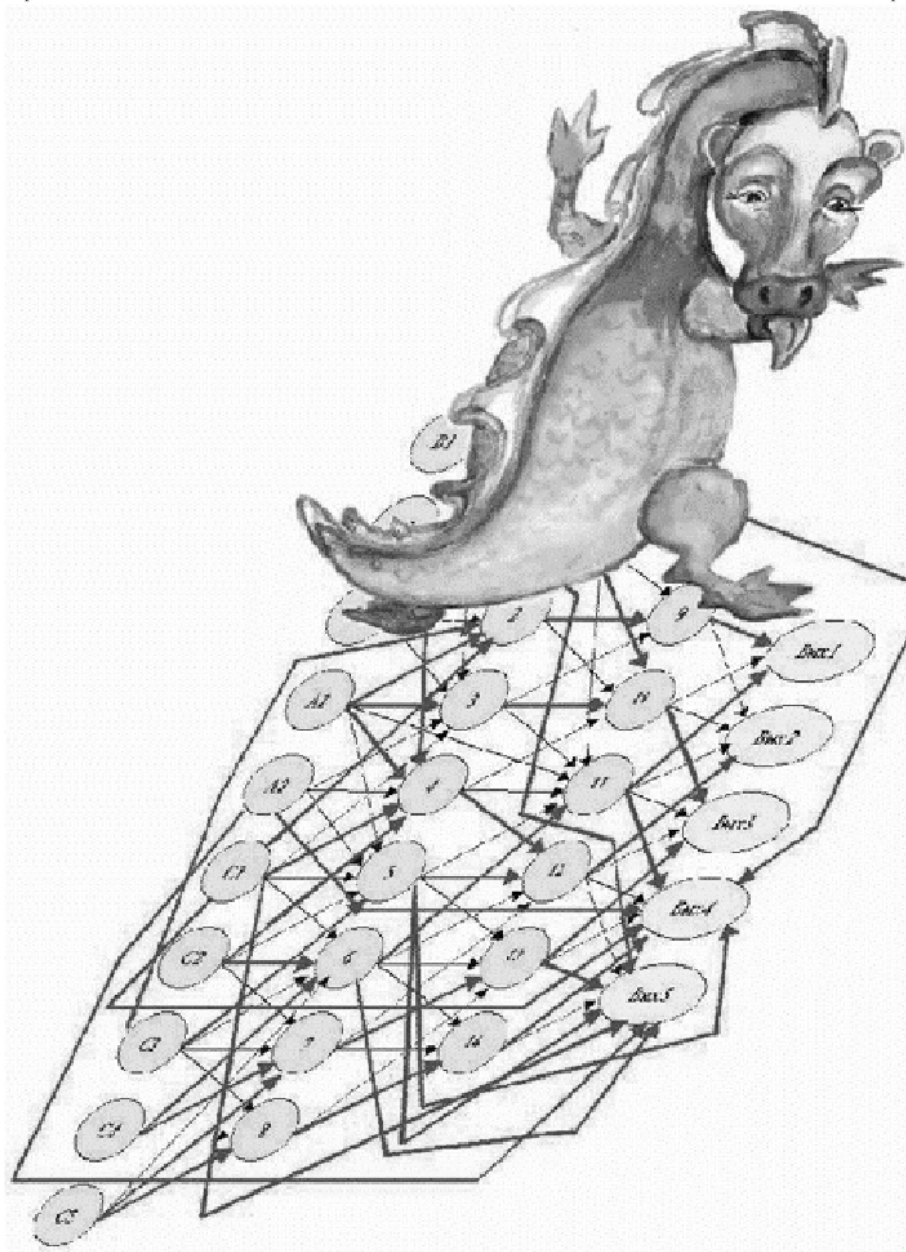


Рис. 5.6. Полностью обученная нейросеть (намек)

5.2. Не задавайте глупых вопросов!

"Родила царица в ночь

Не то сына, не то дочь . . . "

А.С. Пушкин

Мы учим сеть по *обобщенным эталонам*, несомненно опираясь на здравомыслие пользователя. Так, обучая *обобщенному эталону* $A1 \& B2 \& B3 \& C1 \& C2 \& C3$, мы предполагаем задание "логичных" вопросов типа: "Какое решение принять, если Вася на своем прекрасном "Volvo" отправился в павильон "Роксана", в который накануне завезена из Китая большая партия косметики?". Вы задаете сети ясный вопрос, полагая $A1 = B3 = C2 = 1$ (на самом деле вы знаете об истинном происхождении косметики), и получаете столь же ясный ответ: "Принимай решение (или значение твоей прибыли) $R2!$ "

Но что, если при сформировавшихся связях задать вопрос $B3 = C1 = C2 = C3 = 1$? Максимального и равного возбуждения на выходе достигнет даже не один нейрон! Но что означал этот вопрос и что означал ответ? Судить ли по неоднозначности ответа о том, что на входе сформировано "объединение" вопросов, обусловившее неоднозначность ответов? (Каков вопрос - таков ответ?) А всегда ли получается неоднозначный ответ?

Зададим вопрос, положив $A2 = B1 = B2 = 1$. Максимального возбуждения достигнет нейрон $Вых5$. Но предусматривалось ли решение $R5$ в ответ на такую ситуацию? Хотя завоз продукции фирм Красный киллер и Пират в палатки $C1$, $C2$ и $C3$ накануне визита Пети вполне возможен!

Несомненно, для правильного ответа надо правильно ставить вопрос. Корректность использования нейросети должна быть обусловлена теми правилами, которые брались за основу ее разработки. Если мы положили, что события A , B , C обязательно должны участвовать в формировании вопросов, то для задания других вопросов сеть необходимо дополнительно обучить.

Например, вопрос "Фирма $B3$ направила свою продукцию в палатки $C1$, $C2$ и $C3$, - что делать?" требует тщательной проработки нового ответа $R6$ на свободном нейроне $Вых6$ выходного слоя и формирования (*трассировки*) пути возбуждения $B3$, $C1$, $C2$, $C3$

-> Вых6.

Однако возможности неоднозначного ответа (одинаковой величины возбуждения нейронов выходного слоя) этим избежать не удастся. Придется анализировать всю картину возбуждения нейронов выходного слоя и выдавать все ответы вида: "Продукция фирмы В3 поступила в палатки С1, С2, С3 "; "Это впоследствии, скорее всего, приведет к решениям R2 и R4 ". А это, нетрудно видеть, влечет необходимость надстройки сети вторым логическим уровнем, где выходной слой нейронов первого уровня становится (возможно, в совокупности со своим входным слоем) входным слоем. Так могут быть продолжены логические цепочки наших умозаключений.

Это еще раз свидетельствует о том, что нельзя законсервировать себя как носителя нейросети на всю жизнь, получив стандартный объем знаний, удостоверенный "красным" дипломом. Надо думать, надо учиться, надо развиваться, смело задавая "запрещенные", глупые вопросы . . .

Как показал исторический опыт, демократия и гласность без запретов кроме вреда ничего не приносят. Или это не так? Или именно запрещенные вопросы милы Развитию? Рассмотрим это подробнее ниже.

5.3. Познание нового - основа самообучения

"Кто стрелял, куда попал?"

А. Твардовский, "Василий Теркин"

Конечно, вы сами можете открыть квантовую механику. Но лучше спросить "Что это такое?" за кружкой пива у бедного студента, подрабатывающего грузчиком в соседнем гастрономе.

Мы научили сеть множеству эталонов - даже сформированных в обобщенные эталоны. Теперь, подавая на вход даже искаженные образы, мы можем ответить на вопрос "На какой эталон в большей мере

он похож?". И максимально возбужденный нейрон ответит нам на него. Рассмотрим, однако, более внимательно пороги, которым мы уделяли недостаточное внимание.

Что если, более серьезно отнесясь к порогам, установить ограничение на величину возбуждения нейрона выходного слоя, столь решительно дающего определенный ответ? А именно: если величина максимального возбуждения меньше порога H , следует считать, что сеть не знает, что за образ подан на ее вход, и вправе рассматривать его как новый эталон для обучения.

Тогда должен быть найден "свободный" нейрон выходного слоя, и по вышеприведенному алгоритму должна быть произведена *трассировка*.

Так производится запоминание эталона как некоторой диковинки. Для последующего использования новых знаний трудно обойтись без внешнего воздействия, без учителя, ибо неизбежен вопрос: "Что это и что из этого следует?". Мы знаем, насколько длителен путь эволюции, основанный на "чистом" *самообучении*, без воздействия, влияния, поддержки извне. Пример Маугли нас не вдохновляет.

Мы даже можем представить себе возможный диалог между Нейрокомпьютером (НК) и Учителем (У):

НК (Величина возбуждения ни одного нейрона выходного слоя не одолела порога):

- Это что-то новенькое! Повтори вопрос; может, величины возбуждения на входе малы?

У:

- Повторяю вопрос.

НК (Результат тот же):

- Все же это что-то новенькое. Приводит ли это к одному из известных решений, или решение новое?

У (Варианты):

1. - Это должно привести к решению R5.
2. - Это новое решение R6.

НК (Соответствующие варианты):

1. - Произвожу *трассировку* по уточненному *обобщенному эталону* для решения R5.
2. - Рассматриваю твой вопрос как новый *обобщенный эталон*. Произвожу *трассировку* к *выходному нейрону* Вых6, соответствующему решению R6.

(Пауза)

Здесь идет *самообучение* системы "Учитель - *Нейрокомпьютер*". Полное или близкое к нему самостоятельное обучение *Нейрокомпьютера* возможно тогда, когда сеть - многоуровневая, то есть одни выводы являются посылками для других. Оно возможно, если сеть знакома хотя бы с понятиями "хорошо" и "плохо", то есть учитывает критериальную функцию, так необходимую для моделирования. Это уже высшие сферы обучения, которые мы наблюдаем пока в природе и лишь экспериментально, но постепенно надеемся покорить на основе изложенных здесь предпосылок.

Мы уверены, что любая достаточно развитая нейросетевая система должна быть многоуровневой, допускающей цепочки выводов. Иначе трудно представить себе такие диалоговые системы, с помощью которых производится постепенное уточнение при успешном продвижении к Большой Правде.

Например, диалог при медицинской диагностике формируется, несомненно, на основе отработанной стратегии допроса. Предполагается, что вопросы структурированы и даже образуют порядок следования, отраженный графически, так, что одни вопросы исключают другие. Например, уместен ли вопрос "Снились ли вам гномики?" после отрицательного ответа на вопрос "Спали ли вы в эту ночь?" - при рецидивах недержания?

В то же время принцип нейросети обязан выручить тогда, когда вопросы нелогичны, несовместимы. Просто все нейроны выходного

слоя не превысят порога, и НК вступит в диалог с Пользователем (П), подобный предыдущему. Если же ответ на логичный вопрос не был предусмотрен экспертами (Учителем), то этот недостаток должен быть устранен *трассировкой* нового решения.

Правда, новым решением - реакцией на этот вопрос - может быть ответ: "Не задавай глупых вопросов!"

В общем случае диалог Пользователя с Нейрокомпьютером может выглядеть следующим образом.

НК (Первый вопрос после входа):

- Что у вас болит?

П:

- Голова, живот, пальчик.

НК (После ввода и обработки сигналов - величин возбуждения нейронов входного слоя, соответствующих понятиям "голова", "живот", "левый указательный палец", максимально возбуждается нейрон промежуточного выходного слоя, инициирующий следующий вопрос):

- Ковыряли ли вы в носу на сон грядущий?

П:

- Да.

НК:

- Мыли ли вы руки, ложась спать?

П (Варианты):

1. - Я мою только ноги.
2. - Нет.

НК (Варианты):

1. - Не задавайте глупых вопросов!
2. (На правах рекламы) - Не волнуйтесь, но вы страдаете манией величия. Вам срочно необходимо обратиться в Российский Пенсионный Фонд!

В этом месте остановись, читатель, и отрешенным взором обозначь ясную мысль: "Так может ли существо, называемое Божьей Тварью, жить и действовать одним лишь мозгом - нейросетью в механическом, бессознательном состоянии? Достаточен ли мозг в такой ограниченной интерпретации аппарата, инструмента, - для творчества и самопознания? Или все нами изложенное не достойно даже червячка? Есть ли в нас что-то выше такого мозга?" и т.д. и т.д.

Нет, не хочется чувствовать себя только думающей машиной! - И мы размашисто осеняем себя крестным знаменем...

Нейронные сети с обратными связями

Обсуждается вопрос введения в нейронную сеть обратных связей как элемента самонастройки, уточнения возможной ситуации и на основе этого – выбора целесообразного решения. Предлагаются некоторые правила введения обратных связей.

6.1. К вопросу происхождения человека

"...если бы человек, властитель мира, умнейшее из дыхательных существ, происходил от глупой и невежественной обезьяны, то у него был бы хвост и дикий голос. Если бы мы происходили от обезьян, то нас теперь водили по городам цыганы на показ и мы платили бы деньги за показ друг друга, танцуя по приказу цыгана или сидя за решеткой в зверинце... Разве мы любили бы и не презирали бы женщину, если от нее хоть немножко пахло бы обезьяной, которую мы каждый вторник видим у Предводителя Дворянства?"

А.П. Чехов. "Письмо к ученому соседу"

Тезисы "человек произошел от обезьяны" и "человек произошел не от обезьяны", в основе которых лежит обидная похожесть, по-видимому, одинаково неверны, ибо из закона *отрицания отрицания* следует, что правда лежит где-то посередине. Вероятно, два процесса успешно сосуществовали: один – в соответствии с непреложностью идеи Развития, другой – в соответствии с неотвратимостью внешнего цивилизованного воздействия.

В схоластической защите лишь одного из этих тезисов сказывается отсутствие обратной связи: влияния сделанного вывода на исходную посылку. Наличие обратной связи способно отвергнуть посылку, произвести доказательство "от противного" так, как это блестяще выполнил Василий Семи-Булатов.

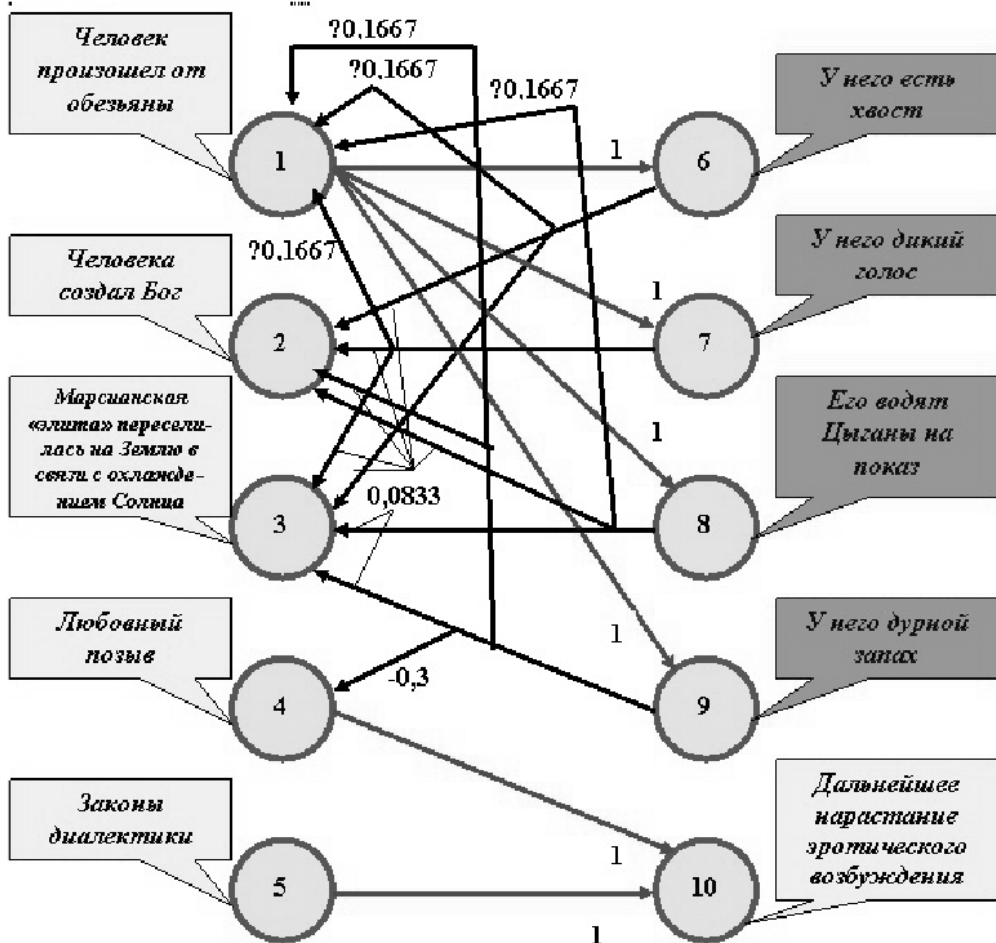


Рис. 6.1. Нейросеть для решения вопроса о происхождении человека

Но не только сделанный вывод должен влиять на исходную посылку. На нее посылку должны влиять и другие выводы, в совокупности определяющие знания. (Ведь вспомнили же мы закон отрицания отрицания!) Обратные связи без знаний могут свести на нет весь блеск интеллекта и привести только лишь к показанному великим писателем каламбуру.

Попробуем построить проект, эскиз, набросок фрагмента нейросети, отражающей глубину умозаключений чеховского героя и дополненной доступными нам понятиями более высокого свойства (рис. 6.1).

При построении нейросети учитываем следующее.

Посылка "человек произошел от обезьяны" обязательно должна привести к абсурдному выводу "у человека есть хвост" (нейрон 6). Тогда обязательно должна существовать *отрицательная обратная связь*, которая ведет к нейрону 1 *входного слоя*, "ответственному" за указанный тезис, и снижает заданную извне величину его возбуждения. Это значит, что этот нейрон должен получать возбуждение нейрона выходного слоя с помощью дендрита, обладающего отрицательным синапсическим весом. *Отрицательная связь* помечена на рисунке знаком "-".

Примечание. Ранее мы отличали нейроны-рецепторы *входного слоя* от других нейронов. Рецепторы возбуждались только извне. Теперь мы видим, что это такие же нейроны, как и все другие. Просто некоторые их дендриты получают возбуждение извне, а некоторые – от других нейронов сети.

Одновременно положительная обратная связь должна усиливать возбуждение нейронов 2 и 3, "ответственных" за тезисы о божественном и марсианском происхождении человека.

Такое же влияние на нейроны 1–3 должны оказывать и другие выводы, за которые "ответственны" нейроны 7–9.

Однако действенен и довод "То, что дурно пахнет, недостойно любви". Дурной запах давит, угнетает позывы любви, порождая *отрицательную обратную связь* – признак сомнений в происхождении человека.

На стрелках, обозначающих связи, указаны их веса ω_j , в том числе – отрицательные. Выбор весов обусловлен тем, что три события-высказывания, соответствующие нейронам 1–3, образуют *исчерпывающее множество событий*. Тогда, в соответствии с условием нормировки, уменьшение достоверности одного из них должно увеличивать достоверность других.

Выберем передаточную функцию

$$V := \sum_j \omega_j V_j$$

$$V_i := \text{if } V > h_i \text{ then (if } V < 1 \text{ then } V \text{ else } 1) \text{ else } 0$$

Выберем значения порогов равными нулю для нейронов входного слоя и равными 0,33 для остальных нейронов, формирующих обратные связи.

Зададим значения возбуждения нейронов входного слоя, соответствующие критикуемым положениям теории Дарвина, а также любовным настроениям и широте философских познаний чеховского героя.

А именно, пусть $V_1 = 1$, $V_2 = V_3 = 0$, $V_4 = V_5 = 0,5$.

При этом оценку величины любовного позыва связываем с напряженной умственной деятельностью, а величину возбуждения нейрона 5 – с самокритичной оценкой познаний и доверия к столь заумным философским положениям.

По заданным значениям произведем первый цикл обсчета величины возбуждения остальных нейронов: $V_6 = \dots = V_{10} = 1$.

При втором цикле обсчета получаем $V_1 = V_2 = V_3 = 0,33$, $V_4 = 0,2$ при неизменившихся значениях величин возбуждения других нейронов.

Третий цикл обсчета позволяет уточнить значения $V_6 = V_7 = V_8 = V_9 = 0$, $V_{10} = 1$.

Видим, что значения возбуждения нейронов, порождающих обратные связи, более не влияют на величину возбуждения нейронов входного (он же выходной) слоя.

Таким образом, сеть "высказалась" за равную действенность всех возможных утверждений о появлении человека на Земле.

Более того, если в начале исследований мы уже недостаточно уверены в предположении, что человек произошел от обезьяны, т.е. положили $V_1 < 1$, то это лишь усиливает правоту предположения о божественном или марсианском его происхождении.

Например, положим $V_1 = 0,8$, $V_2 = V_3 = 0,1$.

В результате первого же уточнения этих значений получим $V_1 = 0,267$, $V_2 = V_3 = 0,367$. И лишь возбуждение нейрона 10 остается неизменным, утверждающим тезис о том, что философское отношение к действительности благотворно влияет на физиологические возможности.

Конечно, мы сплутовали: мы получили то, что хотели. Совсем в духе *силлогистики* Аристотеля, призванной обслуживать софистику тогдашних философов. Но ведь всегда высшая идея довлеет над нашими решениями.

Построенная *нейросеть* с ее параметрами является лишь фрагментом, первым приближением истинной, полной, учитывающей взаимодействие всех возможных факторов сети. Построение такой сети должно опираться не только на законы диалектики, но и на теорию познания, на логические приемы индуктивного и дедуктивного мышления.

Например, мы не учли, что божественное происхождение отнюдь не гарантирует отсутствие дурного запаха и даже отсутствие хвоста. А это тотчас требует введения и положительных, и *отрицательных обратных связей* и т.д. Тогда, возможно, обеспечивалась бы постоянная и одинаковая сходимости величин возбуждения. Сеть давала бы один и тот же ответ, сходилась бы к нему, независимо от начальных предположений.

В то же время как с позиций диалектики согласовать правомочность всех трех предположений о происхождении человека?

Напрашивается следующая модель. Бог, конечно же, создал человека, но – на Марсе! В результате охлаждения Солнца марсианская "элита", в лице отнюдь не лучших представителей рабочего класса и трудовой интеллигенции, воспользовавшись высокими достижениями в области науки и техники (несомненно – и в области искусственного интеллекта), перебралась на Землю, где успешно спуталась с местными обезьянами.

Тогда вполне справедливо предположение о том, что в скором времени, по мере дальнейшего остывания Солнца, та формирующаяся новая

"Элита", что уже сейчас приобретает жилье по четыре тысячи баксов за кв. м, переберется на Венеру, где ее с нетерпением будут ждать тамошние обезьяны, вдохновленные дармовой продукцией фирм Красный Киллер и Ночная Бабочка.

6.2. Как же вводить обратные связи?

"Я дух родного твоего отца,
На некий срок скитаться осужденный
Ночной порой, а днем гореть в огне,
Пока мои земные окаянства
Не выгорят дотла".

В. Шекспир. "Гамлет, принц датский", перевод Б. Пастернака

"... все они считают меня дурным человеком. По утрам и с перепою я сам о себе такого же мнения. Но ведь нельзя же доверять мнению человека, который еще не успел похмелиться! Зато по вечерам – какие во мне бездны! – если, конечно, хорошо набраться за день, – какие бездны во мне по вечерам!"

Венедикт Ерофеев. "Москва–Петушки"

Рассмотрим непрерывную, динамически контролируруемую работу уже обученной сети в реальном времени.

Пусть дядя Рамзай своими собственными глазами видел Васю, беседующего с Оксаной близ ее палатки. И было это не более пяти минут назад. Ясно, что по крайней мере еще минут на двадцать Васю следует исключить из процесса оценки достоверности последующих ситуаций. Желательно этот процесс автоматизировать. Это значит, что на входном слое должна быть учтена информация "Вася уже у Оксаны, следовательно, дальнейшее рассмотрение комбинаций с их участием нецелесообразно". И если вдруг появившийся Вовочка, прося на пряник, сообщает, что только что видел Васю у св. Аполлинарии, то

дядя Рамзай, корректируя исходную информацию на текущий момент, должен игнорировать или учесть незначительно (в случае неверия собственным глазам) информацию Вовочки. Мы хотели бы, чтобы достоверность такой информации, с участием Васи и Оксаны, автоматически снижалась. Либо, наоборот, достоверность ситуаций без их участия должна увеличиваться.

И здесь мы можем не гоняться за достоверностью, требуя принадлежности ее величины диапазону $[0, 1]$, а перейти к более общему понятию *веса информации*. Этот вес необходимо рассматривать как функцию времени. Ведь логично, что с течением времени вероятность участия Васи и Оксаны в последующих ситуациях должна как бы восстанавливаться.

Что необходимо для такой работы сети?

1. Надо с большей определенностью дифференцировать ситуации на выходном слое. Например, если в решениях R_1 и R_2 замешаны не только Вася и Оксана, но и другие участники игры, то это затрудняет принятие конкретного решения только по Васе и только по Оксане. Нам необходимо корректировать дальнейшее решение по вхождению комбинации $A_1 \& C_1$, и не связывать, не объединять его, например, с вхождением комбинации $A_1 \& C_2$. Значит, надо расширить множество $\{R_i\}$ принимаемых решений, повторяя ("размножая", см. [лекцию 1](#)), если необходимо, одно и то же решение для разных эталонных комбинаций.
2. Необходимо ввести обратную связь, осуществляющую влияние решения, осуществляемого по Васе (A_1) и Оксане (C_1), на достоверность или *вес информации*, связанные с их дальнейшим участием в складывающихся ситуациях, то есть на величины возбуждения нейронов A_1 и C_1 , даже вопреки попыткам искусственно задать высокие значения этих величин.

Конкретными рекомендациями могут быть:

1. Детализация, дифференциация эталонных ситуаций, по которым принимается не только "прямое" решение R_i , но и "обратное" решение, корректирующее характер использования исходных

данных.

2. Установление *отрицательной обратной связи*, которая ведет от каждого нейрона, отвечающего за решение по Васе, к нейрону A1 входного слоя.
3. Установление *отрицательной обратной связи*, которая ведет от каждого нейрона, отвечающего за решение по Оксане, к нейрону C1 входного слоя.
4. Установление "положительной" обратной связи от нейронов выходного слоя, отвечающих за решения по Васе, к нейрону A2, что усиливает шансы Пети в дальнейшей игре дяди Рамзая.
5. Установление "положительной" обратной связи от нейронов выходного слоя, отвечающих за решения по Оксане, к нейронам C2, . . . , C5. (Однако этого, как и указанного в пункте 4, можно и не делать.)

Строго следуя данным рекомендациям, на основе "размножения" решений, рекомендуемого в [лекции 1](#), построим более подробную систему предикатов, которую мы воплотим с помощью нейросети:

$A1 \& B1 \& C1 \rightarrow R1$	$A2 \& B3 \& C1 \rightarrow R4$
$A1 \& B1 \& C2 \rightarrow R1$	$A2 \& B3 \& C2 \rightarrow R4$
$A1 \& B1 \& C3 \rightarrow R1$	$A2 \& B3 \& C3 \rightarrow R4$
$A1 \& B1 \& C4 \rightarrow R1$	$A2 \& B3 \& C4 \rightarrow R4$
$A1 \& B1 \& C5 \rightarrow R1$	$A2 \& B3 \& C5 \rightarrow R4$
$A1 \& B2 \& B3 \& C1 \rightarrow R2$	$A2 \& B1 \& B2 \& C1 \rightarrow R5$
$A1 \& B2 \& B3 \& C2 \rightarrow R2$	$A2 \& B1 \& B2 \& C2 \rightarrow R5$
$A1 \& B2 \& B3 \& C3 \rightarrow R2$	$A2 \& B1 \& B2 \& C3 \rightarrow R5$
$A1 \& B2 \& B3 \& C4 \rightarrow R3$	$A2 \& B1 \& B2 \& C4 \rightarrow R5$
$A1 \& B2 \& B3 \& C5 \rightarrow R3$	$A2 \& B1 \& B2 \& C5 \rightarrow R5$

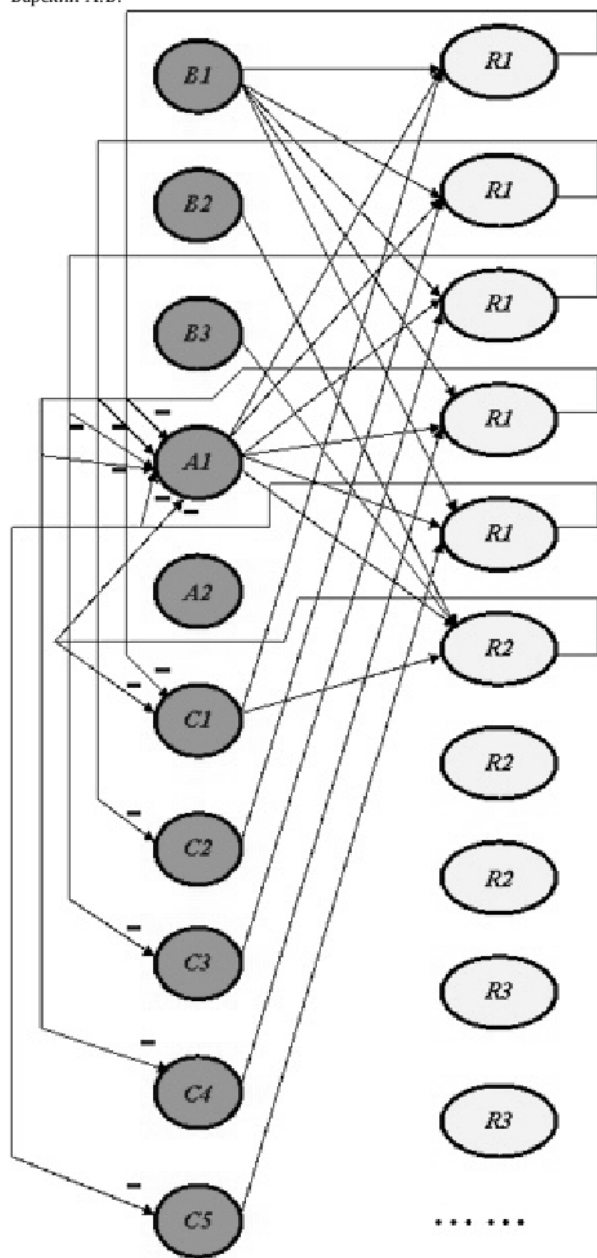


Рис. 6.2. Нейросеть с обратными связями

Мы видим, как резко возросло число нейронов выходного слоя. Что ж, нас теперь интересуют не просто выводы, но и то, каким путем каждый из них получен. Это еще раз свидетельствует о том, что жадничать, в

том числе составляя обобщенные эталоны, объединяя при этом ситуации, ведущие к одинаковому выводу, – нехорошо. Это может серьезно связывать руки при необходимости различения ситуаций. Вряд ли это жадничание правильно отражает принцип мозга, где царит избыточность, обеспечивающая обилие путей и вариантов, развиваемость, резервирование, устойчивость и надежность.

На рис. 6.2 представлен фрагмент, демонстрация начала построения возможной нейросети, которую мы вправе считать окончательным результатом наших исследований и готовы предъявить Заказчику. Чтобы сократить путаницу, мы не стали вводить положительные обратные связи.

Следует обратить внимание на то, что, вводя обратные связи, мы ограничились только лишь проблемой увеличения достоверности исходных данных на основе текущего состояния событий, предоставив читателю право рассуждения о других возможностях.

Модуль веса обратной связи в нашем случае должен быть убывающей функцией времени. Пользуясь допустимыми условностью и приближенностью, преследуя цель скорее качественную, нежели количественную, можно рассчитать величину *отрицательной обратной связи* по простой формуле. Например, если считать, что на один визит Вася расходует не менее 25 минут, то вес ω одного из дендритов нейрона А1 (аналогично – других нейронов *входного слоя*) находится следующим образом:

$$\omega = \begin{cases} k \cdot \frac{t - t_0 - 25}{25}, & \text{при } t_0 < t \leq t_0 + 25 \\ 0, & \text{в противном случае} \end{cases}$$

Здесь:

t_0 – последний момент времени вовлечения соответствующей особы в действие (хранится в составе информации о данной особе – А1, А2, С1 и др.);

$t_0 + 25$ – ожидаемое время повторного допустимого "использования" особы в деле;

t – текущий момент поступления новых разведданных, требующих использования нейросети;

k – некоторый экспериментально подобранный демпфирующий коэффициент, возможно, зависящий от величины подаваемого возбуждения при вводе информации, чтобы разумно "погасить" влияние обратной связи, не довести ее до абсурда.

Напомним, хотя это не имеет принципиального значения, что мы используем понравившуюся нам передаточную функцию для нахождения величины V_i возбуждения i -го нейрона:

$$V := \sum_j \omega_j V_j$$

$$V_i := \text{if } V > h \text{ then } V \text{ else } 0$$

Нейросетевые самообучающиеся системы управления

Исследуется возможность применения логических нейронных сетей для построения самообучающихся систем управления. В основе принципа самообучения лежит накопление знаний о возможных ситуациях и о принимаемых по ним решений. Моделируется принцип ассоциативного мышления интерполяцией знаний, накапливаемых в таблицах.

"Животное полагает, что все его дело – жить, а человек жизнь принимает только за возможность что-нибудь сделать".

А.И. Герцен. "Кто виноват?"

7.1. Самообучение на основе ситуационного управления

Известной основой аппаратной поддержки *самообучающейся системы* управления является *ассоциативная вычислительная система* .

Главное отличие *ассоциативной ВС* от обычной системы последовательной обработки информации состоит в использовании *ассоциативной памяти* или подобного устройства, а не памяти с адресуемыми ячейками.

Ассоциативная память (АП) допускает обращение за данными на основе их признака или ключевого слова: имени, набора характеристик, задания диапазона и т.д.

Распространенный вид АП – таблица с двумя столбцами: " *запросное поле – ответное поле* ". Строка таблицы занимает регистр памяти. По ключевому слову обрабатываются *запросные поля*: производится поиск на основе сравнений и выдается результат из одного (или более) *ответных полей*. При помощи маскирования можно выделить только те поля в ключевом слове, которые нужно использовать при поиске для сравнения.

Типичными операциями сравнения, выполняемыми *ассоциативной памятью*, являются: "равно – не равно", "ближайшее меньше чем –

ближайшее больше чем", "не больше чем – не меньше чем", "максимальная величина – минимальная величина", "между границами – вне границ", "следующая величина больше – следующая величина меньше" и др. Т.е. все это – операции отношения и определения принадлежности.

Поскольку *ассоциативные ВС* характеризуются только активным использованием АП в вычислениях, то в целом эти ВС обладают обычными свойствами, могут производить сложные преобразования данных и принадлежать типу ОКМД (STARAN, PEPE) или МКМД. Для параллельного обращения (для ускорения поиска) АП разбита на модули (32 модуля – в STARAN).

Когда в 1980 г. был провозглашен т.н. "Японский вызов" о построении ВС сверхвысокой производительности, то одним из пунктов была указана необходимость самого широкого использования принципов самообучающихся систем – систем, способных накапливать опыт и выдавать результат решения задачи без счета самой задачи, – на основе ассоциации и интерполяции (экстраполяции). Это значит, что применение *ассоциативных ВС* неотделимо от проблемы искусственного интеллекта.

Покажем, что ничего существенно нового в практику человеческого мышления ассоциативные машины не вносят, что это привычный способ использования интерполяционных таблиц, например таблицы логарифмов.

Предположим, мы пользуемся значениями функции $y = f(x)$. Мы можем запрограммировать счет этой функции на персональном компьютере, и каждый раз, когда нам надо, задаем x и запускаем программу, пользуясь прекрасным современным сервисом.

Предположим, та же функция сложна, а ее счет – важный элемент алгоритма управления в реальном времени. Решение приходит сразу: зададим эту функцию таблично, а для ускорения выборки включим в ВС *ассоциативную память*. Предусмотрим на ней операции, позволяющие производить простейшую интерполяцию. А именно, для данного значения x найти наибольший $x_1 \leq x$ и наименьший $x_2 \geq x$. Для них выбрать $f(x_1)$ и $f(x_2)$. Тогда вместо счета точного

значения $f(x)$ процессору остается произвести хотя бы линейную интерполяцию

$$\tilde{f}(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1)$$

надо только быть уверенным в достижении нужной точности.

Однако в повседневной жизни мы очень часто решаем *трудно формализуемые задачи*, подобно предложенной дядей Рамзаем, когда составить алгоритм решения совсем не просто. Но мы учимся, обретаем опыт, затем оцениваем складывающуюся ситуацию на "похожесть" и принимаем решение.

Простейшая задача: как мы автоматически определяем, на сколько надо повернуть рулевое колесо, чтобы остаться на дороге? Ведь первый раз мы въезжали в бордюр! Значит, некая таблица постепенно сложилась и зафиксировалась в нашем сознании.

Пожалуй, наиболее полно автоматически и с наибольшей практичностью решение задачи самообучения представлено в артиллерии, точнее – в правилах стрельбы.

После занятия огневой позиции подготовка установок для ведения огня занимает много времени. На рассчитанных установках по каждой цели производится пристрелка, когда цель захватывается в широкую вилку, затем по наблюдениям вилка "половинится" до тех пор, пока на середине узкой вилки не переходят на поражение. После стрельбы следует замечательная команда "Стой, записать...", по которой наводчик на щите орудия пишет номер цели и все пристрелянные установки по ней. Такая работа продлевается и по фиктивным целям – реперам. Постепенно при "работе" на данной местности запоминаются пристрелянные установки по многим целям и реперам. С их появлением и развитием подготовка данных по вновь появляющимся целям резко упрощается, т.к. сводится к переносу огня от ближайшей цели или репера, т.е. к внесению поправок по дальности и направлению. Доказано, что при этом достаточно сразу назначать захват цели в узкую вилку, т.е. пристрелка упрощается. Правила стрельбы существуют века, однако вряд ли кто-то осознавал

теоретически, что речь идет о реализации *самообучающейся системы*, у которой не существует аналога!

Для *трудно формализуемых задач* управления или для увеличения производительности вычислительных средств известно т.н. *ситуационное управление*, предложенное Д.А. Пospelовым [23]. Оно заключается в том, что для каждого значения вектора, описывающего сложившуюся ситуацию, известно значение вектора, описывающего то решение, которое следует принять. Если все ситуации отразить невозможно, должно быть задано правило интерполяции (аналог интерполяции, обобщенной интерполяции).

Пусть исходная ситуация характеризуется вектором $X = \{x_1, \dots, x_m\}$. По значению X , т.е. по его компонентам, принимается решение Y , также представляющее собой вектор $Y = \{y_1, \dots, y_n\}$.

(Значения X и Y могут определяться целыми и вещественными, булевыми. Изначально они могут иметь нечисловую природу: "темнее – светлее", "правее – левее", "ласковее – суровее" и т.д. Мы не будем рассматривать проблему численной оценки качественных или эмоциональных категорий.)

Предположим, для любых двух значений X_1 и X_2 , а также Y_1 и Y_2 определено отношение $x_i^{(1)} \leq x_i^{(2)}$ (или наоборот), $y_j^{(1)} \leq y_j^{(2)}$ (или наоборот).

Структура *ассоциативной памяти* и общий вид ее обработки показаны на рис. 7.1.

Пусть поступила входная ситуация X , для которой необходимо найти решение Y . В АП находятся два вектора X_1 и X_2 , минимально отличающиеся по всем координатам от вектора X . Для этих векторов там же записаны векторы решения Y_1 и Y_2 соответственно. Однако если для компоненты x_i выполняется условие $x_i \leq x_i^{(1)}$ ($x_i \geq x_i^{(1)}$), то желательно выполнение условия $x_i \geq x_i^{(2)}$ ($x_i \leq x_i^{(2)}$), $i = 1, \dots, m$. Эта желательность обусловлена преимуществом интерполяции по сравнению с экстраполяцией. Таким

образом, находится "вилка", которой принадлежит входная ситуация.

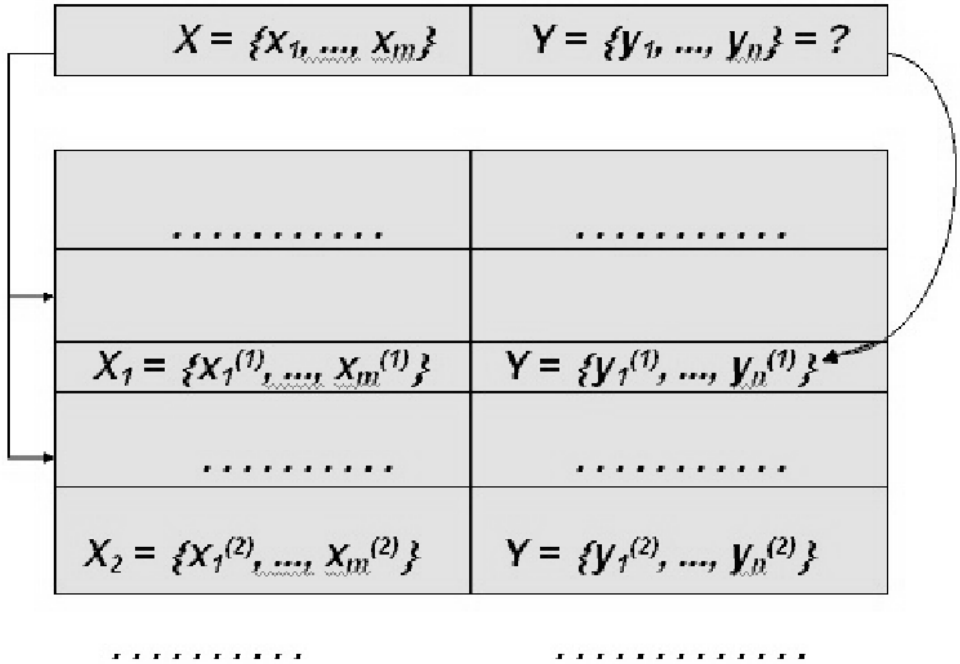


Рис. 7.1. Управление с помощью ассоциативной памяти

Тогда, опираясь на известные решения на границах этой вилки, необходимо выдать промежуточное решение для данной ситуации. Это можно сделать методом той же обобщенной интерполяции:

$$y_i = y_i^{(1)} + (y_i^{(2)} - y_i^{(1)}) \frac{r(X, X_2)}{r(X_1, X_2)}, i = 1, \dots, n;$$

$$r(X, X_2) = \sqrt{\sum_{j=1}^m (x_j - x_j^{(2)})^2}$$

Здесь $r(X, X_2)$ и $r(X_1, X_2)$ (находится аналогично) – расстояния между векторами X и X_2 , а также X_1 и X_2 .

Если известно, что точность Y достаточна, принципиально возможно дополнение АП новой строкой

<i>X</i>	<i>Y</i>
----------	----------

т.е. информацией о новом полученном опыте.

Однако целесообразная динамика развития и уточнения АП как базы знаний представляется иной. Далеко не всегда будет правильным развитие базы знаний только на основе ошибочного принятия решений. "Учение на ошибках" может привести к трагедии (или к частному срыву процесса управления), особенно на этапе обучения системы.

Обучение системы целесообразно проводить на достаточно точной модели, максимально использующей точный расчет компонент решения. Применение модели для обучения используется не только на специально предусмотренном этапе обучения системы, но и вне реального цикла управления, т.е. когда система работает в режиме дежурства, параллельно с функциональным контролем. Такой алгоритм ее работы представлен на рис. 7.2.

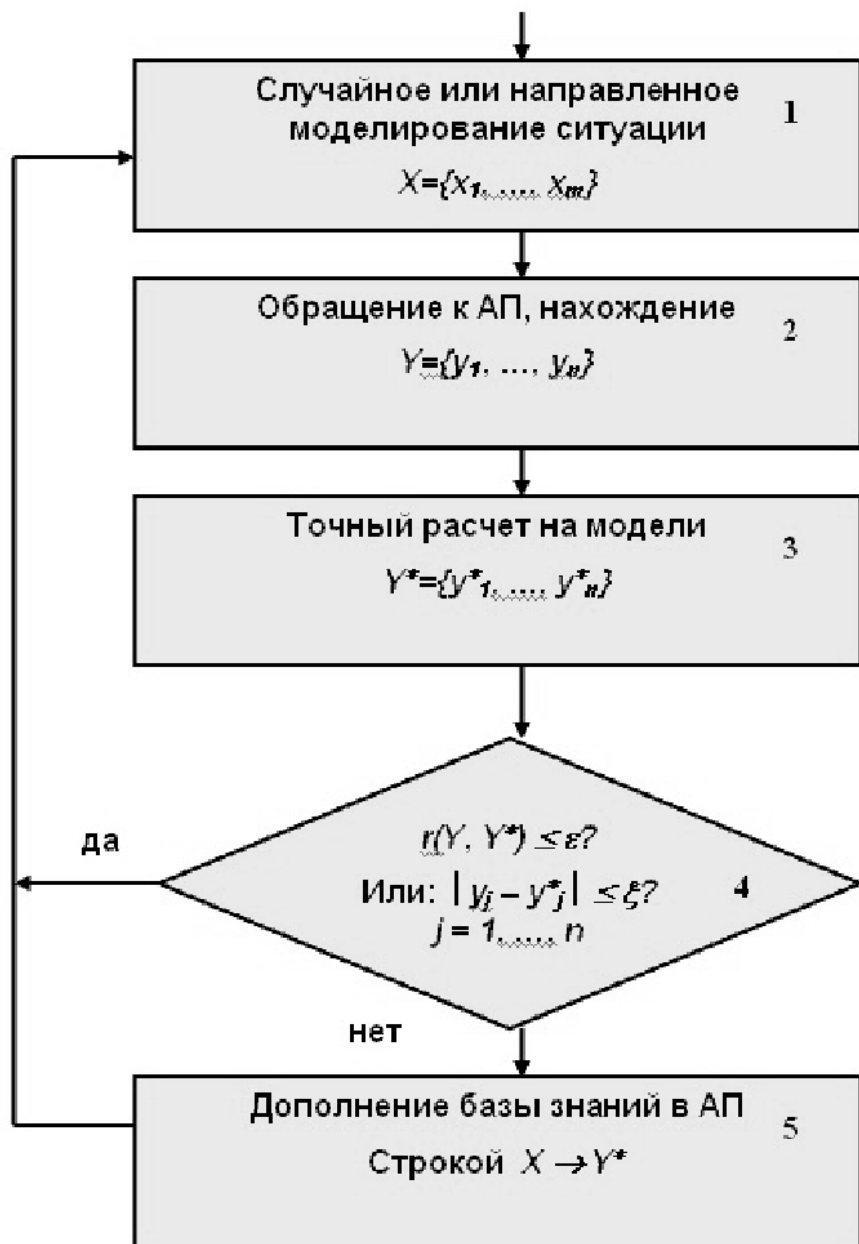


Рис. 7.2. Схема самообучения с ассоциативной памятью и моделью

7.2. Нейросетевое воплощение

" ... Медузин сел за письменный стол и просидел с час в глубокой

задумчивости; потом вдруг "обошелся посредством" руки, схватил бумагу и . . . написал:

1. *Российская грамматика и логика* *МНС*
употребл.
2. *История и география*
употребляет довольно
3. *Чистая математика*
. . . плох
4. *Французский язык*
виноградн. много
5. *Немецкий язык*
пива очень много
6. *Рисование и чистописание*
одну настойку
7. *Греческий язык*
все употребляет

После этих антропологических отметок Иван Афанасьевич записал соответственную им программу:

- Ведро сантурина
. 16 руб.
- 1/2 ведра настойки
. 8 "
- 1/2 ведра пива
. 4 "
- 2 бутылки меду
. 50 коп.
- Судацкого 10 бутылок
. 10 "
- 3 бутылки ямайского
. 4 "
- Сладкой водки шгоф

..... 2 " 50 коп.

Итого: 45 руб."

А.И. Герцен. "Кто виноват?"

Упражнение:

постройте нейросеть, позволяющую предсказывать действительные расходы учителя латинского языка и содержателя частной школы И.А. Медузина при оценке контингента гостей приближающегося дня тезоименитства.

Мы уже построили нейросеть для важной задачи принятия решений и вручили ее Заказчику – дяде Рамзаю. Однако, задумчиво глядя в синее небо под бодрящий рокот набегающей волны теплого течения Гольфстрим, мы не можем прогнать мысли об обобщениях, вариациях и применениях...

Итак, еще раз тщательно проследим принцип динамического совмещения процесса обучения системы управления с процессом выдачи решений, изложенный выше, и установим в схеме на рис. 7.2 целесообразность, место и способ использования нейронной сети как основного реального средства искусственного интеллекта. Прежде всего, мы понимаем, что основным естественным и целесообразным местом применения нейросети здесь является база знаний, ранее реализованная с помощью *ассоциативной памяти* (рис. 7.1). Нам только надо позаботиться о том, чтобы "крутились" там не численные значения информации, а вес этой информации или ее достоверность. То есть не сама информация должна обрабатываться нейросетью, а ее логический эквивалент. Ведь мы поняли, что принцип искусственного интеллекта направлен на параллельную обработку логических высказываний, а вовсе не ориентирован напрямую на числовую обработку информации, как это делается при решении некоторых "нейроподобных" задач при использовании нейросети в роли спецпроцессора.

Пусть нейросеть в роли базы знаний имеет текущий вид,

представленный на рис. 7.3.

Нейроны-рецепторы оказались закрепленными за значениями элементов входного вектора. Это значит, что в режиме обучения (или в процессе динамической корректировки) установилось соответствие на основе положения "величина возбуждения данного p -го нейрона входного слоя принимает значение достоверности того, что $x_i = x_i^{(p)}$ ".

Здесь могут фигурировать не точные значения параметров, а некоторые диапазоны их изменения, как это указывалось в 3.2.

Как мы ранее делали, при обучении мы полагаем эту достоверность равной единице, а в процессе распознавания может быть что угодно, даже не удовлетворяющее свойству полноты событий.

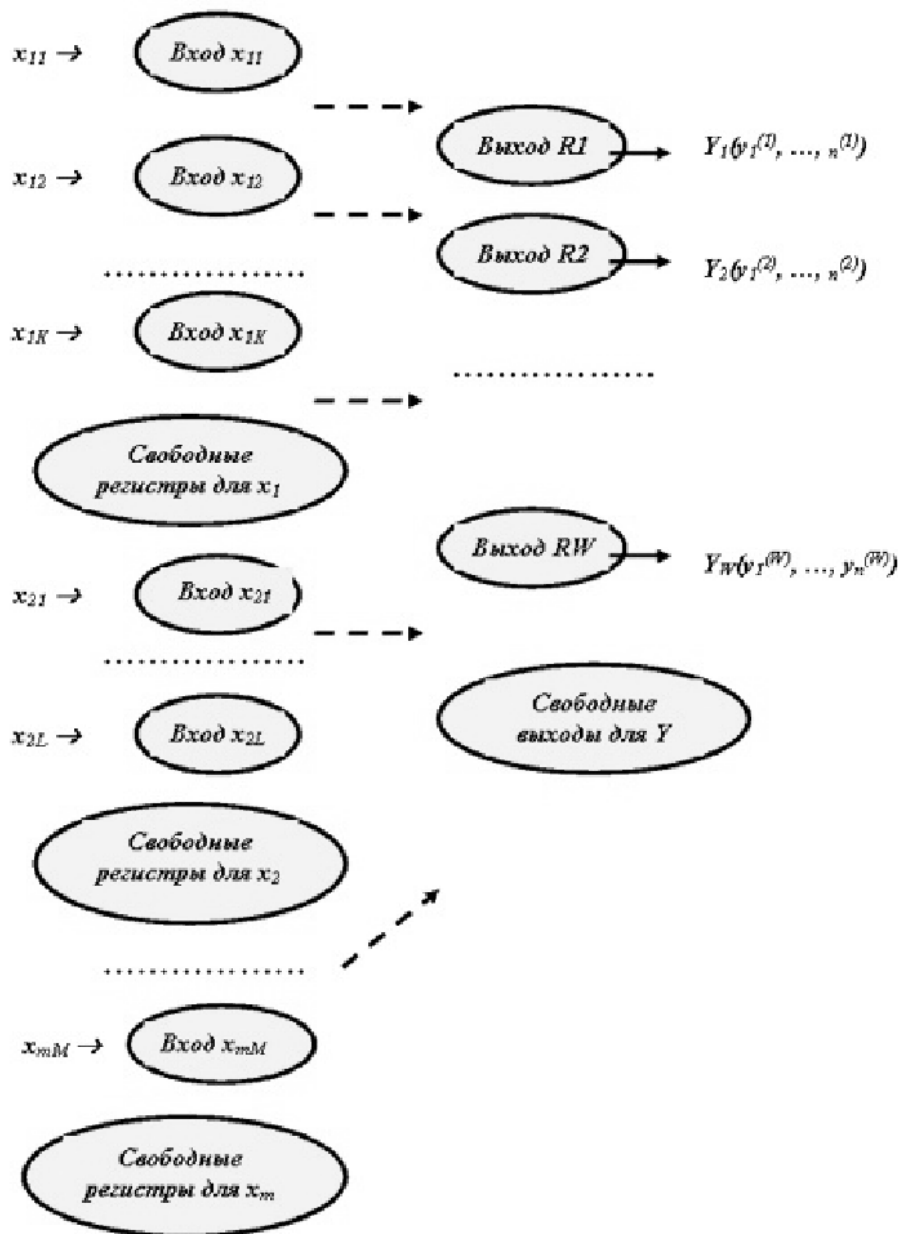


Рис. 7.3. Управляющая система с нейросетью

"Показываая" сети на этапе обучения различные эталоны, методом трассировки мы можем добиться установления соответствия вида $\{\text{Входы } x_i\} \rightarrow Rq$, наполнив тем самым базу знаний. Каждое

решение R_q при этом тоже указывает на определенное значение вектора – результата Y_q , компоненты которого суть управляющие воздействия системы. То есть нейроны R_q фактически закрепляются за ситуациями, по которым мы знаем векторы характеристик или значений параметров принимаемых решений – векторы значений управляющих воздействий.

Для нахождения этих значений управляющих воздействий в зависимости от значений исходных данных может быть использовано моделирование, наряду с экспериментом или с экспертными оценками. Таким образом, *модель*, эксперимент или эксперт играют роль учителя. (Мы ранее уже высказали сомнение насчет существования и действенности "чистого" самообучения.)

Первоначально обученная таким образом нейросеть может быть уже использована в рабочем режиме (распознавания) и в режиме совместной работы с *моделью*. При этом мы понимаем, что нейросеть обучена недостаточно, и, подобно пристрелке реперов "в свободное от работы" время, мы пытаемся с помощью *модели* испытать и, в случае необходимости, дополнить знания нейросети.

Модель случайно или направленно – по обоснованному плану "выбрасывает" некоторую ситуацию, характеризующуюся значением компонент входного вектора X . По каждой компоненте определяется вес или достоверность того, что ее значение совпадает с подмножеством значений этой компоненты, представленных входным слоем нейросети, или с диапазонами значений этой компоненты.

Например, сеть "знает" реакцию на значение (опустим индекс) $x = 2$, $x = 5$, $x = 6$. *Модель* выбросила значение $x = 5, 7$. Это может означать необходимость формирования значений возбуждения $V_{x=2} = 0$, $V_{x=5} = 0,3$, $V_{x=6} = 0,7$. Здесь индексы возбуждений указывают нейроны входного слоя, закрепленные за данным значением параметра. Так делается по всем компонентам входного слоя.

Пусть, "пройдя" по сети, данные возбуждения входного слоя привели к преодолению порога возбуждения нейронов выходного слоя R_5 ($V_{R_5} = a$), R_7 ($V_{R_7} = b$), R_{12} ($V_{R_{12}} = c$). При этом нейрону R_5

соответствует вектор управляющих воздействий $Y_5 = (Y_1^{(5)}, Y_2^{(5)}, \dots, Y_n^{(5)})$, нейрону R7 – вектор $Y_7 = (Y_1^{(7)}, \dots, Y_n^{(7)})$, нейрону R12 – вектор $Y_{12} = (Y_1^{(12)}, \dots, Y_n^{(12)})$.

Тогда мы находим предполагаемый ответ нейросети:

$$y_1 = \frac{ay_1^{(5)} + by_1^{(7)} + cy_1^{(12)}}{a + b + c},$$

.....

$$y_n = \frac{ay_n^{(5)} + by_n^{(7)} + cy_n^{(12)}}{a + b + c}$$

Подставляем данное решение в *модель* и устанавливаем, удовлетворяет ли нас оно, – например, по величине промаха. Если удовлетворяет, мы делаем положительный вывод об обученности нейросети и продолжаем ее испытание по другим исходным данным. Если нет – сеть необходимо "доучить", продемонстрировав самый высокий логический уровень обратной связи.

Для этого мы должны ввести в действие новые рецепторы, закрепив их за теми значениями исходных данных или их диапазонами, которые ранее не были представлены. Например, нам придется закрепить рецептор за значением $x = 5, 7$.

Далее, мы должны выделить нейрон выходного слоя и закрепить его за соответствующим правильным решением, которое мы должны получить с помощью *модели*.

Далее, мы должны выполнить трассировку для того, чтобы задание нового эталона с единичной достоверностью исходных данных приводило к максимальному возбуждению выделенного нейрона выходного слоя, ответственного за получение правильного решения.

Таким образом, сеть может обучаться до тех пор, пока не перестанет давать "сбоев". А поскольку в вероятностном аспекте это вряд ли возможно, то в таком режиме она должна работать в течение всего жизненного цикла, реализуя известную поговорку "Век живи – век учись"...

Здесь наглядно представлена замечательная возможность нейросети: *табличная аппроксимация функции многих переменных*, снабженная процедурой интерполяции (экстраполяции) для нахождения произвольного значения вектора-аргумента и приближенного соответствующего значения векторной функции. При этом входной вектор возбуждений рецепторов преобразуется в максимальное или усредненное значение возбуждения нейронов выходного слоя, указывающего на соответствующее значение вектора-функции. Так что практически столь простым способом мы построили аппроксимацию векторной функции от векторного аргумента!

Такую аппроксимацию можно выполнить и "в более явном виде", ибо каждая компонента y_j решения $Y = \{y_1, \dots, y_n\}$ может отыскиваться отдельно в результате предварительной трассировки (рис. 7.4). То есть сеть может быть построена и обучена так, чтобы заданное значение $X = \{x_1, \dots, x_n\}$ приводило к максимальному (или усредненному) значению возбуждения нейрона выходного слоя, указывающего соответствующее значение y_1 , к максимальному (или усредненному) значению возбуждения другого нейрона выходного слоя, указывающего на значение y_2 и т.д. Выходной слой оказывается разбит на области, каждая из которых закреплена за своим параметром y_i , $i = 1, \dots, n$. Тогда полученное преобразование можно условно записать $X \rightarrow \{\text{Вых}(y_1), \text{Вых}(y_2), \dots, \text{Вых}(y_n)\}$.

Следует обратить внимание не только на высокую производительность такого рода самообучающихся систем в рабочем режиме, но и на их адаптивность, развиваемость, живучесть и т.д.

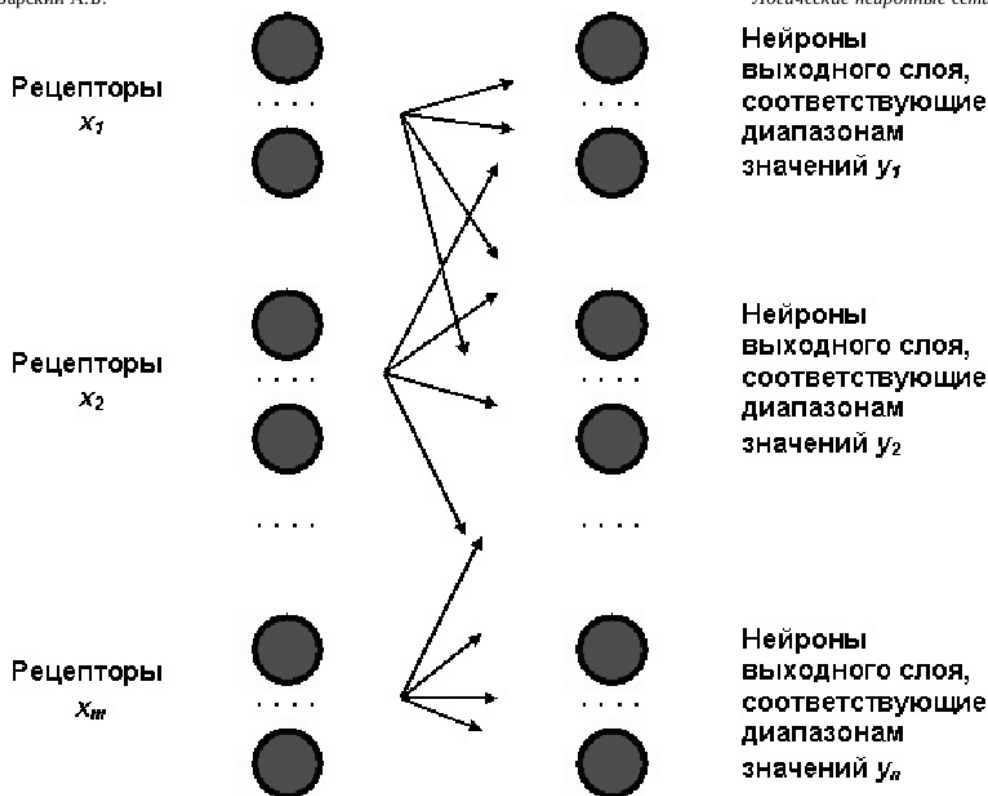


Рис. 7.4. Раздельное нахождение управляющих параметров с помощью нейросети

Логическое программирование нейронной сети

Показывается возможность логической нейронной сети производить логический вывод на основе языка ПРОЛОГ. Предполагается, что нейронная сеть не только способна реализовать дедуктивный вывод, но обладает и зачатками индуктивного мышления.

"Образование есть то, что остается после того, когда забывается все, чему нас учили".

А. Эйнштейн

8.1. ПРОЛОГ-программа

Следуя различными путями дедуктивного и индуктивного мышления, осуществляя различные парадигмы обучения, человек стремился автоматизировать логику мышления, немислимую без формализации. Продуктом этой деятельности явились такие языки логического вывода, как ЛИСП и ПРОЛОГ. Язык ПРОЛОГ следует считать венцом усилий по автоматизации логического вывода, эффективно описывающего, в частности, экспертные системы.

ПРОЛОГ представляет *базу знаний* как совокупность *фактов* и *правил* (вывода). Процедурная структура позволяет включать конструкции любых других алгоритмических языков. То есть он является логической надстройкой, сублимирующей лишь операции логического вывода. Формулируется *цель* логического вывода, и если она не противоречива, выявляются *факты*, из которых эта *цель* следует.

И сознавая, что Природа создала единственное средство мышления — мозг, мы снова и снова пристально раздумываем, как же реализовать то, что гениально воплощено в языке логического программирования ПРОЛОГ?

... Представьте себе село, затерянное в далекой таежной глуши. Навечно изолированные от Большой Жизни, его обитатели долгими зимними вечерами, в перерывах благотворного интеллектуального напряжения игры "Тигра идет", в сумраке демократических потуг вооружившись мозолистыми кулаками, выясняют степень взаимного

родства.

И тут являетесь вы! Словно светоч озарения, сосланный за непримиримость свободолюбивых устремлений, вы, наконец, находите для себя непаханое поле действительно яркой деятельности, полной гуманизма и самопожертвования. Вы решаетесь положить конец сомнениям, и подобно искусному укротителю, внедряете важные элементы государственного акта переписи населения . . .

Рассмотрим упрощенную задачу в виде *ПРОЛОГ-программы*, содержащую все характерные элементы решения проблемы удовлетворения (сложной) цели на основе лишь фрагмента базы знаний (БЗ), содержащего факты и правила.

Факты — *клозы* (отдельные предикаты-высказывания принято называть *клозами*), которые не содержат правых частей, *правила* — *клозы*, которые содержат правые части; одноименные *факты* и *правила* объединяются в *процедуры*.

База знаний

Процедура "мужчина":

мужчина (иван)

мужчина (василий)

мужчина (петр)

мужчина (федор)

мужчина (юрий)

Процедура "женщина":

женщина (марья)

женщина (ирина)

женщина (ольга)

женщина (елена)

Процедура "родитель":

родитель (марья, иван) (Читать: "Марья — родитель Ивана")

родитель (иван, елена)

родитель (марья, василий)

родитель (федор, марья)

родитель (петр, ирина)

родитель (петр, иван)

родитель (федор, юрий)

Процедура "мать":

мать (X, Y): — женщина (X), родитель (X, Y)

Процедура "отец":

отец (X, Y): — мужчина (X), родитель (X, Y)

Процедура "брат":

брат (X, Y): — мужчина (X), родитель (P, X),
родитель (P, Y), $X \neq Y$

Процедура "сестра":

сестра (X, Y): — женщина (X), родитель (P, X),
родитель (P, Y), $X \neq Y$

Процедура "дядя":

дядя (X, Y): — брат (X, P), родитель (P, Y)

Пусть задана некоторая сложная (т.е. опирающаяся не на факт, а

требующая вывода) *цель*, с которой мы обратились в эту БЗ, например:

дядя (X, Y) (запись *цели* образует *фрейм*),

и ее решение (вывод) заключается в нахождении всех пар переменных (имен объектов) X и Y, для которых справедливо утверждение " X является дядей Y ".

Для решения такой задачи используется прием трансформации цели, который заключается в рекурсивном переборе различных вариантов подстановки вместо предикатов, составляющих сложную *цель*, *фактов* или правых частей *кловов* соответствующих *процедур*. Производится фиксация варианта связывания переменных и *унификация*, при которой отбрасываются несовместимые варианты связывания, противоречащие *фактам*. Варианты связывания всех переменных, прошедшие все этапы *унификации*, являются решением.

То есть для решения данной задачи необходимо действовать следующим образом.

Находим первый (а он и единственный) предикат *цели* дядя (X, Y) . Находим в БЗ *процедуру* с этим именем и заменяем найденный предикат правой частью этой *процедуры*. Получим трансформированную *цель* — *фрейм*

брат (X, P), родитель (P, Y).

К первому предикату этого *фрейма* применяем аналогичные действия, получаем *фрейм*

мужчина (X), родитель (Q, X), родитель (Q, P), X<>P, родитель (P, Y)

(во избежание коллизии, развивая *фрейм цели*, вводим новые переменные, отличные от тех, которые до того уже были использованы.)

Вновь входим в *процедуру* с именем первого предиката *цели*. Начинаем первый уровень ветвления. А именно, первый *клов процедуры* — *факт* мужчина (иван). С его помощью производим первое связывание

переменных, т.е. подстановку конкретного значения. Предикат *цели*, породивший *факт*, т.е. это связывание, может быть исключен из трансформируемой *цели*, т.е. заменен своей "пустой" правой частью:

родитель (Q, иван), родитель (Q, P), иван <>P,
родитель (P, Y).

Обращаемся к *процедуре* "родитель", начиная второй уровень ветвления. Первый *кюз* этой *процедуры* родитель (марья, иван) определяет дальнейшее связывание переменных:

родитель (марья, P), иван <> P, родитель (P, Y).

Вновь входим в *процедуру* "родитель" (третий уровень ветвления), находим *кюз* родитель (марья, иван) . Трансформируем *цель* — получаем новый *фрейм*:

иван <> иван, родитель (иван, Y).

Получаем противоречие, т.е. не проходит *унификация*.

Ищем на данном шаге ветвления другой вариант связывания, находим следующий *кюз*:

родитель (марья, василий).

Трансформируем *цель*:

иван <> василий, родитель (василий, Y) ->
родитель (василий, Y).

Вновь входим в *процедуру* "родитель", но не находим там *клоза*, в котором василий указан как чей-либо родитель. Т.е. вновь не проходит *унификация* — установление совместимости варианта связывания переменных.

Возвращаемся на шаг ветвления назад. (Реализуем стратегию поиска с ветвлением и возвращением назад — "*backtraking*".) На втором уровне ветвления пробуем *кюз*, в котором иван указан как сын: родитель (петр, иван) . *Цель* трансформируется в следующий *фрейм*:

родитель (петр, Р), иван <> Р, родитель (Р, Y).

Вновь (на третьем уровне ветвления) обращаемся к *процедуре* "родитель" и выбираем первый *кюз*, в котором петр указан как отец — родитель (петр, ирина).

Цель трансформируется:

иван <> ирина, родитель (ирина, Y) -> родитель (ирина, Y).

Входим в *процедуру* "родитель", но не находим там *клоза*, в котором ирина указана как родитель (не проходит *унификация*).

Возвращаемся на второй уровень ветвления и не находим там больше *клозов*, где иван указан как сын. Возвращаемся на первый уровень ветвления и в *процедуре* "мужчина" выбираем для последующего испытания следующий *кюз* мужчина (василий).

Цель принимает вид *фрейма*

родитель (Q, василий), родитель (Q, Р), василий <> Р, родитель (Р, Y).

Теперь на втором уровне ветвления находим первый (и единственный) *кюз*, в котором василий указан как сын. *Цель* трансформируется в соответствии с новым связыванием переменных, обусловленным найденным *кюзом* родитель (марья, василий):

родитель (марья, Р), василий <> Р, родитель (Р, Y).

На третьем уровне ветвления находим первый *кюз*, где марья — родитель: родитель (марья, иван). Связываем тем самым переменные, *цель* трансформируется

василий <> иван, родитель (иван, Y) -> родитель (иван, Y).

Находим в *процедуре* "родитель" первый *кюз*, в котором иван

указан как родитель — родитель (иван, елена) . Цель выродилась, значит

дядя (X, Y) = дядя (василий, елена) — одно из решений задачи.

Продолжив перебор так, словно на данном шаге *унификация* не прошла, можно найти остальные решения: дядя (юрий, иван), дядя (юрий, василий).

В основе распараллеливания решения этой задачи лежит способ размножения вариантов на основе трансформации цели. Способ обеспечивает отсутствие "*backtracking'a*" (ветвление есть, а возврата назад нет), простоту самой *процедуры* вывода, возможность неограниченного использования ИЛИ-параллелизма (одновременной независимой обработки многих вариантов связывания переменных), конвейерную реализацию И-параллелизма (распараллеливания обработки одного варианта связывания переменных на конвейере, т.к. каждый раз обрабатывается лишь первый предикат каждого *фрейма*).

Однако представляется, что нейросетевая технология, основанная на естественном параллелизме, может оказаться эффективной.

8.2. Нейросеть для решения задачи логического вывода

Рассмотрим предварительно один аспект обучения нейросети методом трассировки.

Первоначально (лекция 4), поясняя "схемотехнический" подход к построению нейросети, мы предполагали, что никаких связей ("проводочков") не было, а мы их проложили, исходя из реализуемых логических функций. Обобщив подход на нечеткую логику (предполагающую вхождение элементов в состав множества с некоторой вероятностью), а также логику передаточных функций, мы заявили, что нашли способ построения уже обученных нейросетей.

Впоследствии (лекции 5 и 6) мы заявили, что нас не смущает отсутствие некоторых связей в "готовых" сетях. Мы их введем как "проводочки", если сочтем это целесообразным.

Таким образом, несовершенство нейросети не стало для нас преградой. А если предположить, как при "схемотехническом" подходе, что первоначально в сети вообще никаких связей нет, ее матрица следования пуста, а нам предстоит создать эту сеть полностью? Логично, что в ней будут присутствовать только те связи, которые обеспечивают ее обучение. То есть мы сразу можем строить обученную нейросеть — нейросеть, *синаптические связи* в которой имеют единичный (максимальный) вес.

Составим (рис. 8.1) нейросеть, соответствующую фрагменту беспредельной базы знаний о жителях далекого таежного села.

Нейросеть будет отражать два уровня моделирования дедуктивного мышления, и состоять из двух частей.

Фактографическая нейросеть отражает факты, содержащиеся в БЗ. *Понятийная нейросеть* отражает правила вывода, оперируя понятиями – процедурами БЗ.

Моделируя лишь дедуктивное мышление, ограничимся возможностью дополнения *фактографической нейросети*, добываемой информацией на основе выводов, производимых с помощью *понятийной нейросети*. Развитие же *понятийной нейросети*, т.е. дополнение БЗ новыми правилами, требует решения проблемы индуктивного мышления, т.е. мышления более высокого уровня, пока не столь широко доступного.

Сформируем начальный вид *фактографической нейросети* (рис. 8.1).

Это – начальный вид, т.к. мы надеемся дополнить ее до состояния насыщения в результате фиксирования новых фактов, получаемых в длительном рабочем режиме эксплуатации, в процессе решения задач логического вывода. Такое развитие *фактографической нейросети* позволит исключить трудоемкий повторный вывод, воспользоваться однажды установленным фактом, словно записью на щите орудия о пораженных целях.

Для реализации правил построим *понятийную нейросеть*, фрагмент которой показан на рис. 8.2.

Рецепторный слой, несомненно, содержит элементы избыточности.

Однако различие в обозначениях повторяющихся переменных (и закрепленных за ними рецепторов) позволяет избежать коллизии при формировании выводов на основе формального описания *правил*.

Пусть задана та же сложная *цель*

дядя (X, Y).

Для нее необходимо сформировать все пары переменных X и Y, для которых справедливо утверждение "X является дядей Y".

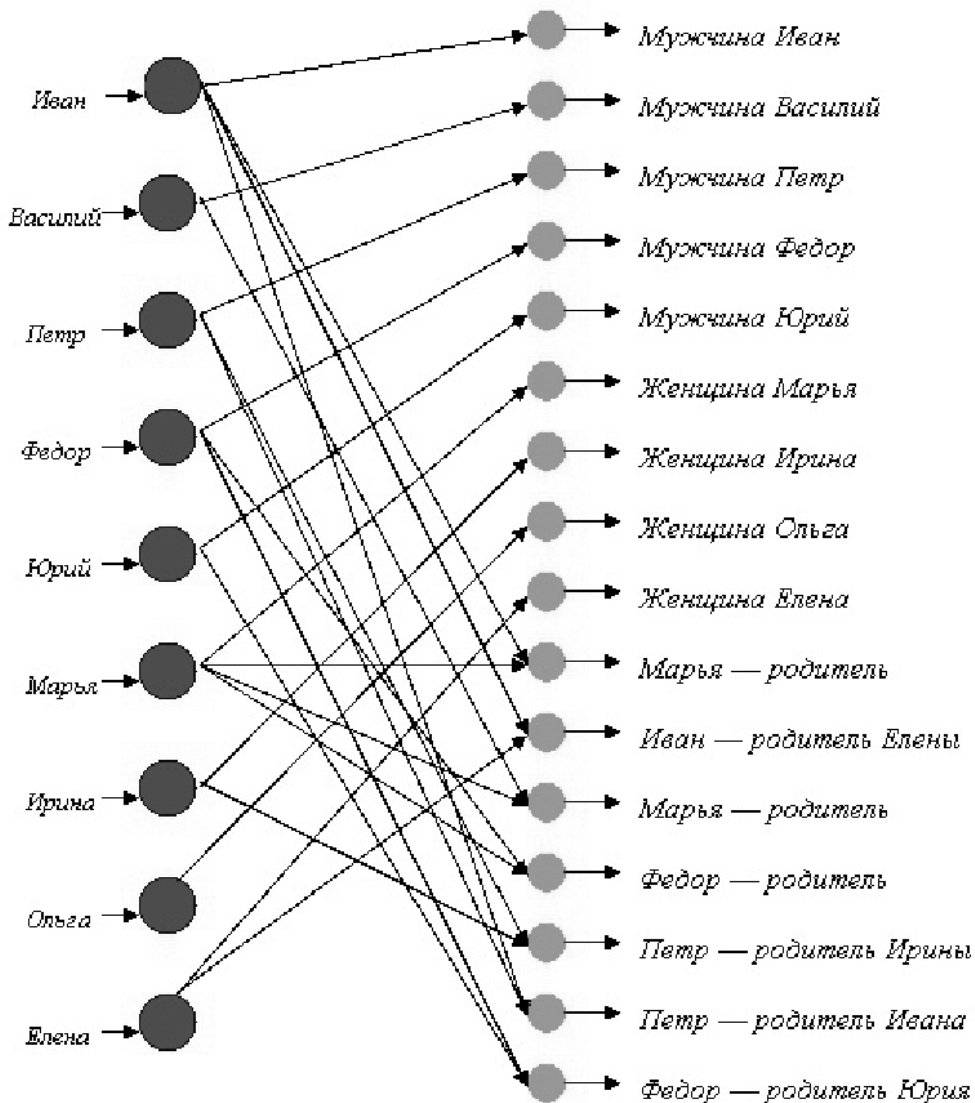


Рис. 8.1. Начальный вид фактографической нейросети

Выберем следующую передаточную функцию:

Порог h подбирается экспериментально, в данном случае, кажется, целесообразно положить $h = 0,4$.

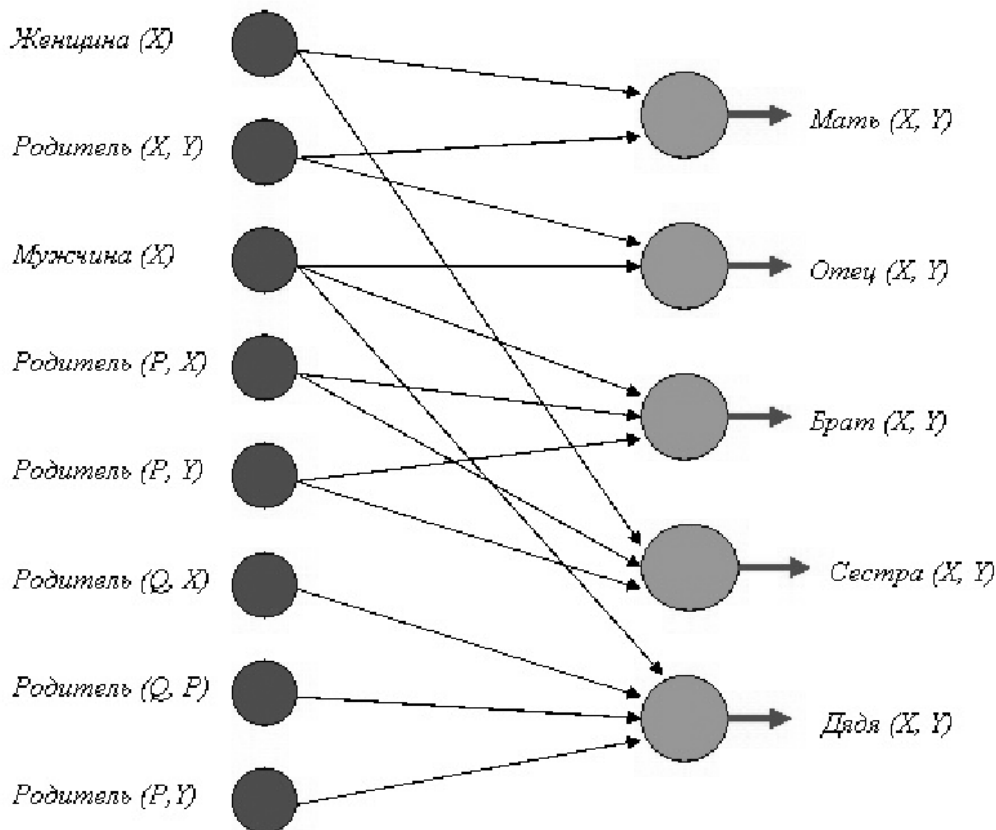


Рис. 8.2. Понятийная нейросеть

Чтобы максимально возбудить нейрон выходного слоя, ответственный за дядю, необходимо подать высокий сигнал возбуждения на рецепторы, передающие возбуждение этому нейрону, выполнив все варианты непротиворечивого связывания переменных X , Q , P , Y .

Таким образом, надо реализовать ту же схему перебора с возвратом, что была рассмотрена ранее, — схему *backtracking* 'а.

Проследим по шагам необходимые действия:

1. Полагаем $X = \text{иван.}$
2. В процедуре *Родитель* находим $Q = \text{мария.}$
3. В той же процедуре находим $P = \text{василий.}$
4. Однако, в той же процедуре не находим указаний на то, что

Василий является чьим-то родителем."

5. Возвращаемся на шаг назад, пытаюсь изменить связывание переменной P , но не находим нового варианта связывания."
6. Возвращаемся еще на шаг и т.д. – повторяем весь ход рассуждений, рассмотренных в подразделе 8.1."

После нахождения пары (василий, елена) следует развить фактографическую сеть, дополнив ее выходной слой нейроном в соответствии с понятием дядя и установив связи с ним от василия и елены.

Это может быть сделано двумя способами.

Первый способ повторяет весь ход логических рассуждений, связывая высказывания, приведшие к данному заключению. Тогда в результате решения всевозможных задач вывода установится окончательный вид *фактографической нейросети*, представленный на рис. 8.3. (Для упрощения рисунка некоторые входные связи недозволенно объединяются в "кабель".)

Как работать с этой сетью?

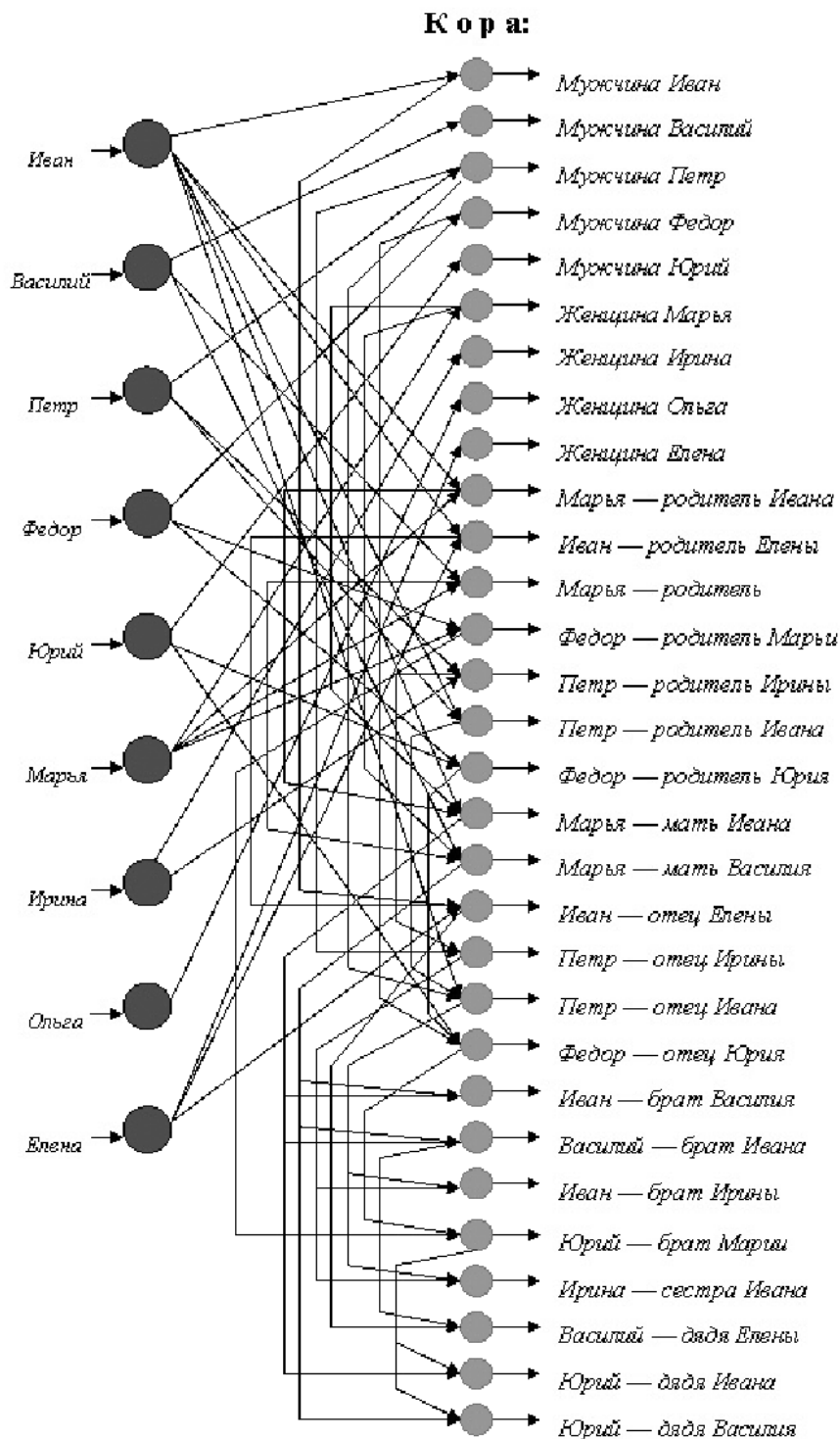


Рис. 8.3. База Знаний Антрополога–Исследователя

Можно себе представить два режима этой работы: для фискального органа — налоговой инспекции, требующей общей картины социального состояния, и для информационно-аналитической службы безопасности, интересующейся гражданами индивидуально, по выбору и привлечению.

А именно, если мы "засветим" (сообщив единичное возбуждение) все нейроны-рецепторы, то мы сможем прочесть все, что нейросеть "знает" о всех своих клиентах: кто из них мужчина, кто — женщина, в каком родстве они состоят.

Но мы можем спросить сеть лишь о некоторых персонажах. Например, мы можем возбудить на входе нейроны-рецепторы, соответствующие Ивану и Василию. Мы получим информацию о том, что они, несомненно, мужчины.

Но подозрительно высокого возбуждения достигнут нейроны, отвечающие за *факт*, что Марья — мать Ивана и она же — мать Василия.

Подозрительно возбудятся и нейроны, отвечающие за *факты* Иван — брат Василия и Василий — брат Ивана, Юрий — дядя Ивана и др. Тогда мы решаем: "А ну-ка, возбудим нейрон-рецептор, соответствующий Марье, и посмотрим, что интересного нам сеть сообщит!" И сеть возбуждением своих нейронов нам сообщает: "Да, действительно, Марья — мать Ивана и Василия".

Заметим, что мы можем подавать на входе и неединичные значения возбуждений. Например, установив, что в "деле" участвовал не то Иван, не то Василий (уж очень они похожи!), мы можем по принципу "фифти-фифти" задать соответствующие значения возбуждений равными 0,5 и установить высокую величину возбуждения нейрона, соответствующего дяде и того и другого — Юрию. И тогда логично возложить на него ответственность за нерадивое воспитание племянников.

Таким образом, составляя эту нейросеть, мы руководствовались желанием сохранить причинно-следственные логические связи,

обусловленные *правилами*. Это отражает ориентацию нейросети на пытливый ум, методом индуктивного мышления старающийся выяснить, какие *факты* достаточны для установления, например, того, что Юрий — дядя Ивана. Мы даже назвали эту нейросеть *Базой Знаний* Антрополога-Исследователя. Здесь же мы пошли на организацию коры. Мы справедливо решили, что все выводы данной сети должны быть равноправными и служить как формированию выводов "внутри себя", так и применению "вне" — для решения некоторой глобальной задачи, сверхзадачи. Равноправность и внешнее использование, в том числе для логической надстройки сети, для получения новых знаний по новым вводимым *фактам* и *правилам*, требует ревизии и особого установления величин возбуждения нейронов, что мы и отразили в передаточной функции.

Но мы ловим себя на лукавом мудрствовании . . . "Зачем так сложно?" — спросит рядовой гражданин, к каковому следует отнести жителя далекого села. — "Какое мне дело до намеков твоих? Ты давай напрямик, *факты* давай!.."

Действительно, зачем нам знать и делать вывод о том, что Юрий — дядя Ивана, потому что он — брат Марьи? Мы можем эти два *факта* задать независимо, предварительно рассчитав по *понятийной нейросети*.

Тогда фрагмент БЗ, которую мы назвали *Базой Знаний* Участкового Уполномоченного, представлен на рис. 8.4, и мы можем вздохнуть облегченно!

И не надо никакой коры! Строго, по-военному и как в анкете: посылка — следствие! Не надо сомнений, размышлений, ревизий и прочего вольнодумствия!

Таким образом, реализовав все причинно-следственные связи, мы построили нейросеть, отображающую только *факты*. Размножив ее для работы многих пользователей, можно получить достаточную информацию и исключить из рассмотрения *правила* вывода.

Рис. 8.4. База Знаний Участкового Уполномоченного

Рассмотренная схема работы с нейросетью отображает модель дедуктивного мышления. Но можно ли на основе полученной нейросети произвести обратные действия, построить обобщения, получить *правила*, достойные индуктивного мышления?

По-видимому, необходим анализ нейросети, сохранившей причинно-следственные связи, для выделения идентичных подструктур, объединяющих *факты* из общих *процедур* – носителей понятий.

Например, пути (*статические цепочки*), приводящие к выводу дядя (василий, елена), дядя (юрий, иван) , дядя (юрий, василий) в одинаковом порядке связывают "представителей" одних и тех же *процедур*. Тогда, выбрав вместо конкретных переменных их абстрактное обозначение, можно восстановить *правило* вывода. А это уже гипотеза — важный элемент индуктивного мышления. На основе этой гипотезы может быть развита *понятийная нейросеть* . Но гипотеза будет оставаться гипотезой, пока в достаточной мере не подтвердится практикой.

Тогда нейросеть обретает возможность широкого развития. Ведь если население села пополнилось родившимся младенцем, то правильность построенного *правила* (гипотезы) может быть успешно проверена!

. . . Да, мы вправе выбирать способ мышления как принцип жизни — для ее облегчения . . .

Структурное обоснование логической нейронной сети

На основе анализа простых примеров производится обоснование правильности применения основных положений математической логики событий при создании систем принятия решений. Показывается, что для повышения информативности нейронной сети заданной структуры обучение нейросети должно следовать структуре логического описания системы, однако требуется коррекция параметров нейронной сети и его логического описания.

"...она остановилась подле вплоть мимо ее проходящего поезда. Она смотрела на низ вагонов, на винты и цепи и на высокие чугунные колеса медленно катившегося первого вагона и глазомером старалась определить середину между передними и задними колесами и ту минуту, когда середина эта будет против нее.

Л.Н. Толстой. Анна Каренина

9.1. "Железнодорожная рулетка"

Как показывает опыт пропагандирования нейросетевых технологий, молодежь слабо воспринимает материал, заставляя лектора каждый раз начинать с пояснения "на пальцах", что собой представляет *нейросеть* на абстрактном уровне и как она работает. Опасаясь, что наши пространственные рассуждения выше отвратили нетерпеливого читателя, но надеясь, что он все же хочет выявить рациональное зерно, мы решили по случаю еще раз на примитивном уровне продемонстрировать подход, поймав читателя в ловушку и заставив его понять самый простой изначальный принцип.

Одновременно необходимо выявить ряд проблем, указывающих на то, что в предыдущих лекциях в области построения систем принятия решений сделано не все.

Построение обученной *нейросети*. Рассмотрим увлекательную детскую игру - "железнодорожную рулетку", - основанную на так хорошо знакомой вам задаче о встрече. Помните: "Из пунктов А и В навстречу друг другу . . . " и т.д.?

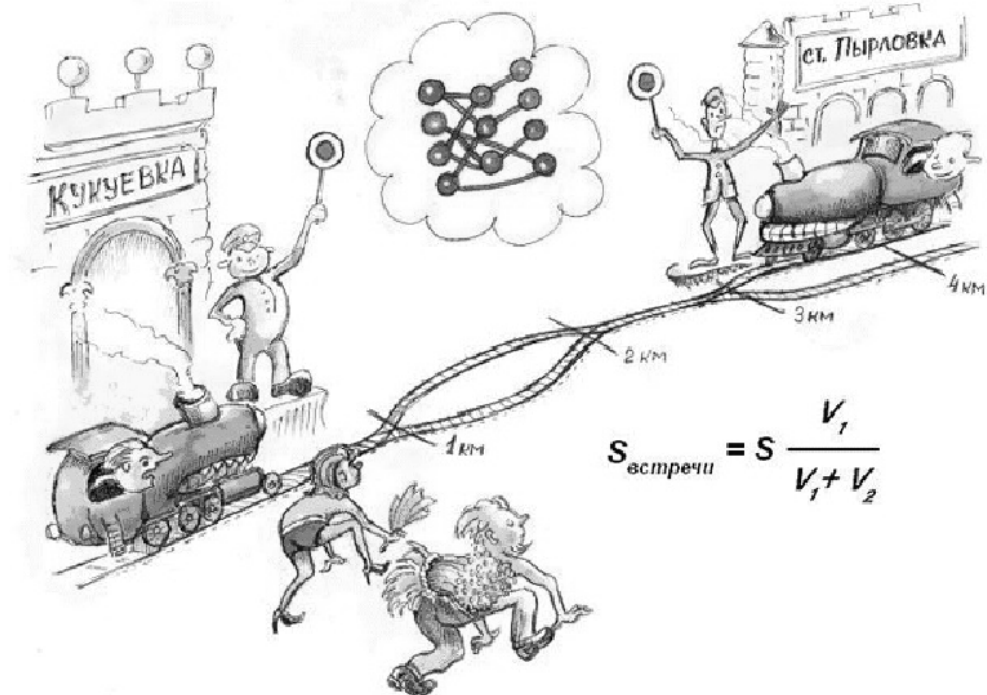


Рис. 9.1. Железнодорожная рулетка

Начальник станции Кукуевка (старший) и начальник станции Пырловка одновременно выпускают навстречу друг другу два паровоза (рис. 9.1) со скоростью либо 60, либо 80 км/ч. Длина перегона составляет 4 км. Небольшой нюанс заключается в том, что пути перегона то сходятся в один, на протяжении одного километра, то расходятся. И тогда, в зависимости от точки встречи, со станции Кукуевка надо выслать на соответствующий километр либо линейного - даму с приветственным платочком, либо линейного с подстилочной соломкой.

Решение о такой посылке усложняется помехами в линии передачи данных, в связи с чем скорости паровозов сообщаются с достоверностью, меньшей единицы. Кроме того, необходимо каждый эксперимент связать с ожидаемыми денежными затратами на одновременную добавку к пенсии линейных.

В отличие от вас, начальник станции Кукуевка неважно учился в пятом классе, и решать оперативно подобную задачу, да еще при

дополнительных условиях, не в состоянии. Он просит о помощи вас, предоставив в ваше распоряжение плохонький компьютер, отказанный в порядке шефской помощи Кукуевской начальной школой. Он объясняет вам, что не хочет считать вообще, а хочет добиться определенности по принципу "если . . . , то . . . ", а в случае недостоверных данных - "на что это более всего похоже и что делать?".

Тогда вы понимаете, что без элементов искусственного интеллекта не обойтись. Стимулируя свою изобретательность кружкой пива "Красный Восток", вы ищете что-то похожее на табличный метод, но с автоматической интерполяцией, что-то связанное с ассоциативным мышлением . . . И вы решаетесь . . .

Произведем предварительные расчеты, чтобы представить себе все варианты будущего поведения нашей системы принятия решений - для ее *обучения*. Представим (рис. 9.2) графически структуру логического функционирования создаваемой системы принятия решений.

1. Кукуевский паровоз имеет скорость 60 км/ч (Событие A_1). Пырловский паровоз имеет скорость 60 км/ч (Событие B_1). Одновременное выполнение этих событий обозначим $A_1 \& B_1$. Тогда точка встречи находится как раз посередине перегона, что, скорее всего, требует помощи линейного с соложкой. Но возможно и везение за счет неточного определения скоростей. Тогда на всякий случай потребуется дама с платочком. Принимаемое решение, заключающееся в отправлении обоих линейных на границу второго и третьего километров, назовем решением R_1 . С ним связаны расходы на единовременное пособие M_1 .

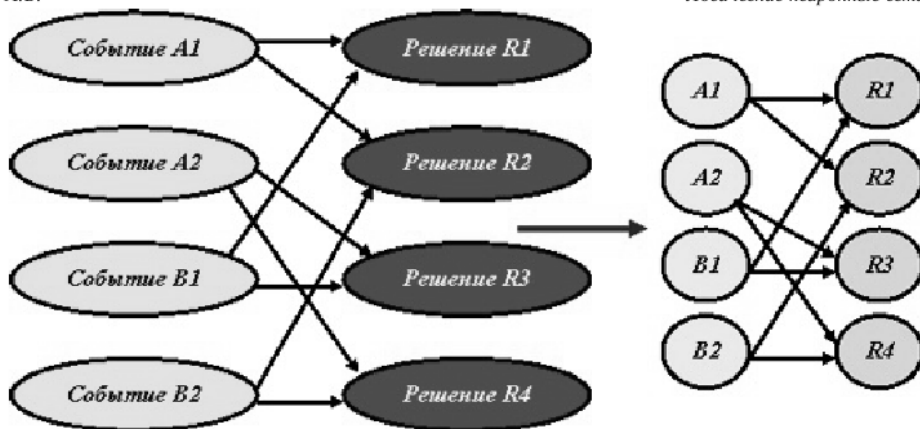


Рис. 9.2. Система принятия решений

2. Кукуевский паровоз имеет скорость 60 км/ч (Событие А1), но пырловский паровоз имеет скорость 80 км/ч (Событие В2). (Выполняется условие $A1 \& B2$.) Тогда их точка встречи находится на втором километре пути, и, следовательно, требует решения R2: "Отправить даму с платочком на второй километр!" В активе указанной дамы появляется сумма M2 условных единиц.
3. Кукуевский паровоз имеет скорость 80 км/ч (Событие А2), пырловский паровоз имеет скорость 60 км/ч (Событие В1). (Выполняется условие $A2 \& B1$.) Тогда их точка встречи находится на третьем километре пути, что требует сочувственного вмешательства линейного с соломкой (решение R3), с оплатой труда в M3 условных единиц.
4. Кукуевский и пырловский паровозы имеют скорость 80 км/ч (Событие $A2 \& B2$), что ввиду высокой скорости перемещения линейных требует решения R4 с затратами M4.

А теперь оживим эту структуру, заставим ее действовать, как, по-видимому, на логическом уровне действуют структуры нашего мозга.

Представим себе, что на месте каждого овала (потом - кружочка, на [рис. 9.2](#) справа) действует *нейроподобный элемент* (просто нейрон). Нейроны входного слоя - рецепторы - приходят в возбужденное состояние (подобно сетчатке глаза) в соответствии с той *ситуацией*, которую мы задаем на входе системы. Например, мы хотим испытать

ситуацию $A1 \& B2$. Тогда мы полагаем величины возбуждения нейронов $A1$ и $B2$ равными единице и записываем $V_{A1} = V_{B2} = 1$. При этом мы не забываем позаботиться о том, чтобы величины возбуждений нейронов $A2$ и $B1$ остались равными нулю.

Подчеркнем тот факт, что возбуждение нейронов-рецепторов осуществляется в результате ввода информации.

Для других нейронов, "принимающих" возбуждение в соответствии со стрелками, введем передаточную функцию, в результате выполнения которой формируется величина V возбуждения каждого нейрона. Для нашего случая, недолго думая (ибо существует большой произвол в выборе вида передаточной функции, на любой вкус), определим вид такой функции:

$$V = \xi \left(\sum_i V_i - h \right),$$

где i - индекс нейрона, "передающего" свое возбуждение данному нейрону, h - порог.

(Напомним: функция $\xi(x)$ заменяет отрицательную величину нулем, т.е.

$$\xi(x) = \begin{cases} x, & \text{если } x \geq 0, \\ 0, & \text{если } x < 0 \end{cases}$$

В нашем случае стрелки со всей определенностью указывают направление передачи возбуждений.

Положим $h = 1$ и рассчитаем величины возбуждения нейронов выходного слоя $R1-R4$ для ситуации $A1 \& B2$:

$$V_{R1} : 1 \div 0 - 1 = 0, \quad V_{R1} = 0;$$

$$V_{R2} : 1 \div 1 - 1 = 1, \quad V_{R2} = 1;$$

$$V_{R3} : 0 \div 0 - 1 = -1, \quad V_{R3} = 0;$$

$$V_{R4} : 0 \div 1 - 1 = 0, \quad V_{R4} = 0.$$

Таким образом, "высветилось" то решение, которое необходимо принять, и старт линейным должен быть дан. Проверим, что так же работает наша сеть по всем *эталонам*, по которым мы ее обучили, проложив "проводочки" от каждой исходной посылки к следствию.

Теперь поэкспериментируем.

Задавая события по принципу "да - нет", "1 - 0", мы предполагали *булевский тип* исходных данных. А что, если поменять, обобщить тип исходных данных, допустив рассмотрение значений, интерпретируемых как достоверность? Или даже каких-то других взаимных оценок, которые используются часто в быту людьми, не сведущими в теории вероятности и не знакомых с понятием "полное множество событий"?

Например, в результате искажения информации начальник станции Кукуевка принял решение считать скорость пырловского паровоза равной не то 60, не то 80 км/ч. Но скорее всего - 60! И подойдя к компьютеру, он по наитию набирает: $A_1 = 1$, $B_1 = 0,7$, $B_2 = 0,4$. На какую *ситуацию* это указывает, и какое решение наиболее правильно? Считаем:

$$V_{R1} : 1 \div 0,7 - 1 = 0,7, \quad V_{R1} = 0,7;$$

$$V_{R2} : 1 \div 0,4 - 1 = 0,4, \quad V_{R2} = 0,4;$$

$$V_{R3} : 0 \div 0,7 - 1 = -0,3, \quad V_{R3} = 0;$$

$$V_{R4} : 0 \div 0,4 - 1 = -0,6, \quad V_{R4} = 0.$$

Мы видим, что максимальной величины возбуждения достиг нейрон R1, определивший главное решение. Но мы вправе учесть и решение R2 с меньшим приоритетом, дав даме с платочком дополнительные указания. И в этом проявится наша мудрость. По известной формуле мы можем оценить математическое ожидание того, на сколько облегчится карман начальника Кукуевской станции:

$$M = \frac{M1 \cdot 0,7 + M2 \cdot 0,4 + M3 \cdot 0 + M4 \cdot 0}{0,7 + 0,4 + 0 + 0}$$

Мы очень просто сформировали уже обученную *нейросеть*. Однако критический взгляд читателя замечает явные "проколы" и выражает недоумение. Ведь ранее мы говорили о большем! Что ж, подойдем к

этому постепенно . . .

А что, если бы мы захотели объединить решения R1 и R4, отличающиеся (для нас) только скоростью передвижения линейных? Следуя тому же принципу формирования, мы получили бы сеть как на рис. 9.3.

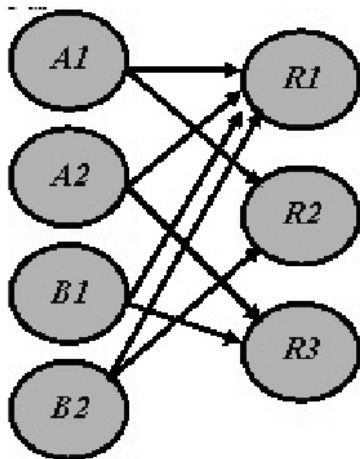


Рис. 9.3. Непригодность однослойной нейросети

Легко видеть, что решение R1 максимально возбуждается всегда, когда мы задаем *ситуацию*, требующую той же величины максимального возбуждения только лишь нейронов R2 и R3. Сеть как бы дает "побочный эффект". Необходим дополнительный инструктаж пользователя. Он заключается в том, что если максимально и одинаково возбуждены два нейрона выходного слоя и один из них R1, то "верить" надо второму. Если максимально возбуждился только нейрон R1, то он и выдает правильное решение.

В данном случае произошла коллизия при огульной замене *операций конъюнкции* \wedge и *дизъюнкции* \vee одной операцией $\&$. Ведь логический предикат возбуждения решения R1 выглядит как

$$(A1 \wedge B1) \vee (A2 \wedge B2) \rightarrow R1$$

Именно это и наводит на предположение (гипотезу) о минимальной длине *статической цепочки*, которая рассматривается в разделе 1.9. По-видимому, *нейросеть* будет работать правильно, если ее структура

полностью воспроизведет структуру указанного логического выражения. Но для этого *нейросеть* должна быть двухслойной! Более того, глубокий анализ показывает, что обобщенный *эталон* $A1, A2, B1, B2$, приводящий к одному решению ($R1$), поглотил обобщенные *эталоны* (например, $A1, B2$), приводящие к другому решению. О какой же однозначности выводов можно говорить!

Тогда легко восстановить справедливость, построив (обученную!) *нейросеть* так, как показано на рис. 9.4, введя в рассмотрение т.н. "скрытые" нейроны 1 и 2.

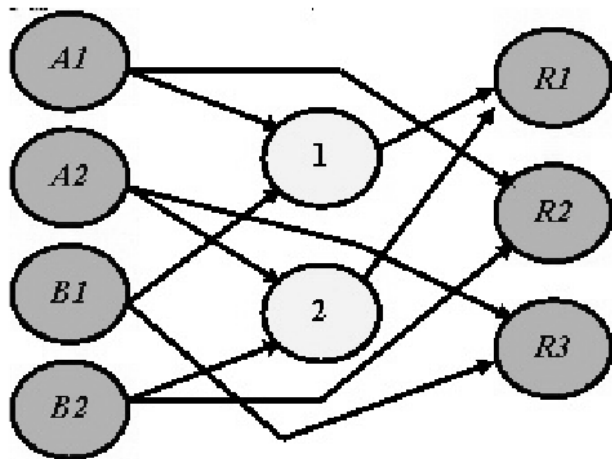


Рис. 9.4. "Правильная" обученная нейросеть

Выбор *нейросети*, обучение - трассировка. Мы построили *нейросеть*, пользуясь приемами, известными специалистам-схемотехникам, конструирующим устройства компьютера. Мы соединили элементы связями-"проводочками", произведя трассировку, для правильного распространения сигнала в соответствии с замысленной функцией. Мы неоднократно указывали выше, что так мы строим уже обученную сеть. При этом мы полностью исключили из рассмотрения тот параметр, настройка которого позволяет обучить сеть, в частности - проложить нужные "проводочки" и перекусить ненужные.

Нейрон только умеет выполнять передаточную функцию, один из видов которой мы рассмотрели. Но более *полная модель* нейрона заключается в следующем: нейрон имеет несколько входов - дендритов, каждый из которых обладает весом *синаптической связи*. В результате выполнения

передаточной функции возбуждение нейрона через ветвящийся аксон передается дендритам других нейронов, с которыми связан этот. Дендрит воспринимает сигнал, умноженный на свой вес! Таким образом нейроны и образуют сеть, в которой различаются входной и выходной слои. Передаточная же функция, с учетом синаптических весов, для нашего простейшего случая (при компьютерном моделировании чаще всего другого и не требуется) имеет вид

$$V = \xi \left(\sum_i \omega_i V_i - h \right),$$

где V_i - величины возбуждения тех нейронов, аксоны которых связаны с дендритами данного нейрона, i - индекс использованного дендрита, ω_i - вес синаптической связи.

И вот весь фокус в построении и в обучении нейросети заключается в том, что синаптические веса регулируются, обуславливая пути прохождения возбуждений в сети!

В частности, представив некоторую абстрактную сеть, мы, построив на ее основе сеть для игры в рулетку, положили некоторые веса связей равными единице (утвердив существование "проводочков"), а некоторые (или все другие) - равными нулю (что соответствует отсутствию "проводочков"). Но ведь можно допустить и некоторые промежуточные значения весов, хотя в практических целях можно поступать так, как поступили мы.

Подойдем иначе к построению нейросети для игры в "железнодорожную рулетку". Ранее нам были известны условия игры, а мы создали сеть. Теперь пусть нам задана нейросеть, а мы обучим ее для игры в рулетку.

Итак, по сошедшему вдохновению мы нарисовали некоторый ориентированный ациклический граф (рис. 9.5) и намерены вложить в него смысл нейросети, поставив в соответствие его вершинам-нейронам (кроме предполагаемых рецепторов) все ту же передаточную функцию.

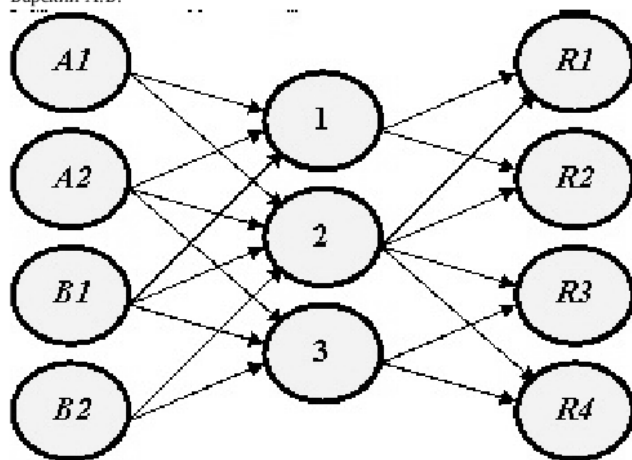


Рис. 9.5. Нейросеть, предложенная для обучения

Вот только каким способом заставить сеть так реагировать на очевидные *эталоны*, чтобы максимального возбуждения достигли нейроны выходного слоя, соответствующие решениям? Для этого необходимо, полагая первоначально все веса нулевыми (или минимальными), увеличить некоторые веса, довести до максимального значения или до единицы. Проще всего именно так и действовать: сначала все веса нулевые (хоть "проводочки" есть, их сопротивление чрезвычайно высоко). Затем некоторые веса (и наша задача выбрать, какие) мы полагаем равными единице. Это и будет равносильно тому, что какие-то "проводочки" мы оставим, а какие-то перекусим. Это действие по *обучению нейросети* мы называем трассировкой.

Продемонстрируем алгоритм трассировки, введя, по сравнению с рассмотренным в [лекции 4](#), некоторые упрощения.

Компьютерная обработка *нейросети* значительно упрощается, если сеть представлена *матрицей следования* S ([рис. 9.6](#)), где наличие связи обозначается ее весом.

A1	■											
A2		■										
B1			■									
B2				■								
1	0	0	0		■							
2	0	0	0	0		■						
3		0	0	0			■					
R1					0	0		■				
R2					0	0			■			
R3						0	0			■		
R4						0	0					■

Рис. 9.6. Матрица следования

1. Произведем трассировку возбуждений нейронов {A1, B1} -> R1.

Исключим из матрицы S строки и столбцы, соответствующие не интересующим нас нейронам входного и выходного слоев. Матрица примет вид S_1 на рис. 9.7.

A1	■									Возб.
B1		■								Возб.
1	0	0	■							
2	0	0		■						
3		0			■					
R1			0	0			■			

Рис. 9.7. Матрица следования для трассировки первого решения

Моделируем прохождение возбуждения следующим образом.

Присвоим нейронам, соответствующим нулевым строкам - входам

матрицы S_1 признак "возбужден". Выделим столбцы, соответствующие этим входам. В совокупности этих столбцов найдем первую строку, содержащую максимальное число нулей. Эта строка соответствует нейрону 1. Заменяем нули единицами (увеличиваем веса), введенные изменения отражаем в матрице S . К матрице S присоединяем столбец (чтобы не отягощать пример, мы этого не сделали, но учитываем его наличие в последующих построениях), в каждой позиции которого указывается число введенных единиц в строке. В данном случае в строке этого столбца, соответствующей нейрону 1, записываем $\tau_1 = 2$. Это необходимо для возможности "переиспользования" нейронов при получении других решений. Исключаем из матрицы S_1 строки и столбцы, "передавшие" свое возбуждение. Нейрону 1 присваиваем признак "возбужден". Матрица S_1 принимает вид на рис. 9.8.

1					Возб.
2					
3					
RI	0	0			

Рис. 9.8. Шаг преобразования матрицы следования

Исключаем из матрицы S_1 строки и столбцы, соответствующие входам, не отмеченным признаком "возбужден". Эти строки соответствуют нейронам 2 и 3. Матрица S_1 принимает вид как на рис. 9.9.

1			Возб.
RI	0		

Рис. 9.9. Шаг преобразования матрицы следования

Повторяем очевидные действия, уже описанные выше, что приводит к замене единицей единственного нуля.

Не приводя промежуточного рисунка, отметим, что мы подтвердили высоким весом (единичным) связи $A1 \rightarrow 1$, $B1 \rightarrow 1$, $1 \rightarrow R1$.

2. Произведем трассировку $\{A1, B2\} \rightarrow R2$.

Сформируем матрицу S_2 , исключив из рассмотрения нейроны $A2, B1, R1, R3, R4$ (рис. 9.10).

A1	■						
B2		■					
1	1		■				$m_1=2$
2	0	0		■			
3		0			■		
R2			0	0		■	
						■	

Рис. 9.10. Матрица следования для трассировки второго решения

Строка, соответствующая нейрону 1, содержит одну единицу при том, что $\tau_1 = 2$. Исключаем из рассмотрения и этот нейрон, как не годный к переиспользованию. Матрица S_2 принимает вид как на рис. 9.11.

A1	■						Возб.
B2		■					Возб.
2	0	0	■				
3		0		■			
R2			0		■		

Рис. 9.11. Шаг преобразования матрицы следования

(Для краткости изложения мы не рассматриваем транзитивные связи, легко вводящиеся в матрицу следования. Тогда мы могли бы исключить из рассмотрения нейрон 3, так как нет связи $3 \rightarrow R2$.)

Присваиваем строкам, соответствующим входам матрицы S_2 , признак "возбужден". Находим в совокупности соответствующих им столбцов строку, содержащую максимальное число нулей. Это строка, соответствующая нейрону 2. Заменяем в ней нули единицами, что отмечаем в матрице S . Полагаем $\tau_2 = 2$. Присваиваем нейрону 2 признак "возбужден", а нейроны A1 и B2 исключаем из рассмотрения. Среди оставшихся строк оказывается "пустая" строка, которая соответствует нейрону 3, не обладающему признаком "возбужден". Исключаем и ее, вместе с соответствующим столбцом. Матрица S_2 принимает вид как на [рис. 9.12](#).

2	■	□	Возб.
R2	0	■	

Рис. 9.12. Шаг преобразования матрицы следования

Повторение очевидных действий приводит к замене оставшегося нуля единицей.

Таким образом, в результате трассировки на данном шаге сложились связи с единичными весами $A1 \rightarrow 2$, $B2 \rightarrow 2$, $2 \rightarrow R2$.

3. Повторив схему построений, легко найдем связи с единичными весами $A2 \rightarrow 3$, $B1 \rightarrow 3$, $3 \rightarrow R3$, в результате трассировки $\{A2, B1\} \rightarrow R3$.
4. А вот трассировка последнего пути возбуждения, $\{A2, B2\} \rightarrow R4$, преподносит сюрприз, следующий из "слоистости" сети. (Сравните с сетью на [рис. 9.4](#), где в этом смысле допустимы связи через слой.)

Сформируем матрицу S_4 , свободную от представительства тех нейронов, в строках которых число единиц меньше соответствующего значения τ . Такая матрица показана на [рис. 9.13](#). Но ведь никаких связей в ней нет вообще!

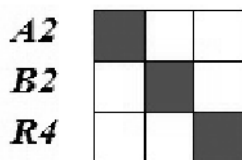


Рис. 9.13. Матрица следования для трассировки четвертого решения

Придется их ввести, да еще с единичными весами. Ведь мы готовы создавать нужную нам сеть, а не обязаны приспосабливаться. Да и что мы можем сделать - только отвергнуть эту сеть и искать новую? Мы видим, что больше трех *эталонов* эта сеть все равно не способна воспринять. Таким образом, мы вводим дополнительные связи с единичными весами $A2 \rightarrow R4$, $B2 \rightarrow R4$.

Окончательно обученная сеть представлена на [рис. 9.14](#), где выделены связи с единичными весами. (Другие "проводочки" мы могли бы перекусить.)

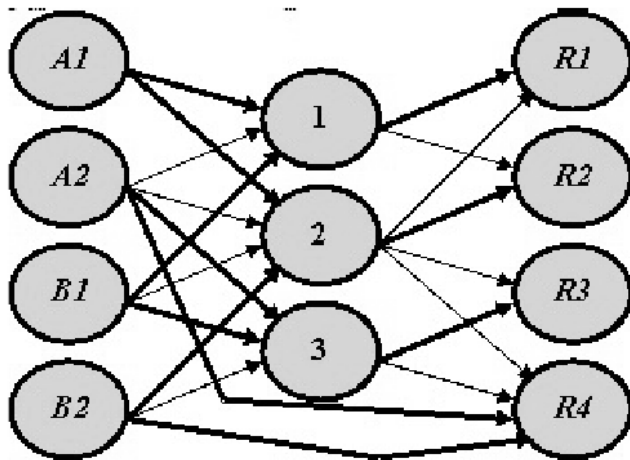


Рис. 9.14. Обученная нейросеть

Но радоваться рано. Посчитаем с помощью нашей передаточной функции величины возбуждения нейронов для, например, *ситуации* $A1 \& B2$: $V_1 = 0$, $V_2 = 1$, $V_3 = 0$, $V_{R1} = 0$, $V_{R2} = 0$, $V_{R3} = 0$, $V_{R4} = 0$. Ни один нейрон выходного слоя не возбудился!

То же - для *ситуаций* A1&B1 и A2&B1. Рассчитаем *ситуацию* A2&B2: $V_1 = 0, V_2 = 0, V_3 = 0, V_{R1} = 0, V_{R2} = 0, V_{R3} = 0, V_{R4} = 1$. Мы видим, что построенная *нейросеть* распознает единственную *ситуацию*.

Анализируя, мы убеждаемся, что в процессе распространения по сети возбуждение "гаснет", не доходя до выходного слоя. Тогда мы начинаем "подкручивать" пороги, уменьшая их. Но тотчас замечаем другую неприятность: величины возбуждения нейронов выходного слоя различны для разных *ситуаций*, ибо различна длина путей возбуждения. Это затрудняет правильное участие этих нейронов в следующих логических слоях, когда данный выходной слой является входным для следующего логического уровня сети. Мы видим, что наша передаточная функция не годится для выбранной структуры *нейросети*.

Но мы же располагаем свободой выбора, которая допускает условности, вероятно, нереализованные в природе, находящейся в жестких рамках установленных законов и средств.

Рекомендуем "хорошую" передаточную функцию, определяющую величину V возбуждения нейрона:

$$V = \begin{cases} 0, & \text{если } \sum_i \omega_i V_i \leq h, \\ \sum_i \omega_i V_i, & \text{в противном случае} \end{cases}$$

Рассчитайте и убедитесь, что, например, для $h = 1$ сеть правильно распознает все *эталонные ситуации*, обеспечивая равную величину возбуждения нейронов выходного слоя. Так, при *ситуации* A1&B1 получаем следующие величины возбуждения нейронов: $V_1 = 2, V_2 = V_3 = 0, V_{R1} = 2, V_{R2} = V_{R3} = V_{R4} = 0$. Аналогично - для *ситуаций* A1&B2 и A2&B1. Для *ситуации* A2&B2 находим $V_1 = V_2 = V_3 = 0, V_{R1} = V_{R2} = V_{R3} = 0, V_{R4} = 2$.

Сеть работает прекрасно, гарантируя правильность реакции на недостоверные *ситуации*, и позволяя находить среднее.

А главное, сколько прекрасных вариантов развития имеет игра, стимулируя наше предвидение! Например, что, если скорость паровозов

- величина переменная и случайная, так же как и чередование и длина однопутных участков, и решение следует принимать и корректировать в динамике, в зависимости от длины пройденного пути и значения скорости? Что, если один машинист охвачен идеей суицида, а другой желает уклониться от столкновения? и т.д. (Бедные линейные!)

9.2. Практический подход и обоснование структуры логической нейронной сети для системы принятия решений

В лекции 1 проводилось обоснование системы принятия решений (СПР) на основе основных положений математической логики событий. В последующих лекциях рассматривались примеры практического построения СПР. Однако в предыдущем разделе данной лекции возникли проблемы, связанные с тем, что при заданной структуре и количестве нейронов в обучаемой *нейронной сети* не всегда удастся предусмотреть однозначность выводов. Указывается на важность следования альтернативным правилам: либо "размножением" решений сводить сеть к однослойной, либо при ее трассировке строго соблюдать скобочную структуру логических функций, описывающих СПР.

Необходимость популяризации логических *нейронных сетей* требует вновь, на более высоком уровне, вернуться к рассмотрению и анализу более обобщающих примеров для выработки практических рекомендаций по конструированию логических *нейронных сетей* и их трассировке. Тем более это необходимо в том случае, если нетерпеливый читатель отвергнет лекцию 1, превращающую, на его взгляд, проблему в "темную и запутанную".

Пусть гипотетическая СПР контролера электропоезда основана на следующих высказываниях:

X_1 = "пассажир предъявил билет";

X_2 = "пассажир не предъявил билет";

X_3 = "в билете указана дата (число) этого дня";

X_4 = "в билете указана дата (число) не этого дня";

- X_5 = "в билете указан текущий месяц";
 X_6 = "в билете указан не текущий месяц";
 X_7 = "в билете указан текущий год";
 X_8 = "в билете указан прошлый год";
 X_9 = "в билете указан более ранний год";
 X_{10} = "предъявлены проездные документы работника МПС";
 X_{11} = "предъявлено пенсионное удостоверение";
 X_{12} = "не предъявлено пенсионное удостоверение";
 X_{13} = "предъявлено удостоверение работника МПС";
 X_{14} = "не предъявлено удостоверение работника МПС";
 X_{15} = "предложена взятка".

Принимаемые решения:

- R_1 = "поблагодарить и извиниться за беспокойство";
 R_2 = "взыскать штраф 100 рублей";
 R_3 = "взыскать штраф 300 рублей";
 R_4 = "вызвать милицию";
 R_5 = "пожурить".

Зададим логическое описание СПР:

$$\begin{aligned}
 &(X_1 \wedge X_3 \wedge X_5 \wedge X_7) \vee X_{10} \rightarrow R_1, \\
 &(X_1 \wedge X_4 \wedge X_7) \vee (X_1 \wedge X_6 \wedge X_7) \vee (X_2 \wedge X_{12} \wedge X_{14}) \rightarrow R_2, \\
 &(X_1 \wedge X_8) \vee (X_{10} \wedge X_8) \rightarrow R_3, \\
 &(X_1 \wedge X_9) \vee (X_{10} \wedge X_9) \vee (X_2 \wedge X_{15}) \rightarrow R_4, \\
 &(X_2 \wedge X_{11}) \vee (X_2 \wedge X_{13}) \rightarrow R_5.
 \end{aligned} \tag{9.}$$

Инструкция контролеру на этапе ее согласования может учитывать не все *ситуации*, с которыми он встречается. Например, *ситуация* $X_1 \wedge X_{10}$ требует особой благодарности контролера от лица МПС!

Именно возможная неполнота данных обуславливает применение ассоциативного мышления, моделируемого нейросетью.

Не ограничивая общности рассуждений, будем считать, что каждая

логическая функция F_i , $i = 1, \dots, S$, представляет собой *дизъюнктивную нормальную форму* (ДНФ) в смысле [22], т.е. имеет вид дизъюнкции конъюнкций, где количество членов, составляющих различные конъюнкции, может быть различным.

Отметим, что традиционно в схемотехнике за ДНФ принимают так называемую совершенную ДНФ (СДНФ), формируемую по известному [22] правилу построения на основе таблицы значений. В ней все конъюнкции имеют одинаковую длину, и их составляют одни переменные в совокупности с отрицанием других.

Однако заведомо излишне учитывать в каждой *ситуации*, например, предложена взятка или нет. Таким образом, ДНФ, как исходная форма представления, вполне достаточна для полного описания СПР.

Более того, при корректном описании СПР не используется *операция отрицания*, так как применяются лишь исчерпывающие множества событий (здесь читателю все же придется обратиться к лекции 1). Действительно, отрицание некоторого события равно дизъюнкции остальных событий того же множества. Так, в нашем примере для исчерпывающего множества событий X_1 и X_2 справедливо равенство $X_2 = X_1$.

Граф-схема выполнения *системы логических выражений* (9.2), т.е. логическая схема (И-ИЛИ сеть по терминологии [25]) представлена на рис. 9.15. (Не следует пока обращать внимание на веса и пороги.)

Пусть вершины $1-11$ этого графа соответствуют логическим элементам - *конъюнкторам*, а вершины R_1-R_5 - *дизъюнкторам*. Тогда построенная схема отображает функционально законченное устройство, реализующее таблицу, с помощью которой контролер осуществляет свои действия. А именно, задавая на входе значения булевых переменных, характеризующие *ситуацию*, он на одном из выходов хочет получить булево значение "1", указывающее на принимаемое решение. (Далее мы обнаружим ошибку.)

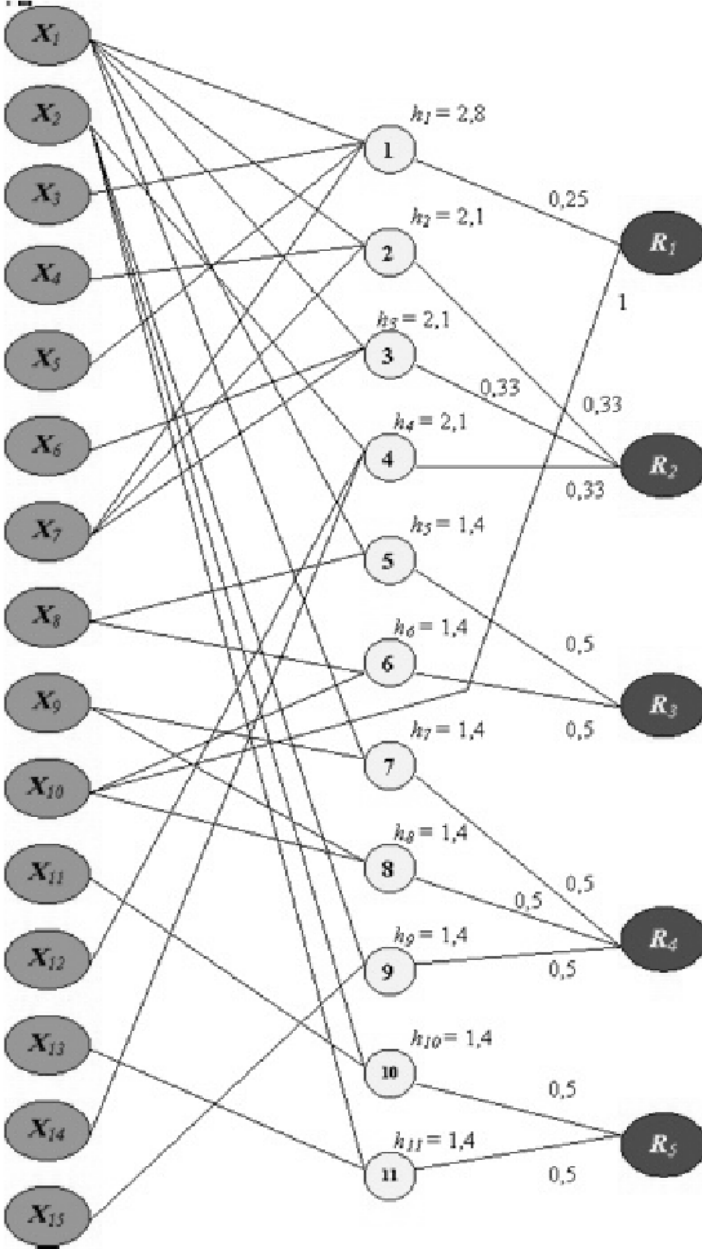


Рис. 9.15. Логическая схема СПР - структура обученной нейросети

Однако предлагаемый табличный метод обусловлен не только тем, что данная задача относится к типу *трудно формализуемых задач*, т.е. задач, для которых нетипично строгое математическое описание, влекущее

построение конструктивных алгоритмов вычисления. Главным образом СПР характеризуется не столько отсутствием математических зависимостей между ее составляющими, сколько недостоверностью данных, противоречивостью информации, работой в условиях помех и т.д. В этом случае СПР реализует модель ассоциативного мышления, которая по неполной, недостоверной, "зашумленной" информации должна выдать ответ на вопрос "На что более похожа *ситуация* и какое решение наиболее правильно?"

Таким образом, рассмотренная реализация табличного метода, предусматривающего точное задание данных о складывающейся *ситуации* по принципу "да - нет", должна распространяться на случай неполных, недостоверных данных.

Это означает, что система, *отображенная графом* на рис. 9.15, должна работать не с булевыми переменными на входе, а с действительными, смысл которых основан на достоверности, вероятности принадлежности (интервалу, значению и др.), экспертной оценке и т.д. Таким образом, должна использоваться не точная информация о *ситуации* на входе создаваемой СПР, а лишь оценки этой информации. Это было отмечено в предыдущих лекциях.

Отметим, что в "грамотно" построенной системе такие оценки могут быть вероятностными, являющимися оценками достоверности. Однако принципиально допустима и недостаточная грамотность пользователя. Например, он может быть не осведомлен о понятии *исчерпывающего множества событий*. Главное, чтобы оценки были относительными, отображающими принцип "больше - меньше" (аналогично экспертным системам), т.к. этого достаточно для моделирования ассоциативного мышления.

Однако *конъюнкторы* и *дизъюнкторы* определены лишь для булевых переменных. Следовательно, они должны быть заменены некоторым универсальным элементом, реализующим суррогат этих операций - передаточной функцией, способной на логическом уровне осуществлять схожую реакцию на сигналы на ее входе для получения оценочного сигнала на выходе. Это и привело к моделированию нейрона - основного логического элемента мозга, к воспроизведению искусственного интеллекта, одним из основных принципов которого

является ассоциативное мышление.

На основе логической схемы (рис. 9.15) построим *нейронную сеть* той же структуры, обученную для решения нашей задачи. Вершины $X_1 - X_{15}$ соответствуют нейронам-рецепторам входного слоя. От булевых значений их возбуждения перейдем к действительным - к оценкам достоверности соответствующих высказываний ("грамотный" вариант). Эти значения задаются пользователем скорее по наитию, "на глазок", на основе опыта.

Каждая конъюнкция высказываний в записи логических функций (9.1), т.е. совокупность событий, определяет *ситуацию*. *Ситуация* является *эталоном* (*эталонной ситуацией*), если все составляющие ее события обладают достоверностью, равной единице. В логической схеме каждой конъюнкции соответствует вершина из множества $\{1, \dots, 11\}$. В *нейронной сети* эти вершины обозначают нейроны промежуточного или скрытого слоя. Нейроны $R_1 - R_5$ образуют выходной слой; их возбуждение указывает на принимаемое решение.

Как говорилось выше, нейроны *рецепторного слоя* возбуждаются пользователем, задающим предполагаемую вероятность (или другую оценку) соответствующего события. Остальные нейроны реализуют передаточную функцию таким образом, чтобы возбуждение нейронов-рецепторов распространялось по сети в соответствии со связями. А именно, если, например, на входе сформирован высокий уровень возбуждения нейронов X_1, X_4, X_7 по сравнению с возбуждением других нейронов-рецепторов, то большая величина возбуждения нейрона 2 должна обеспечить самое высокое возбуждение нейрона R_2 среди всех нейронов $R_1 - R_5$ выходного слоя.

Отметим, что таким образом мы пытаемся построить уже обученную *нейросеть*, где по всем *эталонным ситуациям* максимального возбуждения должны достигать те нейроны выходного слоя, которые ответственны за решения, соответствующие этим *ситуациям*. Таким образом, реализуется таблица, о которой говорилось выше. Если же с помощью достоверности событий задавать на входе *ситуации*, явно не существующие, то нельзя гарантировать правильный ответ. Например, если достоверность всех событий $X_1 - X_7$ положить равной единице,

то столь же бессмысленно будет распределение возбуждения нейронов выходного слоя. Или, если предположить, что мятая бумажка является предъявленным билетом с достоверностью 0,1 (событие X_1), то полагать высоким значение достоверности события X_8 не следует, т.к. эта достоверность является условной вероятностью, и т.д.

То есть логика мышления пользователя и знание элементов теории вероятности должны возобладать.

9.3. Выбор передаточной функции

Выбор передаточной функции остается творческой проблемой, во многом определяемой решаемой задачей. Важно лишь то, что полное копирование нейрона, созданного природой, излишне. В то же время передаточная функция - пороговая функция, активно использующая значение порога h .

В нейронных сетях учитывают веса связей - синаптические веса ω . Рассмотрение этих весов актуально, если структура сети первоначально задана и следует ее приспособить (обучить) для решения данной задачи. Пока мы рассматриваем построение уже обученной *нейросети*, поэтому вес сформированных связей принимаем равным единице. Однако далее будет показано, что *корректировка весов* необходима даже при построении обученных сетей. Эти веса могут быть скорректированы и в процессе эксплуатации для учета влияния событий на результат - принимаемое решение.

При выборе передаточной функции и порога h руководствуются следующими требованиями:

- эти функции в области преодоления порога должны монотонно возрастать по каждому сигналу на входе нейрона;
- не должно быть "угасания" сигнала возбуждения при его прохождении по сети;
- сигналы возбуждения на выходном слое должны быть четко различимы по величине для различных *эталонных ситуаций*;
- должен быть примерно равным диапазон изменения величин возбуждения нейронов выходного слоя, закрепленных за разными

решениями.

Практически, для логического решения задач, достаточно применять одну из следующих, например, используемых в [7], передаточных функций, определяющих величину V возбуждения нейрона в зависимости от величин V_i возбуждения связанных с ним нейронов, весов ω_i этих связей, а также порога h :

$$V = \xi(\sum_i \omega_i V_i - h),$$

$$1. \left(\xi(x) = \begin{cases} x, & \text{если } x \geq 0, \\ 0, & \text{в противном случае} \end{cases} \right)$$

$$2. \begin{aligned} V &:= \xi(\sum_i \omega_i V_i - h), \\ V &:= \text{if } V > A \text{ then } A \text{ else } V \end{aligned}$$

$$3. \begin{aligned} V &:= \sum_i \omega_i V_i, \\ V &:= \text{if } V > h \text{ then } V \text{ else } 0 \end{aligned}$$

4.

$$\begin{aligned} V &:= \sum_i \omega_i V_i, \\ V &:= \text{if } V \geq h \wedge V < 1 \text{ then } V - h \text{ else if } V \geq 1 \text{ then } 1 - h \text{ else } 0 \end{aligned}$$

$$5. \begin{aligned} V &:= \frac{1}{n} \sum_i \omega_i V_i, \\ V &:= \text{if } V \geq h \text{ then } V \text{ else } 0, \end{aligned}$$

где n – количество активных входов нейрона

Поясним последнее требование к передаточной функции. Оно отражает, например, распознавание букв и знаков препинания. При этом целесообразно использовать передаточную функцию 3. Однако распознавание, например, запятой и буквы "А" приводит к резкому различию величин возбуждения соответствующих нейронов выходного слоя. При условии "шумов" запятая становится практически неразличимой, что приводит к необходимости преобразования величин возбуждения нейронов выходного слоя в единый диапазон изменения. Это либо достигается вводом в рассмотрение коэффициентов приведения, как в [разделе 4.3](#), - для одинаковой коррекции всех весов

связей каждого нейрона выходного слоя, либо приходится решать эту проблему отдельно для каждой связи такого нейрона, как будет показано далее. В рассматриваемом примере проблема уравнивания сигналов на выходном слое для различных *эталонов* также актуальна.

Таким образом, выбор и модификация передаточной функции производятся экспериментально, хотя легко установить общие черты функций, рекомендуемых здесь.

9.4. Анализ примера

Проверим, достаточны ли наши действия по построению *нейросети*. Показывает ли она на правильные решения по тем *эталонным ситуациям*, по которым создавалась логическая схема? Однозначен ли ее ответ при предъявлении различных *эталонных ситуаций*? Одинаковы ли величины возбуждения нейронов выходного слоя при предъявлении различных *эталонов*, что служит помехоустойчивости *нейросети* и возможности ее вложения в другие *нейросети* при формировании "длинных" логических цепочек рассуждений? Необходима ли коррекция параметров сети (порогов и весов связей) для ее правильной работы?

Отметим, что опыт исследований склоняет в пользу преимущественного применения передаточной функции 3. Она обладает таким важным свойством (если позволяет порог), как ассоциативность, позволяющая "собирать" сигнал независимо от пути прохождения возбуждения.

Выберем эту передаточную функцию, предположив, что веса всех связей равны единице, общий для всех нейронов порог $h = 0,3$.

Рассчитаем для различных *эталонных ситуаций* значения возбуждения нейронов выходного слоя и, следовательно, определим принимаемые решения. Расчеты сведены в табл. 9.1.

Таблица 9.1. Примеры расчета принимаемых решений

Решение	R_1	R_1	R_3	R_2	R_2	R_3	R_4	R_4	R_4	R_5	R_5
	R_2	R_4									

Возбуждение нейронов выходного слоя	R_5	0	0	0	2	0	0	0	2	3
	R_4	1	1	1	1	1	3	3	2	1
	R_3	1	1	1	0	3	1	0	0	0
	R_2	4	0	5	3	0	2	0	1	1
	R_1	4	1	2	0	1	1	1	0	0
Возбуждение нейронов промежуточного слоя	11	0	0	0	1	0	0	0	1	1
	10	0	0	0	1	0	0	0	1	2
	9	0	0	0	1	0	0	0	2	1
	8	0	1	0	0	1	1	2	0	0
	7	1	0	1	0	0	2	1	0	0
	6	0	1	0	0	2	0	0	0	0
	5	1	0	1	0	1	1	0	0	0
	4	0	0	0	3	0	0	0	1	1
	3	2	0	2	0	0	1	0	0	0
	2	2	0	3	0	0	1	0	0	0
	1	4	0	2	0	0	1	0	0	0
Ситуация	X_{15}	0	0	0	0	0	0	0	1	0
	X_{14}	0	0	0	1	0	0	0	0	0
	X_{13}	0	0	0	0	0	0	0	0	0
	X_{12}	0	0	0	1	0	0	0	0	0
	X_{11}	0	0	0	0	0	0	0	0	1
	X_{10}	0	1	0	0	1	0	1	0	0
	X_9	0	0	0	0	0	1	1	0	0
	X_8	0	0	0	0	1	0	0	0	0
	X_7	1	0	1	0	0	0	0	0	0
	X_6	0	0	0	0	0	0	0	0	0
	X_5	1	0	0	0	0	0	0	0	0
	X_4	0	0	1	0	0	0	0	0	0
	X_3	1	0	0	0	0	0	0	0	0
	X_2	0	0	0	1	0	0	0	1	1

	X_1	1	0	1	0	0	1	0	0	0
	№	1	2	3	4	5	6	7	8	9

Анализируя таблицу, видим, что даже при предъявлении *эталонов* сеть работает неправильно. По некоторым *эталонам* (столбцы 1, 2 и 8) она дает неоднозначный ответ.

Тем самым сеть демонстрирует "побочный эффект". Из-за наличия общих событий, составляющих разные *ситуации*, эти события определяют одинаковый исход даже в том случае, если другие события обеспечивают различие *ситуаций*. Так, конъюнкция $X_1 \wedge X_7$, определившая вместе с событиями X_3 и X_5 правильное решение R_1 , дважды (нейроны 2 и 3) при использовании передаточной функции 3 участвуют в формировании возбуждения нейрона R_2 (столбец 1). Аналогичный вывод следует из анализа столбца 8. Необходимо подавить "побочный эффект", снизив величину возбуждения нейронов, не участвующих в возбуждении "нужного" нейрона выходного слоя. (Столбец 2 пока обсуждать не будем.)

Далее, анализ таблицы показывает, что даже при правильном ответе величины возбуждения нейронов выходного слоя, закрепленные за разными решениями, различны. Более того, эти величины могут различаться даже при предъявлении *эталонов*, требующих одного и того же решения (столбцы 1 и 2, 3 и 4, 8 и 9).

Таким образом, требуется *корректировка* параметров *нейросети*.

Корректировка параметров, дистрибутивные преобразования, однослойные и совершенные логические нейронные сети

Для логической нейронной сети, составленной по ранее рекомендуемой схеме на основе произвольной формы логического описания системы принятия решений, предлагаются способы корректировки весов и порогов. Такая корректировка призвана "спасти" нейросеть, обеспечив корректность и однозначность предлагаемых ею решений. Для этой же цели предлагаются дистрибутивные преобразования логического описания системы принятия решений. Обсуждается возможность построения однослойных и совершенных логических нейронных сетей, а также проблема корректного задания исходных данных.

"И одноглазый Грач сказал свое мнение:

– Бенья говорит мало, но он говорит смачно. Он говорит мало, но хочется, чтобы он сказал еще что-нибудь".

И. Бабель. "Одесские рассказы"

10.1. Корректировка порогов

В лекции 2 показано, что при моделировании конъюнктора с помощью нейрона для того, чтобы этот нейрон приходил в возбужденное состояние только после прихода сигналов возбуждения от всех связанных с ним нейронов, необходимо задать ему высокое значение *порога*. Так, рассчитывая на максимальную единичную достоверность (аналог булевой единицы) высказываний, считаем, что нейрон 1 должен прийти в возбужденное состояние, если $X_1 = X_3 = X_5 = X_7 = 1$, т.е. предъявлен соответствующий эталон. Это возможно, если *порог* h_1 превышает значение 3, равное уменьшенному на единицу числу возбуждаемых входов. Однако этот порог должен быть преодолен и в том случае (а это и является достоинством нейросети), если отдельные события, составляющие ситуацию, появляются с достоверностью, меньшей единицы. В этом случае необходимо так подобрать пороги для всех нейронов, прообразами которых являются

конъюнкторы, чтобы "нужные" нейроны возбуждались, а "побочные эффекты" исключались.

Это требование приводит к важной исходной предпосылке создания нейросети, основанной на понятии существенности события.

На этапе проектирования нейросети выбирается некоторое значение N достоверности. Тогда событие является существенным, если его достоверность не ниже значения N .

Это не означает, что пользователь ограничен значениями предполагаемой достоверности событий, которую он задает на *входном слое*. Во-первых, в корректно представленной СПР, использующей исчерпывающие множества событий, низкая достоверность одних событий свидетельствует в пользу высокой достоверности других. Во-вторых, значение N – предпочтительная граница достоверности, учитываемая при формировании *порога*. Более того, это значение может характеризовать лишь среднюю величину возбуждения, подаваемого на один вход нейрона, взявшего на себя роль конъюнктора.

В данном примере, дабы избежать побочного эффекта, необходимо, чтобы нейроны 2 и 3 не возбуждались при высоких значениях X_1 и X_7 в то время, когда высокое значение возбуждения имеют нейроны X_3 и X_5 . Они должны возбуждаться при высоком значении возбуждения нейронов X_4 и X_6 . Однако в этом случае нейрон 1 должен возбуждаться лишь при высоком уровне возбуждения нейронов X_3 и X_5 .

Учитывая приблизительность и неточность, лежащие в основе имитации *ассоциативного мышления* с помощью нейросети, необходимо согласиться с эвристическим или экспериментальным выбором *порогов*, используемых при счете *передаточной функции* нейронами-конъюнкторами, предоставив лишь рекомендации.

Обозначим n_i – количество активных входов нейрона i . Пусть $h_i = Nn_i, i = 1, \dots, 11$. Положим $N = 0,7$. Тогда (рис. 9.15) $h_1 = 2,8, h_2 = h_3 = h_4 = 2,1, h_5 = \dots = h_{11} = 1,4$.

Рассчитаем значения возбуждения нейронов выходного слоя для

ситуаций, отображенных в табл. 10.1, и, в частности, приведших к неоднозначности решения.

Для ситуации 1 имеем $V_1 = 4 (4 > 2, 8)$, $V_2 = V_3 = 0 (2 < 2, 1)$, $V_4 = \dots = V_{11} = 0$, $R_1 = 4$, $R_2 = \dots = R_5 = 0$.

Для ситуации 2 находим $V_1 = \dots = V_{11} = 0$, $R_1 = 1$, $R_2 = \dots = R_5 = 0$.

Для ситуации 3 имеем $V_1 = 0 (2 < 2, 8)$, $V_2 = 3 (3 > 2, 1)$, $V_3 = 0 (2 < 2, 1)$, $V_4 = \dots = V_{11} = 0$, $R_1 = 0$, $R_2 = 3$, $R_3 = R_4 = R_5 = 0$.

Ситуация 4: $V_1 = V_2 = V_3 = 0$, $V_4 = 3 (3 > 2, 1)$, $V_5 = \dots = V_{11} = 0$, $R_1 = 0$, $R_2 = 3$, $R_3 = R_4 = R_5 = 0$.

Ситуация 5: $V_1 = \dots = V_5 = 0$, $V_6 = 2 (2 > 1, 4)$, $V_7 = V_{11} = 0$, $R_1 = R_2 = 0$, $R_3 = 2$, $R_4 = R_5 = 0$.

Для ситуации 8 имеем $V_1 = \dots = V_8 = 0$, $V_9 = 2 (2 > 1, 4)$, $V_{10} = V_{11} = 0$, $R_1 = R_2 = R_3 = 0$, $R_4 = 2$, $R_5 = 0$.

Проверив остальные эталоны, убеждаемся в том, что неоднозначность решения, принимаемого по всем эталонным ситуациям, ликвидирована. (Не появится ли она при последующих действиях?)

10.2. Корректировка весов связей нейронов выходного

ДНФ наглядно демонстрирует смысл задачи, так как каждая конъюнкция, включая содержащую единственное высказывание, соответствует отдельной ситуации. Дизъюнкция таких конъюнкций определяет множество ситуаций, приводящих к одному решению. Отсюда, прообразами нейронов выходного слоя являются *дизъюнкторы*, реализующие операцию ИСКЛЮЧАЮЩЕЕ ИЛИ, т.е. предполагающие анализ не более чем одного единичного сигнала на входе.

Выше было установлено, что при замене логических операций счетом единой *передаточной функции* даже для эталонных ситуаций возможен побочный эффект, при котором на один нейрон выходного слоя для совместной обработки поступают высокие сигналы возбуждения более чем одного нейрона, т.е. отображающие одновременное наличие разных ситуаций. Это способствует неоднозначности решений.

Показано, что с помощью *порогов* можно ликвидировать побочный эффект, и мы вправе считать, что высокий сигнал возбуждения, соответствующий действительной ситуации, воспринимается нейроном выходного слоя с единственного входа, что в большей степени адекватно имитируемой логической операции.

Однако величина возбуждения нейронов выходного слоя осталась различной в связи с различной величиной возбуждения нейронов скрытого (промежуточного) слоя.

Введем *веса* связей (входов) каждого нейрона выходного слоя, равные обратной величине уровня сигнала при предъявлении эталонной ситуации, поступающего на каждый вход. Этим мы добьемся принадлежности величин возбуждения нейронов выходного слоя диапазону $[0, 1]$. Этот диапазон будет реализован при "грамотном" задании достоверности событий на рецепторном входе.

Выбранный диапазон уравнивает роль входного и выходного слоев в случае, если при создании "длинных" логических цепочек *умозаключений* реализуется вложенность нейросетей. Это означает, что выходной слой одной нейросети тотчас используется в качестве *входного слоя* другой.

В исследуемом примере *веса* связей вводятся на основе анализа величин возбуждения нейронов выходного слоя при рассмотрении всех возможных эталонов, как представленных в [табл. 9.1](#), так и немногих оставшихся. Полученные *веса* отображены на [рис. 9.15](#).

Однако после *корректировки весов*, приведшей к одинаковому, единичному значению возбуждения нейронов выходного слоя в ответ на каждую эталонную ситуацию, вновь замечаем возникшую неоднозначность решения. А именно, высказывание X_{10} образует

эталонную ситуацию, приводящую к решению R_1 . В то же время X_{10} участвует в создании ситуаций $X_8 \wedge X_{10}$ и $X_9 \wedge X_{10}$, приводящих к решениям R_3 и R_4 соответственно. То есть задание эталона $X_8 = X_{10} = 1$ приводит в единичное возбуждение нейроны R_1 и R_3 , а задание $X_9 = X_{10} = 1$ порождает значения $R_1 = R_4 = 1$.

Исследуя пути исправления ошибки, приходим к выводу, что в основе обнаруженной коллизии лежит некорректное описание СПР, необнаруженное противоречие в задании на разработку. Еще на этапе составления логической схемы (рис. 9.15) мы могли обнаружить получение единичных сигналов не на единственном выходе.

Тогда вспоминаем, что бесплатный билет – проездной документ – предъявляется обязательно вместе с удостоверением работника МПС!

Дополним описание (9.1), уточнив логическое выражение для R_1 :

$$(X_1 \wedge X_3 \wedge X_5 \wedge X_7) \vee (X_{10} \wedge X_{13}) \rightarrow R_1 \quad (10.1)$$

Появление новой конъюнкции высказываний приводит к увеличению числа нейронов промежуточного слоя (рис. 10.1), т.к. вместо связи $X_1 \rightarrow R_1$ вводится нейрон 12. Порог $h_{12} = 1,4$. Вес связи нейрона 12 с нейроном R_1 принимает значение 0,5.

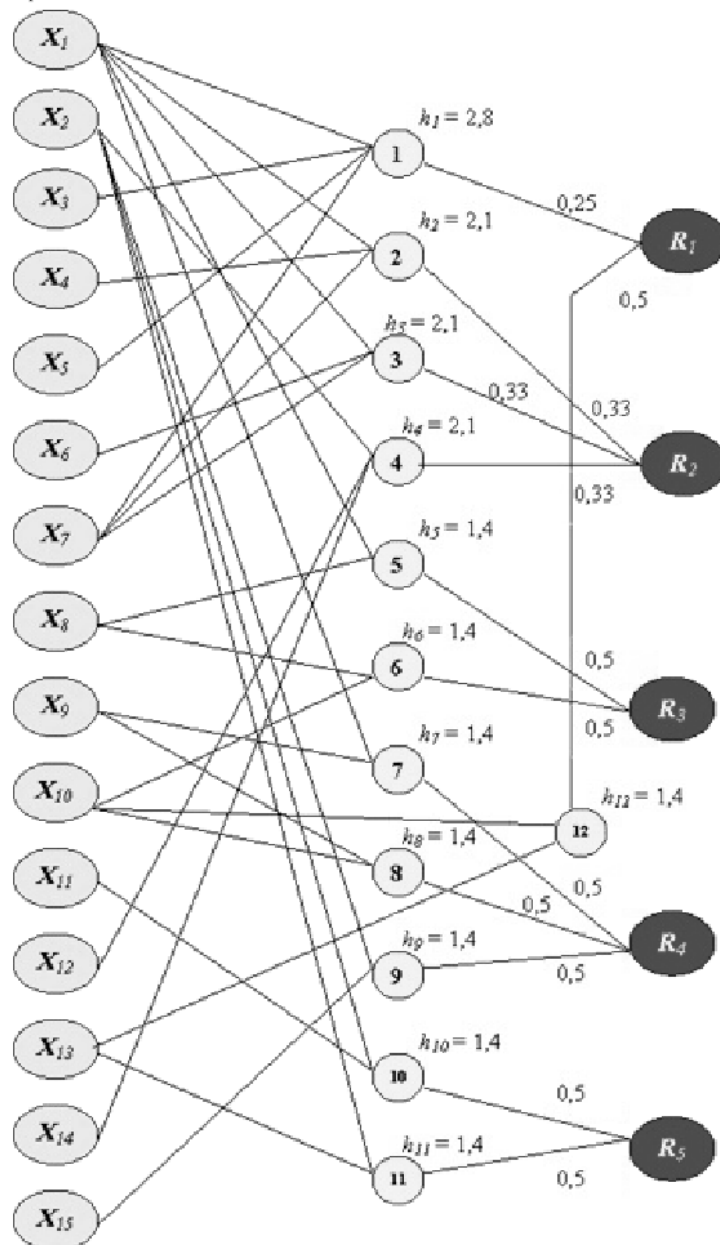


Рис. 10.1. Нейросеть после уточнения логического описания

В табл. 10.1 представлены результаты расчета принимаемых решений по предъявленным эталонным ситуациям (для проверки правильности) и по ряду неопределенных ситуаций (столбцы 8-10). Анализ таблицы

показывает, что решения, принятые нейросетью, вполне объяснимы. Это относится даже к неоднозначному решению (столбец 9).

Таблица 10.1. Расчет принимаемых решений по уточненной нейросети после корректировки порогов и весов связей

Решение	R_1	R_2	R_2	R_2	R_3	R_4	R_5	R_1	R_4 R_5	R_2
Возбуждение нейронов выходного слоя	R_5	0	0	0	0	0	1	0	0,75	0
	R_4	0	0	0	0	0	1	0	0,75	0
	R_3	0	0	0	0	1	0	0	0	0
	R_2	0	1	1	1	0	0	0	0,77	0
	R_1	1	0	0	0	0	0	0	0,87	0
Возбуждение нейронов промежуточного слоя	12	0	0	0	0	0	0	0	0	0
	11	0	0	0	0	0	2	0	1,5	0
	10	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	1,5	0
	8	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	2	0	0	0
	6	0	0	0	0	2	0	0	0	0
	5	0	0	0	0	0	0	0	0	0
	4	0	0	0	3	0	0	0	0	2,5
	3	0	0	3	0	0	0	2,3	0	0
	2	0	3	0	0	0	0	0	0	0
1	4	0	0	0	0	0	3,1	0	0	
X_{15}	X_{15}	0	0	0	0	0	0	0	0,6	0
	X_{14}	0	0	0	1	0	0	1	0	1
	X_{13}	0	0	0	0	0	1	0	0,6	0
	X_{12}	0	0	0	1	0	0	0	0	1
	X_{11}	0	0	0	0	0	0	1	0,4	0
	X_{10}	0	0	0	0	1	0	0	0,1	0
	X_9	0	0	0	0	0	1	0	0	0,3

Ситуация	X_8	0	0	0	0	1	0	0	0,1	0	0,3
	X_7	1	1	1	0	0	0	0	0,9	0	0,3
	X_6	0	0	1	0	0	0	0	0,5	0	0,5
	X_5	1	0	0	0	0	0	0	0,5	0	0,5
	X_4	0	1	0	0	0	0	0	0,2	0	0,5
	X_3	1	0	0	0	0	0	0	0,8	0	0,5
	X_2	0	0	0	1	0	0	1	0,1	0,9	0,5
	X_1	1	1	1	0	0	1	0	0,9	0,1	0,5
	№	1	2	3	4	5	6	7	8	9	10

10.3. Дистрибутивная форма логического описания системы принятия решений

Приведенные выше построения предполагали представление логических выражений, описывающих СПР, в ДНФ. Однако известно, что каждое логическое выражение на основе алгебры высказываний может быть представлено и в виде *конъюнктивной нормальной формы (КНФ)*. Такая форма также приводит к достаточности не более чем двухслойной нейросети. Мы не будем приводить такое представление для анализируемого примера, усложнившее описание СПР, но не приведшее к новым интересным выводам, однако сделаем общее замечание о важности минимизации количества слоев нейронной сети.

Развитие систем принятия решений, таких как медицинская или техническая диагностика, прогнозирование рынка ценных бумаг, управление перевозками железнодорожным транспортом и др., приводит к весьма большому числу исследуемых факторов и, следовательно, к еще большему числу нейронов сети. Расчет для каждого нейрона даже несложной *передаточной функции* в общем цикле обработки всех нейронов (в соответствии с их частичной упорядоченностью на основе преимственности информации) может привести к значительным временным затратам, ставящим под сомнение оперативность управления.

Необходима "аппаратная поддержка", что породило большое число предложений [14] в области разработки *нейрокомпьютеров (НК)*,

реализующих нейросети. В основе НК лежит принцип *распараллеливания вычислений*, что фактически означает распределение нейронов (программных процедур, моделей *нейроподобных элементов* и т.д.) между исполнительными вычислительными устройствами – процессорами для их обработки. Эффективны НК, выполненные в виде приставки к персональному компьютеру или рабочей станции. Ориентация НК на обработку универсальной двухслойной нейросети на основе ДНФ жестко распределяет функции слоев и возможные связи его процессоров, используя элементы стандартизации и унификации.

В лекции 1 обсуждалась проблема минимизации длины логической цепочки и было показано, что с помощью "размножения" решений такая длина может быть даже доведена до единичной.

Кроме того, известно [6], что эффективность распараллеливания, основным компонентом которой является минимум времени выполнения комплекса частично упорядоченных работ, зависит от длины *критического пути* в *информационном графе*, каким отображается *нейросеть*. Эта длина должна быть минимальной, что и обеспечивается не более чем двумя слоями нейросети. Следовательно, такая сеть гарантирует возможность оптимального распараллеливания.

Подобное заключение известно и специалистам по схемотехнике, всегда стремящимся минимизировать максимальную длину логической цепочки в схеме разрабатываемого устройства, влияющей на длительность такта его работы.

Однако, требуя обязательного представления логических выражений, описывающих СПР, в дизъюнктивной нормальной форме, мы предполагаем скобочную запись, с открытыми относительно операции конъюнкции некоторыми потенциальными скобками. Это, например, приводит к тому, что в выражении (9.1), определяющем решение R_2 , дважды присутствует конъюнкция $X_1 \wedge X_7$. В выражении, определяющем решение R_3 , дважды присутствует событие X_8 и т.д. При построении нейронной сети по логической схеме с использованием *передаточной функции*, суммирующей сигналы, это привело к неоднократному учету некоторых сигналов в качестве слагаемых. Так (см. табл. 9.1), расчет величины возбуждения нейрона R_2 при предъявлении эталона 1 (по нему максимального возбуждения должен

достичь нейрон R_1) требует сложения величин возбуждения нейронов 2 и 3, каждая из которых формируется сложением сигналов X_1 и X_7 . Это и привело к побочному эффекту, с которым пришлось бороться соответствующим повышением *порогов*.

Тогда справедливо предположение о том, что, наоборот, вынесение за скобки способно не только упростить запись, обеспечив единственное вхождение переменных, но и тем самым исключить побочный эффект.

Для этого воспользуемся правилами *эквивалентных преобразований*. Определяющим в данном применении является *симметричное свойство* дистрибутивности конъюнкции относительно дизъюнкции и наоборот. (В алгебре применим лишь дистрибутивный, распределительный закон умножения относительно сложения.) Данное свойство (см. лекцию 1) записывается как

$$\begin{aligned}(x \vee z) \wedge (y \vee z) &= z \vee (x \wedge y), \\ (x \wedge z) \vee (y \wedge z) &= z \wedge (x \vee y)\end{aligned}$$

Предполагается возможность такого преобразования произвольной логической функции – композиции операций конъюнкции и дизъюнкции, при котором обеспечивается единственность вхождения каждой переменной в полученную запись. Основная операция, которая при этом используется – вынесение за скобку.

Однако, как следует из примера, рассмотренного в лекции 1, такое преобразование не всегда приводит к успеху. Это свидетельствует о том, что в крайнем случае, для обеспечения единственности вхождения переменных, все же следует прибегать к способу "размножения" решений. Такое вынужденное "размножение" также отнесем к акту *дистрибутивного преобразования*.

Важность используемого в этом преобразовании свойства дистрибутивности приводит к целесообразности его учета в названии соответствующего *логического описания* СПР. Полученную таким образом *нейросеть* справедливо назвать нейросетью на основе дистрибутивного *логического описания* или просто дистрибутивной.

Упростим описание СПР, воспользовавшись операциями вынесения за

скобки и выделения общих выражений:

$$\begin{aligned}
 & ((X_1 \wedge X_7) \wedge (X_3 \wedge X_5)) \vee (X_{10} \wedge X_{13}) \rightarrow R_1, \\
 & ((X_1 \wedge X_7) \wedge (X_4 \vee X_6)) \vee (X_2 \wedge X_{12} \wedge X_{14}) \rightarrow R_2, \\
 & X_8 \wedge (X_1 \vee X_{10}) \rightarrow R_3, \\
 & (X_9 \wedge (X_1 \vee X_{10})) \vee (X_2 \wedge X_{15}) \rightarrow R_4, \\
 & X_2 \wedge (X_{11} \vee X_{13}) \rightarrow R_5.
 \end{aligned} \tag{10.2}$$

Наблюдается единственность вхождения каждого высказывания в описание логической функции (для сравнения см. (9.1)). При этом можно считать, что нам "повезло": "размножить" решения не потребовалось.

Форма представления (10.2) не только стала проще, она стала естественнее и понятнее. Однако наличие вложенных скобок вселяет сомнение в возможность использования не более чем двух слоев нейросети.

На рис. 10.2 (пока не следует обращать внимание на веса связей) отображена логическая схема реализации (10.2), она же – дистрибутивная *нейросеть*. Чтобы "не потерять" информацию, в кружках, обозначающих нейроны, указан тип логической операции преобразы – конъюнктора или дизъюнктора.

Выберем ту же *передаточную функцию* 3 и проверим, необходима ли здесь коррекция *порогов* и *весов* связей. Предварительно отметим, что описание (10.2) затрудняет перебор эталонных ситуаций. Для удобства их формирования все же воспользуемся аналогичной системой (9.1)-(10.1). Выберем малое значение *порога* $h = 0,3$, "не мешающее" суммированию сигналов, и, исключая очевидные вычисления, проанализируем отображение эталонных ситуаций.

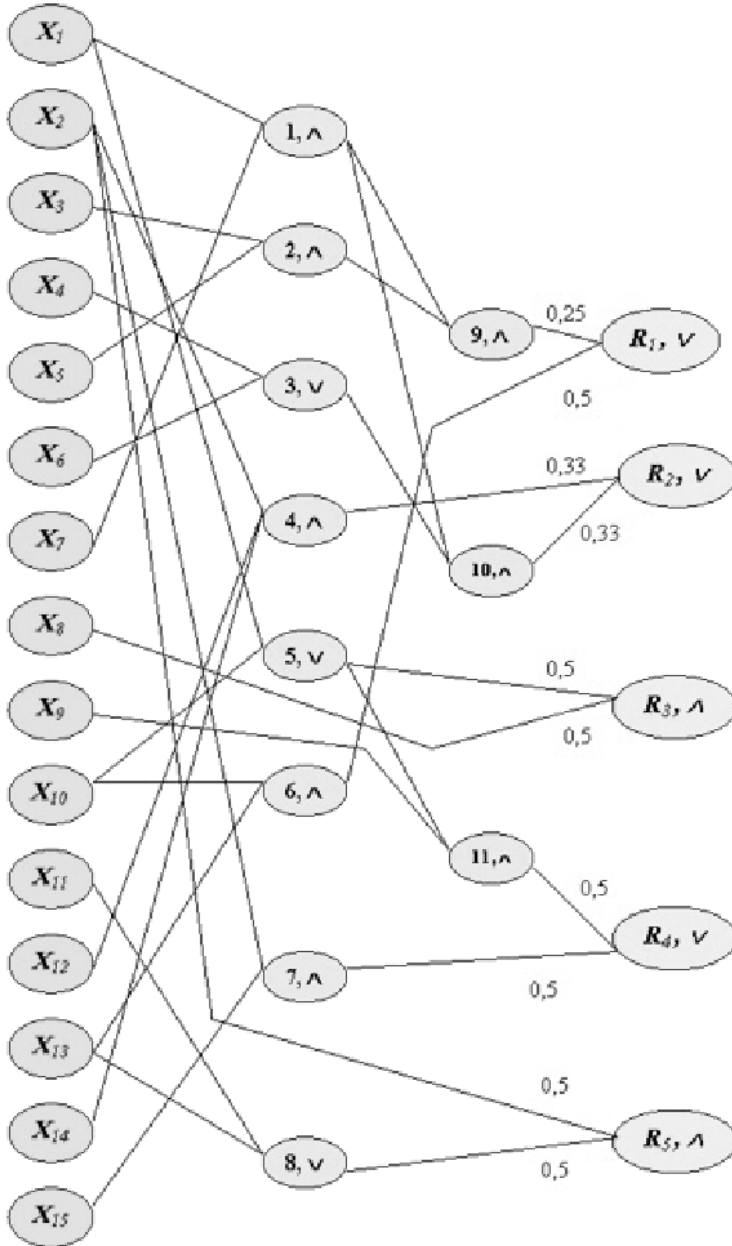


Рис. 10.2. Дистрибутивная нейросеть

Анализ показывает, что побочные эффекты отсутствуют. Это легко было предвидеть, т.к. выше их появление объяснялось неоднократным вхождением некоторых переменных-высказываний в одни и те же

логические выражения при применении аддитивной *передаточной функции*.

При корректном задании исходной информации, т.е. при правильно сформулированном запросе на основе "физического смысла" СПР, имитируются функции *конъюнкторов* и *дизъюнкторов* – прообразов нейронов. То есть высокий уровень сигнала нейрона, принявшего роль конъюнктора, возникает в том случае, если все его входы предельно возбуждены. Нейроны, принявшие на себя роль *дизъюнкторов*, могут возбуждаться лишь при максимальном возбуждении одного из входов.

Однако величины возбуждения нейронов выходного слоя различны. Различаются и сигналы, приходящие на один нейрон от разных эталонных ситуаций.

Как и ранее, уточним *веса* связей нейронов выходного слоя (рис. 10.2). А именно, если нейрон принял роль дизъюнктора, то *вес* каждой связи уточняется отдельно по величине возбуждения, формируемой на основе соответствующего эталона. Новый *вес* равен единице, деленной на эту величину.

Если нейрон принял роль конъюнктора, то все *веса* его связей, до того равные единице, делятся на сумму сигналов, пришедших по всем связям при предъявлении соответствующего эталона, т.е. *веса* сохраняют одинаковые значения, а величина возбуждения этого нейрона становится равной единице.

Составим табл. 10.2 на основе расчета ситуаций, отображенных в табл. 10.1, и проанализируем ее. По эталонам (представлены не все, остальные легко рассчитать) сеть работает правильно.

По столбцу 8 видим, что сеть высказалась за решение R_2 . Этому решению следует больше доверять, т.к. высокое значение *порога* ранее существенно уменьшало величину распространяемого сигнала. Значение возбуждения R_2 , превышающее единицу, следует отнести за счет некорректности вопроса: предъявление пенсионного удостоверения или удостоверения работника МПС должно оцениваться исходя из *условной вероятности*, а именно, при условии, что билет не предъявлен. Таким образом, исходные оценки, подаваемые на

рецепторный слой, предполагают структуризацию вопросов.

Столбец 9 демонстрирует довлеющую роль пенсионного удостоверения или удостоверения работника МПС, несмотря на подозрение на взятку, что исключает неоднозначность решения в табл. 10.1.

Столбец 10 свидетельствует о том, что полная неопределенность, следующая из предъявленных проездных документов, наказуема, хотя и малым штрафом.

Таблица 10.2. Расчет решений по дистрибутивной нейросети

Решение	R ₁	R ₂	R ₂	R ₂	R ₃	R ₄	R ₅	R ₂	R ₅	R ₂	
Возбуждение нейронов выходного слоя	R ₅	0	0	0	0,25	0	0	1	0,55	0,95	0,25
	R ₄	0,5	0,5	0,5	0,5	0,5	1	0,5	0,55	0,8	0,65
	R ₃	0,33	0,5	0,5	0	1	0,5	0	0,65	0,05	0,4
	R ₂	0,67	1	1	1	0	0,33	0,33	1,2	0,33	1,43
	R ₁	1	0,5	0,5	0	0,5	0,25	0,5	0,93	0,33	0,45
Возбуждение нейронов промежуточного слоя	11	1	1	1	0	1	2	0	1	0,1	0,8
	10	2	3	3	0	0	1	0	2,5	0,1	1,8
	9	4	2	2	0	0	1	0	3,1	0,1	1,8
	8	0	0	0	0	0	0	1	1	1	0
	7	0	0	0	1	0	0	1	0,1	1,5	0,5
	6	0	0	0	0	1	0	1	0,1	0,6	0
	5	1	1	1	0	1	1	0	1	0,1	0,5
	4	0	0	0	3	0	0	1	1,1	0,9	2,5
	3	0	1	1	0	0	0	0	0,7	0	1
	2	2	0	0	0	0	0	0	1,3	0	1
1	2	2	2	0	0	1	0	1,8	0,1	0,8	
X ₁₅	X ₁₅	0	0	0	0	0	0	0	0	0,6	0
	X ₁₄	0	0	0	1	0	0	0	1	0	1
	X ₁₃	0	0	0	0	0	0	1	0	0,6	0
	X ₁₂	0	0	0	1	0	0	0	0	0	1
	X ₁₁	0	0	0	0	0	0	0	1	0,4	0

Ситуация	X_{10}	0	0	0	0	1	0	0	0,1	0	0
	X_9	0	0	0	0	0	1	0	0	0	0,3
	X_8	0	0	0	0	1	0	0	0,1	0	0,3
	X_7	1	1	1	0	0	0	0	0,9	0	0,3
	X_6	0	0	1	0	0	0	0	0,5	0	0,5
	X_5	1	0	0	0	0	0	0	0,5	0	0,5
	X_4	0	1	0	0	0	0	0	0,2	0	0,5
	X_3	1	0	0	0	0	0	0	0,8	0	0,5
	X_2	0	0	0	1	0	0	1	0,1	0,9	0,5
	X_1	1	1	1	0	0	1	0	0,9	0,1	0,5
	№	1	2	3	4	5	6	7	8	9	10

Таким образом, *нейросеть* отражает здравый смысл, склоняясь к вполне обоснованному, подобно карточному гаданию, решению.

10.4. Возможность применения однослойных нейросетей

Исследуем *передаточную функцию* Z и использующую ее *нейросеть* на рис. 10.2. Анализируя прохождение сигнала по нейросети при предъявлении s -й эталонной ситуации, т.е. подавая на вход соответствующую комбинацию ps единиц, видим, что при единичных *весах* связей нейронов выходного слоя значение возбуждения нейрона, соответствующего решению, равно $n \cdot s$. Эта величина не зависит от пути прохождения возбуждения каждого рецептора, т.е. от того, сколько нейронов было на его пути. Сказывается свойство ассоциативности выбранной *передаточной функции* при малом значении *порога*. В этом случае свойство ассоциативности вырождается в свойство аддитивности, где функция от набора значений равна сумме функций от каждого значения.

Действительно, комбинация $X_1 = X_3 = X_5 = X_7 = 1$ приводит к величине возбуждения нейрона R_1 , равной 4, значение $X_{10} = 1$ приводит к единичному возбуждению того же нейрона, комбинация X_1

$= X_6 = X_7 = 1$ приводит к величине возбуждения нейрона R_2 , равной 3, и т.д.

Возникает вопрос: не проще ли сформировать *нейросеть*, приблизив ее к табличному виду с помощью связей, непосредственно соединяющих множество тех рецепторов, которые образуют ситуации, приводящие к одному решению, с нейроном, закрепленным за этим решением?

Итак, создавая *нейросеть* на основе логической схемы обработки высказываний, мы заменили операции \wedge и \vee единой операцией, выполняемой передаточной функцией. Обозначим эту операцию $\&$. С учетом ее ассоциативности, реализуемой нейронами "на пути" распространения сигнала возбуждения, представим описание (10.2) в виде, соответствующем действительному нейросетевому воплощению:

$$\begin{aligned} X_1 \& X_3 \& X_5 \& X_7 \& X_{10} \& X_{13} &\rightarrow R_1, \\ X_1 \& X_2 \& X_4 \& X_6 \& X_7 \& X_{12} \& X_{14} &\rightarrow R_2, \\ X_1 \& X_8 \& X_{10} &\rightarrow R_3, \\ X_1 \& X_2 \& X_9 \& X_{10} \& X_{15} &\rightarrow R_4, \\ X_2 \& X_{11} \& X_{13} &\rightarrow R_5. \end{aligned} \quad (10.3)$$

Множество всех высказываний или соответствующих им рецепторов, участвующих в формировании одного решения, в лекции 3 была названа обобщенной ситуацией.

Множество $\{X_1, X_3, X_5, X_{10}\}$ является обобщенной ситуацией, аналогично – множество $\{X_1, X_2, X_4, X_6, X_7, X_{12}, X_{14}\}$ и т.д.

В сущности, это не означает, что при формировании каждого решения все нейроны-рецепторы, образующие обобщенную ситуацию, должны получать ненулевое возбуждение. Обобщенные ситуации имеют "технологическое" значение. Они объединяют нейроны-рецепторы, между которыми, с одной стороны, и нейроном выходного слоя – с другой должны быть введены связи в *однослойной нейросети*. Нейроны-рецепторы по-прежнему возбуждаются в соответствии с ситуациями – конъюнкциями высказываний, для которых формируется *нейросеть*. А оператор нейросети следит за тем, чтобы "физический

смысл" явлений не пропадал, а именно – чтобы соблюдались права исчерпывающих множеств событий.

По описанию (10.3) построим *однослойную нейросеть* (рис. 10.3, на веса пока не обращаем внимания) и проанализируем ее работу при ситуациях, отображенных в таблицах 10.1 и 10.2.

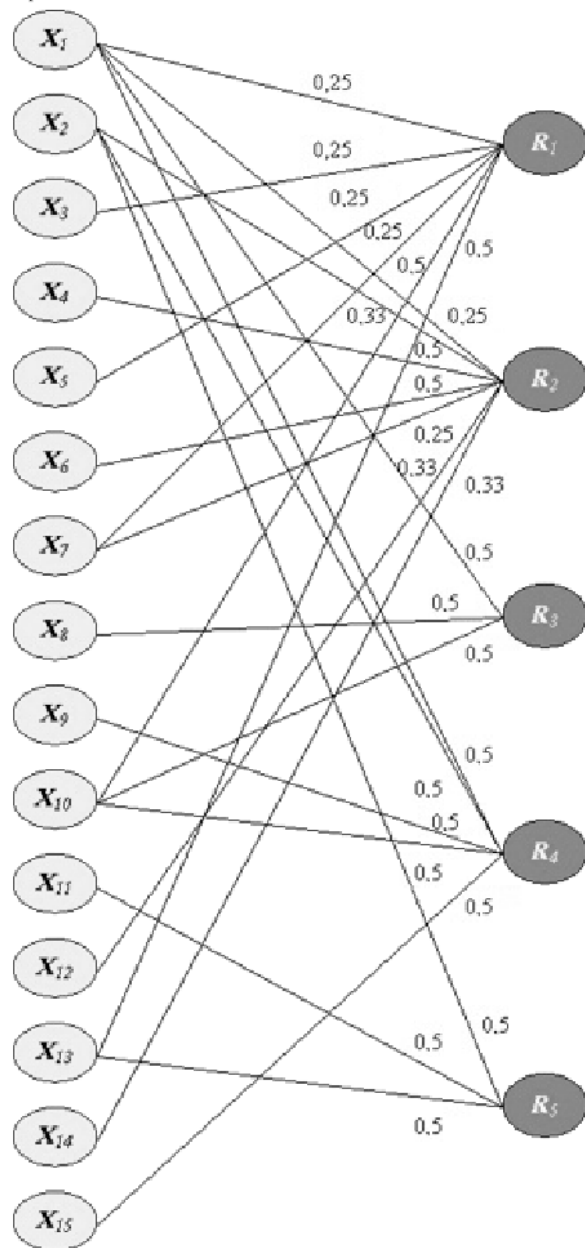


Рис. 10.3. Однослойная нейросеть

Видим, что *нейросеть* в целом делает верные выводы. Взаимно-относительные оценки возбуждения нейронов выходного слоя правильно определяют ситуации, однако абсолютные значения

возбуждения этих нейронов различны для различных ситуаций. Тогда, как и прежде, приведем их к отрезку $[0, 1]$, скорректировав *веса* связей. Для воспроизведения структуры и взаимодействия высказываний и ситуаций придется обратиться к описанию (10.2). Определим по первой логической функции в (10.2), задающей решение R_1 , последнюю операцию. Это операция дизъюнкции.

Она делит множество нейронов обобщенной ситуации на два подмножества $\{X_1, X_3, X_5, X_7\}$ и $\{X_{10}, X_{13}\}$. Закрепим предварительно за ними единичное значение *весов* связей.

В этом разбиении на непересекающиеся подмножества нам помогает отмеченная ранее единственность вхождения каждого выражения в запись логической функции.

Отдельно исследуем сформированные подмножества.

По первому из них находим последнюю выполняемую логическую операцию. Это *операция конъюнкции*, разбивающая подмножество на два меньших: $\{X_1, X_7\}$ и $\{X_3, X_5\}$. Закрепим за каждым из них значение *веса* 0,5 (ранее найденную единицу делим поровну).

Вновь, начиная с первого, анализируем последовательность трех подмножеств. Последняя операция, производимая над элементами первого подмножества, является конъюнкцией. Она делит подмножество на два, каждое из которых содержит единственный элемент. Закрепляем за ними (окончательно, за соответствующими связями) значение *веса* 0,25, т.е. ранее найденное значение 0,5 делим поровну (рис. 10.3).

Анализ следующего подмножества приводит к аналогичному результату. Последнее подмножество содержит два элемента, объединенных в (10.2) операцией конъюнкции. За каждым из них окончательно закрепим *вес* связи, равный 0,5.

Рассмотрим следующую логическую функцию, приводящую к решению R_2 . Здесь последняя производимая операция – дизъюнкция. Она делит множество нейронов, составляющих обобщенную ситуацию, на два подмножества: $\{X_1, X_7, X_4, X_6\}$ и $\{X_2, X_{12}, X_{14}\}$. Присвоим

им предварительно единичные *веса* связей.

Анализируя первое подмножество, находим последнюю логическую операцию – конъюнкцию, разделяющую его на два меньших подмножества $\{X_1, X_7\}$ и $\{X_4, X_6\}$. Введенный ранее единичный *вес* делим поровну, принимая уточненные *веса* связей нейронов равными 0,5.

Вновь анализируем первое из сформированных множеств нейронов. *Операция конъюнкции* делит это подмножество на два, содержащие по одному элементу: $\{X_1\}$ и $\{X_7\}$. Делим поровну найденный ранее *вес*, полагая *веса* соответствующих связей нейрона R_2 равными 0,25.

Последней логической операцией, связывающей высказывания X_4 и X_6 второго подмножества, является дизъюнкция. Она окончательно сохраняет ранее определенный *вес* связей, равный 0,5.

Анализируя третье, последнее подмножество, видим, что в записи, объединяющей высказывания в (10.2), отсутствуют скобки. Это (хотя существует ранжирование логических операций при их выполнении) свидетельствует о том, что все высказывания объединяет одна операция. Так как это – конъюнкция, делим поровну предварительно найденный *вес*, полагая соответствующие *веса* связей равными 0,33.

Аналогично корректируем остальные *веса* связей.

Рассмотрение примера приводит к простому формальному описанию алгоритма коррекции *весов* связей нейронов выходного слоя *однослойной нейросети*, которое здесь не приводим.

В табл. 10.3 даны результаты расчета ситуаций, отображенных в табл. 10.1 и 10.2.

Видим, что по эталонным ситуациям *нейросеть* "работает" правильно, что порождает доверие к ней в процессе эксплуатации. Вполне объяснимы и другие решения, совпадающие с ранее полученными.

Рассуждения об ассоциативности *передаточной функции* приводят к предположению о полном совпадении результатов расчетов

возбуждения нейронов выходного слоя, отображенных в [табл. 10.2](#) и [табл. 10.3](#). Однако это не так. А именно: нахождение *весов* связей внесло коррективы, и результаты расчетов несущественно отличаются. Несущественность этого несовпадения определяется тем, что, во-первых, для всех эталонов совпадают принимаемые решения, во-вторых – для одинаковых ситуаций сохраняется предпочтительный ряд таких решений.

Таблица 10.3. Расчет принимаемых решений по однослойной нейросети

Решение		R ₁	R ₂	R ₂	R ₂	R ₃	R ₄	R ₅	R ₂	R ₅	R ₂
Возбуждение нейронов выходного слоя	R ₅	0	0	0	0,5	0	0	1	0,55	0,95	0,25
	R ₄	0,5	0,5	0,5	0,5	0,5	1	0,5	0,55	0,8	0,65
	R ₃	0,5	0,5	0,5	0	1	0,5	0	0,55	0,05	0,4
	R ₂	0,5	1	1	1	0	0,25	0,25	1,16	0,33	1,54
	R ₁	1	0,5	0,5	0	0,5	0,25	0,5	0,78	0,33	0,45
Ситуация	X ₁₅	0	0	0	0	0	0	0	0	0,6	0
	X ₁₄	0	0	0	1	0	0	0	1	0	1
	X ₁₃	0	0	0	0	0	0	1	0	0,6	0
	X ₁₂	0	0	0	1	0	0	0	0	0	1
	X ₁₁	0	0	0	0	0	0	0	1	0,4	0
	X ₁₀	0	0	0	0	1	0	0	0,1	0	0
	X ₉	0	0	0	0	0	1	0	0	0	0,3
	X ₈	0	0	0	0	1	0	0	0,1	0	0,3
	X ₇	1	1	1	0	0	0	0	0,9	0	0,3
	X ₆	0	0	1	0	0	0	0	0,5	0	0,5
	X ₅	1	0	0	0	0	0	0	0,5	0	0,5
	X ₄	0	1	0	0	0	0	0	0,2	0	0,5
	X ₃	1	0	0	0	0	0	0	0,8	0	0,5
	X ₂	0	0	0	1	0	0	1	0,1	0,9	0,5
	X ₁	1	1	1	0	0	1	0	0,9	0,1	0,5
№	1	2	3	4	5	6	7	8	9	10	

Это означает, что, например, для ситуации 8 по обеим таблицам предпочтительным рядом принимаемых решений (по невозрастанию величин возбуждения нейронов) является последовательность R_2, R_1, R_3, R_4, R_5 . По ситуации 9 – R_5, R_4, R_1, R_2, R_3 . По ситуации 10 – R_2, R_4, R_1, R_3, R_5 .

Однако всегда ли можно так упростить задачу создания СПР, сведя ее к получению *однослойной нейросети*? В разделе 9.1 показано, что это не всегда возможно. Там приведена попытка воспроизведения *логического описания СПР*

$$\begin{aligned} (A_1 \wedge B_1) \vee (A_2 \wedge B_2) &\rightarrow R_1; \\ (A_1 \wedge B_2) &\rightarrow R_2; \\ (A_2 \wedge B_1) &\rightarrow R_3 \end{aligned} \quad (10.4)$$

с помощью *однослойной нейронной сети*, представленной на рис. 9.3. Только полное воспроизведение логической (скобочной) структуры функций, описывающих принимаемые решения, позволило построить систему (рис. 9.4), однозначно определяющую эти решения.

Однако дело не столь безнадежно. Рассмотрим обобщенные эталоны, определяющие каждое решение:

$$\begin{aligned} \{A_1 \& B_1 \& A_2 \& B_2\} &\rightarrow R_1; \\ \{A_1 \& B_2\} &\rightarrow R_2 \\ \{A_2 \& B_1\} &\rightarrow R_3 \end{aligned} \quad (10.5)$$

Видим, что первый обобщенный эталон включает в себя два других. По-видимому, это и является источником неоднозначности при построении *однослойной нейросети*. Ведь в такой сети мы "тянем" связи непосредственно от каждого рецептора к соответствующему нейрону выходного слоя. В этом случае ситуации как бы поглощают друг друга.

Изменим логическое описание, "размножив" одинаковые решения так, чтобы ни один обобщенный эталон не оказался включенным в другой. Для этого введем условно два одинаковых решения $R_{11} = R_{12} = R_1$.

Представим систему (9.6) в виде

$$\begin{aligned}
 A_1 \wedge B_1 &\rightarrow R_{11}; \\
 A_2 \wedge B_2 &\rightarrow R_{12}; \\
 A_1 \wedge B_2 &\rightarrow R_2; \\
 A_2 \wedge B_1 &\rightarrow R_3.
 \end{aligned}
 \tag{10.6}$$

Тогда окончательный вид нейросети (притом – однослойной) представлен на рис. 10.4.

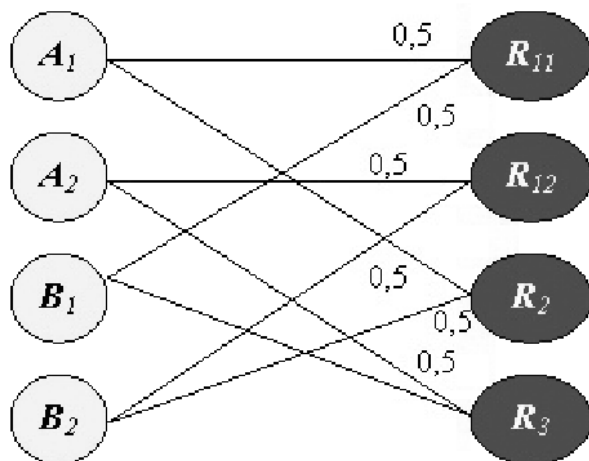


Рис. 10.4. Нейросеть, построенная по описанию (10.6)

Легко видеть, что в ней отсутствует побочный эффект, т.е. *корректировка порогов* не требуется.

Таким образом, для применения *однослойных нейросетей* необходимо (достаточно ли?), чтобы обобщенная ситуация, соответствующая одному решению, не поглощала все нейроны, которые образуют некоторую ситуацию, ведущую к другому решению.

Так, если не была бы найдена *логическая ошибка*, в результате чего ситуация X_{10} в (9.1) не сменилась ситуацией $X_{10} \wedge X_{13}$ в (10.1)-(10.2), обобщенные ситуации, ведущие к решениям R_3 и R_4 , поглотили бы эту ситуацию. Это значит, что при задании эталона $X_8 = X_{10} = 1$ единичное значение возбуждения получали бы нейроны R_1 и R_3 . Аналогично, предъявление эталона $X_9 = X_{10} = 1$ привело бы к

единичному значению возбуждения нейронов R_1 и R_4 .

10.5. Совершенные нейронные сети

В лекции 1 для систем принятия решений введено известное в теории вероятностей понятие *исчерпывающего множества событий* (высказываний). События (высказывания) образуют исчерпывающее множество, если сумма их вероятностей (достоверностей) равна единице (известное условие нормировки).

При организации СПР это означает обязательный учет всех значений, состояний или возможностей использования каждого фактора. Например, множества $\{X_1, X_2\}$, $\{X_7, X_8, X_9\}$ – исчерпывающие множества высказываний о событиях. Однако фактор взятки не образует такого множества, как говорилось ранее, – из-за его недостаточной актуальности в формируемой СПР.

Допустим, что для создаваемой СПР, исходя из специализации, по каждому фактору необходимо учитывать все возможные варианты его значений. То есть рассмотрение каждого фактора при построении нейросети приводит к формированию исчерпывающего множества высказываний относительно него.

Далее, пусть каждая ситуация представляется конъюнкцией, в которой обязательно участвуют высказывания относительно всех факторов, по которым формируется *нейросеть*. Тогда все конъюнкции (ситуации) имеют одинаковое число высказываний.

Пусть любые две отличные друг от друга ситуации приводят к различным решениям. Это означает, что в *логическом описании* СПР отсутствует операция дизъюнкции. Если, исходя из смысла создаваемой СПР, такая операция предполагается, т.е. неединственная ситуация приводит к одному и тому же решению, то с помощью "размножения" решений, как это было сделано в примере предыдущего раздела, можно добиться исключения операции дизъюнкции.

Нейросеть, в которой каждая исследуемая ситуация имеет постоянное число образующих ее событий (высказываний), отображающих все факторы, и в которой взаимно отличающиеся ситуации приводят к

различным решениям, назовем *совершенной* .

Привлекательность *совершенных нейронных сетей* заключается в их сводимости к *однослойным* (которыми они по своей природе и являются). Более того, при применении таких передаточных функций, как 1, 2, 3, 5, не требуется *корректировка порогов*. Вес всех связей (в *однослойной нейросети*) одинаков и равен $1/n$, где n – число используемых факторов.

Кроме того, *совершенная нейросеть*, являющаяся однослойной, позволяет непосредственно и наглядно корректировать влияние каждого фактора на принимаемое решение. Для этого до общей *корректировки весов* необходимо вес соответствующей связи положить равным тому коэффициенту, с которым предполагается учет влияния данного фактора на принимаемое решение. (Таким образом реализуется понятие слабой или сильной зависимости.) После выполнения такой операции для тех весов связей, для которых это необходимо, производится *общая корректировка весов*. Для каждого нейрона выходного слоя она заключается в делении каждого веса на сумму весов n связей, ведущих к данному нейрону. (Фактически это приводит к применению *передаточной функции 5*.)

Рассматриваемый ранее достаточно универсальный пример действий контролера электропоезда, очевидно, не порождает *совершенную нейросеть*, хотя построенная *нейросеть* и сводится к *однослойной*.

Здесь уместен пример из лекции 3, а также модель, где действуют несколько строительных компаний, несколько субподрядных организаций и несколько заводов – изготовителей стройматериалов. В каждой ситуации обязательно представлены все участники: компания, субподрядная организация, завод-изготовитель.

Другим важным примером может служить рассматриваемая в лекции 12 система банковского мониторинга [28], в которой каждая ситуация предполагает обязательное вхождение оценок (по диапазонам изменения) всех факторов:

- *собственный капитал*;
- *сальдированные активы*;

- *ликвидные активы*;
- обязательства до востребования и т.д.

Следует отметить и другие СПР на основе оценки политических, социальных и экономических факторов, а также системы управления на основе конкретных наборов или видов как возмущений, так и регулируемых параметров. Здесь везде речь идет о таблично заданных (на этапе обучения или построения нейросети) функциях многих переменных (векторов) при заданной постоянной размерности. Переменными являются оценки достоверности исследуемых факторов. Скалярным ответом является распределенное возбуждение нейронов выходного слоя, которое, в свою очередь, может указывать на вектор, компоненты которого – конкретные управляющие воздействия, параметры поведения, характеристики рынка, прогноз биржевых сделок и т.д.

Таким образом, *совершенные нейронные сети*, сводящиеся к *однослойным*, по-видимому, имеют самое широкое применение. Усилия по их разработке примитивны и не отвлекают от решения проблемы накопления опыта на практике и в науке: в торговых сделках, сомнительных инвестициях, банковских операциях, а также при моделировании сложных систем управления, исторических и социальных процессов и т.д.

10.6. Корректное задание исходных данных – условие правильности выводов нейросети

Ранее не раз говорилось о "грамотном" задании исходных данных для адекватной реакции нейросети. Опыт пользователя должен развиваться на основе понимания логической и вероятностной взаимосвязей событий, являющихся предметом ее исследований. Необоснованное, случайное задание достоверности высказываний может привести лишь к утверждению известной шутки "каков вопрос – таков ответ".

Мы можем предварительно, на основе интуиции, лишь догадываться, что, например, достоверность высказывания "в билете указана дата (число) этого дня" зависит от достоверности высказывания "пассажир предъявил билет".

В лекции 1 для представления структуры взаимосвязанных событий рассматривались *деревья логических возможностей* (ДЛВ).

Составим *дерево логических возможностей*, связывающее вероятности событий, которые лежат в основе примера СПР (рис. 10.5), анализируемой в настоящей лекции.

При составлении ДЛВ учитывается, что каждое ветвление на основе некоторого события определяет исчерпывающее множество последующих событий при условии наступления данного. Следовательно, на ребрах (не на всех, чтобы не перегружать рисунок) указаны *условные вероятности*. Тогда полная вероятность некоторого события отыскивается как сумма произведений вероятностей, которые найдены по всем путям ДЛВ, приводящим к этому событию.

На основе анализа ДЛВ можно получать рекомендуемые диапазоны задания достоверности некоторых высказываний. Например, вероятность $P(X_7)$ события X_7 (в билете указан текущий год) находится следующим образом:

$$P(X_7) = P_{01}(P_{13}(P_{35}P_{57} + P_{36}P_{67}) + P_{14}(P_{45}P_{57} + P_{46}P_{67})).$$

Для грубой оценки можно считать все элементы даты (число, месяц, год) независимыми и одинаково подверженными искажению. Тогда $P_{13} = P_{35} = P_{45} = P_{57} = P_{67} = P$, $P_{14} = P_{36} = P_{46} = 1 - P$, а $P(X_7) = P_{01}P$.

Если установлено, что в любом билете из-за неразборчивой печати каждый элемент даты обладает достоверностью $P = 0,9$, и если, как в столбце 10 табл. 10.2, $X_1 = 0,5$, то с достаточной точностью можно задать $X_3 = X_5 = X_7 = 0,45$, $X_4 = X_6 = 0,55$, $X_8 = X_9 = 0,5(1 - 0,45)$. Ветвь ДЛВ, обусловленная событием X_2 , не может быть исследована в вероятностном аспекте, т.к., например, предъявление или непредъявление пенсионного удостоверения – свершаемые события, обладающие единичной достоверностью.

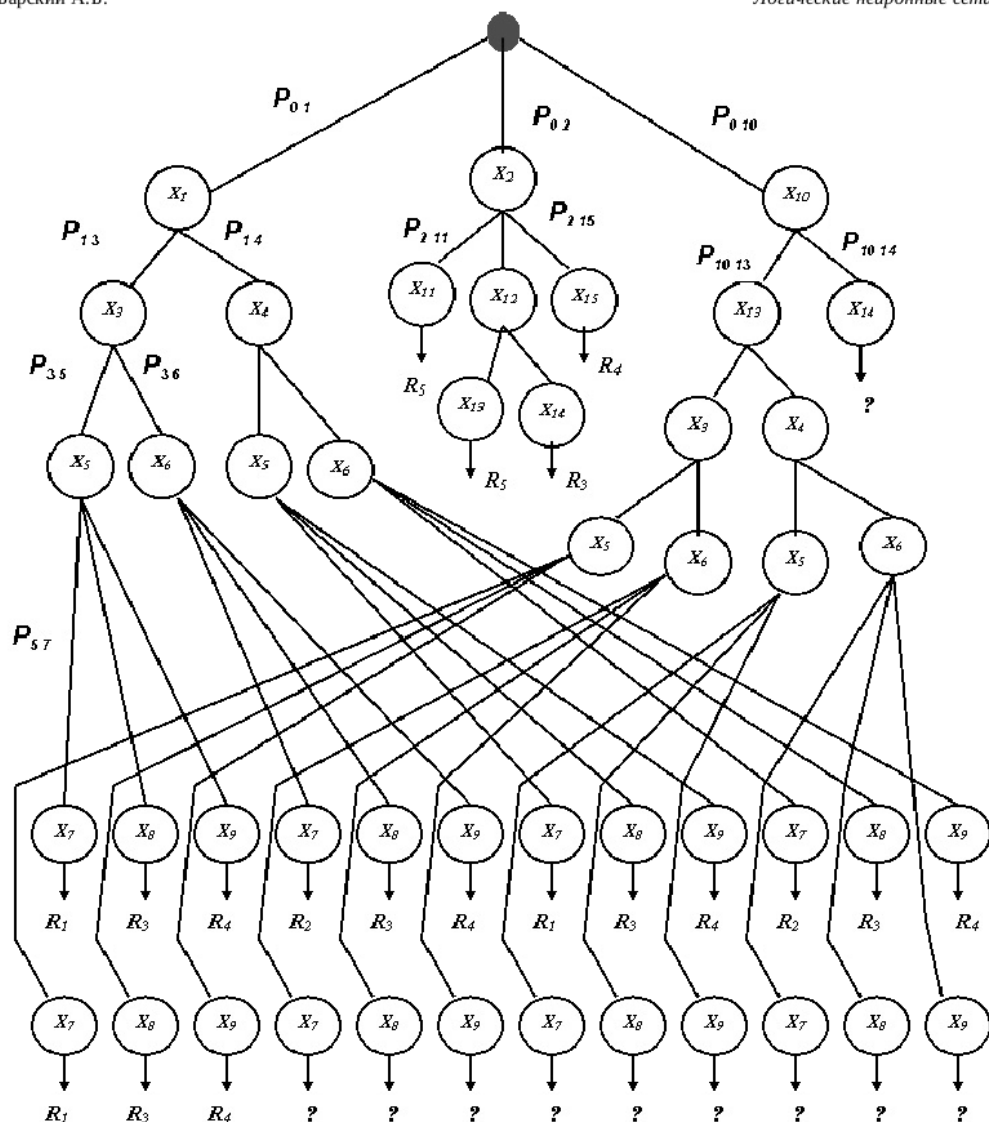


Рис. 10.5. Дерево логических возможностей

Кроме того, возникает вопрос: почему при предъявлении нестандартных ситуаций величина максимального возбуждения нейронов выходного слоя часто превышает единицу? Ведь проводилась коррекция *весов* связей этих нейронов. Следует отметить, что на вход нейросети могут подаваться ситуации в самом непредсказуемом совмещении, отличном от идеального задания эталонов. Особенность и цель построения

нейросети, как средства *ассоциативного мышления*, – возможность разобраться в наибольшей похожести. Коррекция весов связей по эталонам гарантирует лишь приблизительный интервал изменения величин возбуждения нейронов выходного слоя.

Таким образом, ДЛВ является вспомогательным средством, рекомендуемым пользователю на этапе обретения опыта, позволяющим учитывать явную зависимость событий и не допускающим абсурдных предпосылок при формировании вопросов нейросети.

Однако составление ДЛВ позволяет не только установить зависимость между достоверностью отдельных высказываний (событий), но и выявить те комбинации событий, т.е. ситуации, которые оказались неучтенными при проектировании СПР.

Пример 1. Что делать, если пассажир предъявил проездные документы, но не предъявил удостоверение работника МПС? Сформулируем вопрос нейросети, задав эталон этой ситуации, т.е. положим $X_{10} = X_{14} = 1$ и исследуем реакцию дистрибутивной нейросети на рис. 10.2 и однослойной нейросети на рис. 10.3. В обоих случаях получаем $R_1 = 0,5$, $R_2 = 0,33$, $R_3 = R_4 = 0,5$, $R_5 = 0$. Конечно, по данному эталону можно принять одно из рекомендуемых решений R_1, R_3, R_4 , тем более что по ситуации, не известной сети, мог максимально возбудиться единственный нейрон выходного слоя, как рассмотрено ниже.

Пример 2. Обнаруживаем, что при *логическом описании* СПР не учтены все комбинации дат выдачи проездных документов. Сформулируем вопрос нейросети, задав непредусмотренную эталонную ситуацию $X_{10} = X_4 = X_6 = 1$. Обе нейросети отвечают максимальным возбуждением нейрона R_2 . Рассчитывая на приблизительность, лежащую в основе *ассоциативного мышления*, можно вполне удовлетвориться полученным решением.

Однако с учетом предыдущего примера необходимо отметить, что *нейросеть* "фантазирует" и полагаться на адекватность ее выводов по непредусмотренным ситуациям не следует.

Несомненно, пробегающее по сети возбуждение при предъявлении непредусмотренной ситуации приводит к тому, что некоторый (некоторые) нейрон выходного слоя возбуждается более других, отвечая на вопрос: "на что более всего похожа данная ситуация?" Это так характерно для нашей жизни! Однако аналитический ум не удовлетворяется подобным ответом. Критический подход использует высокое значение *порога* достоверности на выходном слое.

Повысим значение *порогов передаточной функции*, реализуемой нейронами выходного слоя, положив их, например, равными 0,7. Тогда обе нейросети не будут реагировать на ситуацию в примере 1. Дистрибутивная *нейросеть* на рис. 10.2 не реагирует также на предложенную в примере 2 эталонную ситуацию, т.к. величина возбуждения нейрона R_2 составляла 0,66. Однако величина возбуждения этого нейрона в *однослойной нейросети* на рис. 10.3 составляла единицу, что требует дальнейшего повышения *порога*. Оно же может привести к ликвидации правильной реакции на предусмотренные ситуации. Ведь в столбце 9 табл. 10.2 максимальное возбуждение нейрона выходного слоя, обуславливающее адекватный вывод, составляет всего 0,95.

Если все же предъявленная ситуация "не одолела" предельно повышенный *порог* на выходном слое, ее следует считать не знакомой нейросети, не похожей ни на что, чему ее учили. Сеть требует вмешательства учителя, обучения новой ситуации.

В случае же возникновения "фантазий", не подвергающихся критическому анализу, можно предусмотреть самообучение нейросети, т.е. трассировку, закрепляющую новую причинно-следственную связь. Это справедливо в том случае, если "работа" нейросети, т.е. принимаемые ею решения, контролируются с помощью критериальной функции, оценивающей функционирование СПР. А именно: если решение, принятое в результате "фантазии", привело к удовлетворительному исходу, оно утверждается, и *нейросеть* получает разрешение на трассировку нового эталона.

Трассировка логической структуры нейросети

Предлагаются: алгоритм обучения-трассировки нейронной сети при первоначальном отсутствии синаптических связей, алгоритм оптимального закрепления событий за рецепторами и общий алгоритм трассировки нейронной сети с учётом сложной структуры логических выражений.

"Чтоб бешеной пляской землю овить,
скучную, как банка консервов,
давайте весенних бабочек ловить
сетью ненужных нервов!"

В.В. Маяковский. "Гимн здоровью"

11.1. Задачи обучения нейросети методом трассировки

Под обучением нейросети будем понимать наложение требуемой структуры (обученной) нейросети на заданную графическую структуру.

По логическому описанию СПР мы составляем логическую схему и "объявляем" ее нейросетью, назвав такой подход "схемотехническим". Однако он касался обучения обобщенным эталонам, т.е. воспроизведению потенциально однослойных нейросетей. Там требовалось обеспечить лишь прохождение импульса возбуждения от *рецептора* к нейрону выходного слоя.

Обобщим метод *трассировки*, ориентируя его на произвольно структурированное (использующее скобки) *логическое описание* СПР, в частности, – на его *дистрибутивную форму*.

Начнем с рассмотрения аналогий.

Задачу *трассировки* соединений в логической матрице для построения электронной схемы можно представить в двух вариантах.

1. На плате размещены логические элементы. Среди них заданы входные и выходные. Необходимо соединить элементы так, чтобы получить требуемое функциональное устройство. (При ориентации на технологии изготовления электронных плат, например печатных, учитывают дополнительные требования минимизации числа пересечений связей.)
2. На плате размещены логические элементы и в некоторой конфигурации введены связи между ними. Топология связей, при выделении входов и выходов, может быть как случайной, так и регулярной. Необходимо подтвердить существование некоторых связей и ликвидировать другие так, чтобы сформировать требуемое функциональное устройство. При этом может допускаться введение минимального количества недостающих связей.

Как правило, при таких постановках задачи *трассировки* не рассматриваются обратные связи.

Исходя из аналогий, обусловленных общей логической базой, формулируем и решим две задачи построения обученной (*обучения*) *нейронной сети* в соответствии с ее специализацией в составе СПР.

Используя предложенную нам для обучения нейросеть, позволим ее модификацию – введение недостающих связей (отвергнуть негодную сеть – значит, рассчитывать на случай), если очевидно, что без таких связей правильного функционирования добиться невозможно.

11.2. Формирование обученной нейросети на основе заданного множества нейроподобных элементов

Под *нейроподобным элементом*, который мы условились называть нейроном, подразумевается не столько овеществленный образ – электронный аналог, программный модуль и др., сколько реализуемая им передаточная функция.

Пусть задано множество нейронов $X = \{x_1, \dots, x_m\}$ и *логическое описание* СПР в *дистрибутивной форме* ¹⁾ По "логической" технологии, обоснованной ранее, необходимо сформировать связи

между нейронами так, чтобы при единичных значениях достоверности, совпадающих с единичными булевыми значениями переменных, максимального возбуждения на выходном слое при данной передаточной функции достигал тот нейрон, который закреплен за решением, обусловленным единичным значением соответствующей ему логической функции.

Скобочная запись задает последовательность формирования обученной нейросети. Процесс такого формирования (обучения) заключается в рекурсивной коммутации подструктур – термов на основе содержимого скобок в соответствии с их вложенностью.

Продолжим рассмотрение примера на основе логического описания (10.1). Перепишем его, используя символ обобщенной операции, реализуемой передаточной функцией, но сохранив скобки:

$$\begin{aligned}
 & ((X_1 \& X_7) \& (X_3 \& X_5)) \& (X_{10} \& X_{13}) \rightarrow R_1, \\
 & ((X_1 \& X_7) \& (X_4 \& X_6)) \& (X_2 \& X_{12} \& X_{14}) \rightarrow R_2, \\
 & X_8 \& (X_1 \& X_{10}) \rightarrow R_3, \\
 & (X_9 \& (X_1 \& X_{10})) \& (X_2 \& X_{15}) \rightarrow R_4, \\
 & X_2 \& (X_{11} \& X_{13}) \rightarrow R_5.
 \end{aligned}
 \tag{11.1}$$

Прежде всего из множества X необходимо выделить нейроны входного – рецепторного – и выходного слоев. Исходя из этого, а также, задав количество необходимых и предполагаемых нейронов разного функционального назначения, закрепим и переобозначим нейроны: $X = \{X_1, \dots, X_{15}, 1, \dots, 11, R_1, \dots, R_5\}$. Нейроны $1, \dots, 11$ в будущих построениях исполняют роль нейронов скрытых слоев – скрытых нейронов.

Составим *матрицу следования*, первоначально содержащую "пустые" элементы. На рис. 11.1 представлена такая матрица S , соответствующая уже полученному решению. Однако пока не следует учитывать единичные элементы: их необходимо сформировать в процессе решения задачи.

Для удобства пользования и учитывая, что данная матрица при отсутствии обратных связей – треугольная, она выполнена в виде

"шахматки", где диагональные элементы отмечены.

Итак, чтобы сигналы возбуждения направленно проходили от входных нейронов к выходным, необходимо сформировать связи с единичными весами, используя скрытые нейроны. Пусть скрытые нейроны образуют ресурс – множество M свободных нейронов, первоначально $M = \{1, \dots, 11\}$.

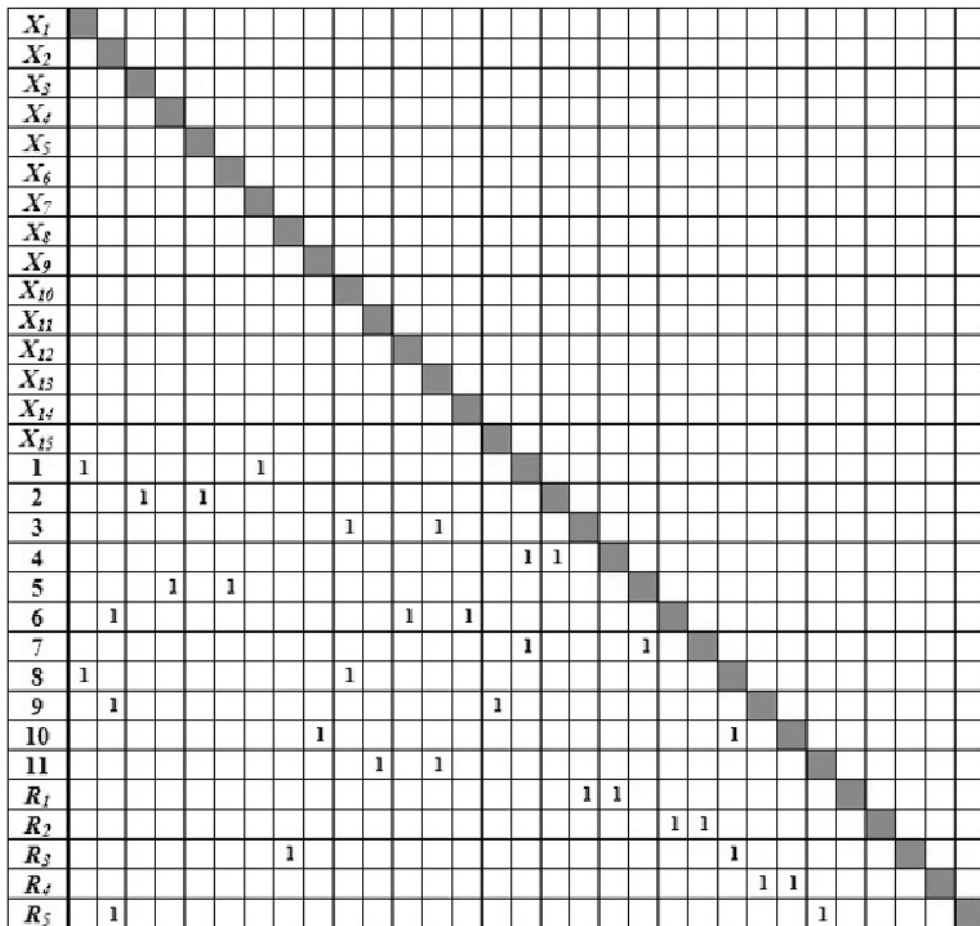


Рис. 11.1. Матрица следования после обучения нейросети

Организуем последовательный анализ выражений (11.1).

При просмотре первого выражения, анализируя открывающие и закрывающие скобки, найдем первую скобку самого высокого уровня

вложенности. Она соответствует выражению $(X_1 \& X_7)$. Закрепим за ним первый свободный нейрон 1 из M , предположив тем самым, что он выполняет аналог операции, указанной в скобках. Сформируем уточненную строку *матрицы следования* S , соответствующую нейрону 1, записав веса связей, равные единице, в столбцы, соответствующие нейронам X_1 и X_7 .

Обозначим исследованную "скобку" именем нейрона 1 и перепишем логическое выражение:

$$(1 \& (X_3 \& X_5)) \& (X_{10} \& X_{13}) \rightarrow R_1.$$

Продолжим поиск "скобок" самого высокого уровня вложенности.

Находим "скобку" $(X_3 \& X_5)$. В общем случае такая "скобка" могла встретиться ранее, и за ней мог быть закреплен скрытый нейрон. В целях экономии нейронов сформируем уточненную строку *матрицы следования*, записав единицы в столбцах, соответствующих нейронам X_3 и X_5 . Однако, прежде чем поместить строку на ее место, проверим, нет ли в S идентичной строки? В данном случае такой строки нет. Тогда соотнесем полученную строку с очередным свободным нейроном 2 и поместим ее в S .

Перепишем логическое выражение, заменив обработанную "скобку":

$$(1 \& 2) \& (X_{10} \& X_{13}) \rightarrow R_1.$$

Продолжая так же, закрепим нейрон 3 за "скобкой" $(X_{10} \& X_{13})$ и сформируем соответствующую этому нейрону строку матрицы S . Запись логического выражения примет вид

$$(1 \& 2) \& 3 \rightarrow R_1.$$

Следующий обзор этой записи приводит к замене "скобки" $(1 \& 2)$ именем нейрона 4 и к формированию строки *матрицы следования*, соответствующей этому нейрону. Эта строка содержит единицы в строках, соответствующих нейронам 1 и 2. Запись логического выражения примет вид

$$4 \ \& \ 3 \ \rightarrow \ R_1 .$$

Отсутствие скобок говорит о необходимости введения связей $4 \rightarrow R_1$ и $3 \rightarrow R_1$. Таким образом, строка, соответствующая нейрону R_1 выходного слоя, содержит единицы в столбцах, соответствующих нейронам 3 и 4.

Трассировка решения R_1 закончена.

Анализируем логическое выражение

$$((X_1 \ \& \ X_7) \ \& \ (X_4 \ \& \ X_6)) \ \& \ (X_2 \ \& \ X_{12} \ \& \ X_{14}) \ \rightarrow \ R_2 .$$

Строка матрицы S , сформированная по "скобке" $(X_1 \ \& \ X_7)$, совпадает со строкой, соответствующей нейрону 1. Значит, можно "сэкономить", повторно используя этот нейрон. Преобразуем запись:

$$(1 \ \& \ (X_4 \ \& \ X_6)) \ \& \ (X_2 \ \& \ X_{12} \ \& \ X_{14}) \ \rightarrow \ R_2 .$$

За следующей "скобкой" $(X_4 \ \& \ X_6)$ закрепим нейрон 5, сформировав для него строку матрицы S , и запись примет вид

$$(1 \ \& \ 5) \ \& \ (X_2 \ \& \ X_{12} \ \& \ X_{14}) \ \rightarrow \ R_2 .$$

Продолжив ее обзор, получим новый вид записи $(1 \ \& \ 5) \ \& \ 6 \rightarrow R_2$, а затем $7 \ \& \ 6 \rightarrow R_2$. Сформированные при этом строки матрицы S показаны на рисунке. Аналогично анализируются другие логические выражения (11.1), что не вносит новых особенностей в неформально изложенный алгоритм.

Нейросеть, представленная матрицей следования S , близка к нейросети, показанной на [рис. 10.2](#).

11.3. Оптимальное закрепление рецепторов за событиями в нейросети заданной структуры

Если задана структура нейросети (подразумевается ее "овеществленная"

проекция на плоскость или даже реализация на плате, на кристалле, на решетке связанных процессорных элементов и т.д.), то от распределения рецепторов между событиями зависит возможность формирования отдельных термов без введения дополнительных связей. Например, если достоверность событий X_1 и X_7 подается на входы, значительно отстающие друг от друга, то при заданной, например, регулярной структуре связей могут возникнуть большие трудности в реализации аналога конъюнкции $X_1 \wedge X_7$, используемой далее в качестве отдельного терма, который входит в состав других логических конструкций. Необходимое объединение этих событий на входе одного нейрона может быть достигнуто даже не ранее, чем на выходном слое. Это потребует введения дополнительных связей для построения необходимого терма.

Чтобы минимизировать число вводимых дополнительных связей или исключить их вовсе, необходимо на основе анализа конструкций, входящих в состав описания вида (11.1), построить предпочтительный ряд следования событий, реализующий компромисс в выборе минимальных расстояний между элементами этого ряда.

Пусть на [рис. 11.2](#) представлена структура нейросети, которую следует обучить в соответствии с логическим описанием (11.1) создаваемой СПР. На рисунке отображено окончательное решение задачи трассировки, и связи с единичными весами выделены. Произведем обоснование этого решения и, прежде всего, распределим события между нейронами-рецепторами.

Воспользуемся следующим алгоритмом упорядочения событий.

1. Организуем сквозной просмотр всех выражений (11.1) и выпишем последовательность имен событий в порядке их обнаружения.
2. Просматриваем слева направо сформированную последовательность и находим имя события, входящего в эту последовательность неоднократно.
3. Находим среднее значение позиций, занимаемых именем найденного события в последовательности, и определяем для него ближайшее целое, не меньшее найденного.
4. Найденное значение определяет новый порядковый номер данного события в последовательности. Заносим имя события на

новую позицию и исключаем другие его вхождения в последовательность.

- Шаги 2-4 повторяем до достижения единственности вхождения всех имен событий в последовательность.

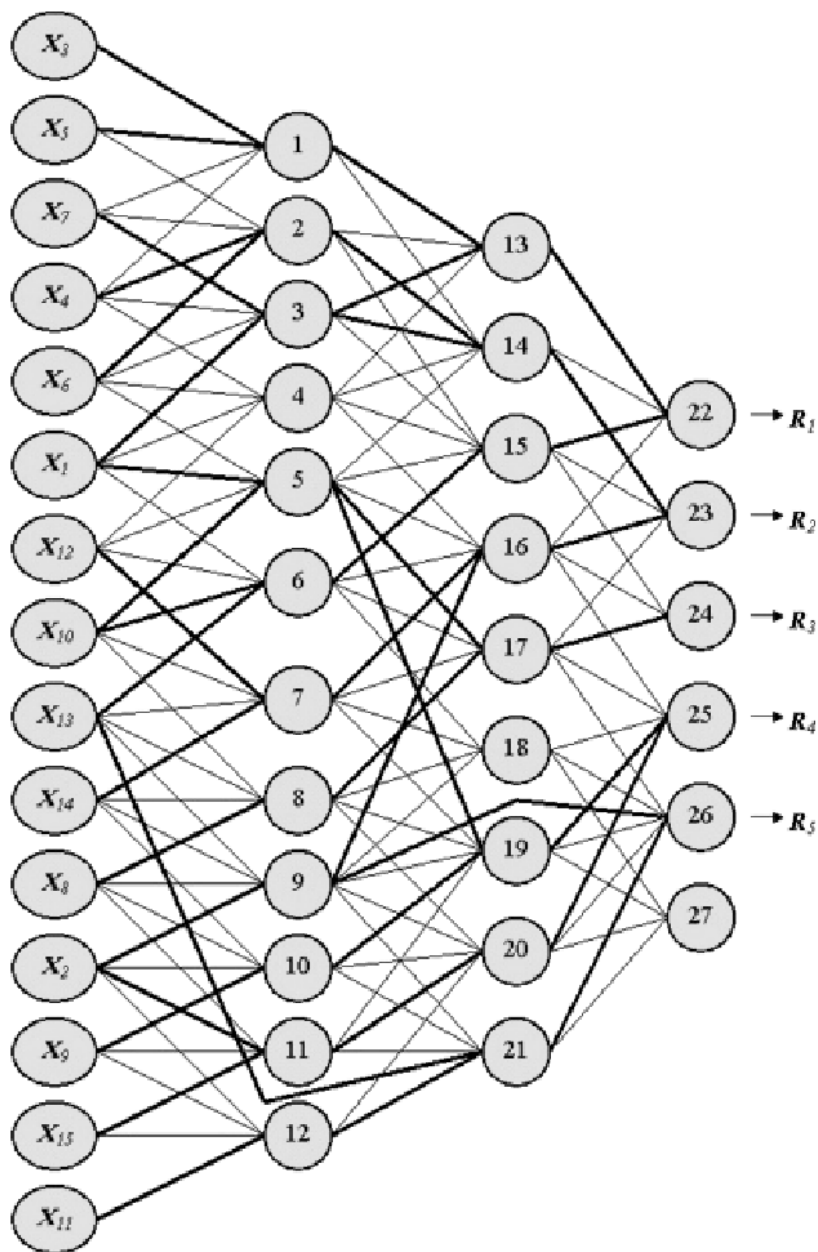


Рис. 11.2. Обученная нейросеть

В рассматриваемом примере последовательность имен событий первоначально имеет вид

|-----↓
 $X_1 X_7 X_3 X_5 X_{10} X_{13} X_1 X_7 X_4 X_6 X_2 X_{12} X_{14} X_8 X_1 X_{10} X_9 X_1 X_{10} X_2 X_{15} X_2 X_{11} X_{13}$

Имя X_1 входит неоднократно. Новая его позиция определяется как $\lfloor (1 + 7 + 15 + 18) : 4 \rfloor = 11$. После переноса X_1 на новое место (показано стрелкой) и исключения его прежних вхождений последовательность имен принимает вид

|-----↓
 $X_7 X_3 X_5 X_{10} X_{13} X_7 X_4 X_6 X_1 X_2 X_{12} X_{14} X_8 X_{10} X_9 X_{10} X_2 X_{15} X_2 X_{11} X_{13}$

Новое место X_7 определяется как $\lfloor (1 + 6) : 2 \rfloor = 4$. Последовательность принимает вид

|-----↓
 $X_3 X_5 X_7 X_{10} X_{13} X_4 X_6 X_1 X_2 X_{12} X_{14} X_8 X_{10} X_9 X_{10} X_2 X_{15} X_2 X_{11} X_{13}$

Новое место X_{10} определяется как $\lfloor (3 + 13 + 15) : 3 \rfloor = 11$. Последовательность принимает вид

|-----↓
 $X_3 X_5 X_7 X_{13} X_4 X_6 X_1 X_2 X_{12} X_{10} X_{14} X_8 X_9 X_2 X_{15} X_2 X_{11} X_{13}$

Новая позиция X_{13} определяется как $\lfloor (4 + 18) : 2 \rfloor = 11$. Получаем новый вид последовательности:

|-----↓
 $X_3 X_5 X_7 X_4 X_6 X_1 X_2 X_{12} X_{10} X_{13} X_{14} X_8 X_9 X_2 X_{15} X_2 X_{11}$

Отыскивается новое место X_2 , как $\lfloor (7 + 14 + 16) : 3 \rfloor = 13$, и последовательность имен событий принимает окончательный вид, отображенный на рисунке:

$$X_3 X_5 X_7 X_4 X_6 X_1 X_{12} X_{10} X_{13} X_{14} X_8 X_2 X_9 X_{15} X_{11}.$$

Заметим, что при реализации на компьютере "кнопки" или окна на панели управления могут располагаться в удобном для пользователя порядке, соединяясь с закрепленными за ними *рецепторами*.

11.4. Обучение-трассировка нейросети заданной структуры

Как видно из примера, рассмотренного в лекции 10, если нейросеть не может быть сведена к однослойной на заданной графической структуре, то при *трассировке* необходимо следовать структуре логического выражения. Это требует такой структуризации обучаемой нейросети, при которой "скобки" в порядке их вложенности соответствуют выделенным подструктурам сети. Тогда подструктуры нейросети образуют расширяемый ресурс, предоставляемый для реализации "скобок", – от максимального уровня вложенности до нулевого.

В рассматриваемой сети (рис. 11.2) максимальная длина статического пути возбуждения равна трем. Более того, данная нейросеть является трехслойной в "традиционном" смысле [13, 15, 19, 20], т.е. в ней нет связей "через слой".

Запишем первое логическое выражение из описания (11.1):

$$((X_1 \& X_7) \& (X_3 \& X_5)) \& (X_{10} \& X_{13}) \rightarrow R_1.$$

Ему соответствует максимальная длина логических цепочек, равная трем, что следует из порядка вложенности скобок. Тогда справедливо предположение о том, что "скобки" $(X_1 \& X_7)$ и $(X_3 \& X_5)$ могут быть реализованы не далее чем на первом слое ($3 - 2 = 1$), чтобы не воспрепятствовать формированию подструктур, в которые они входят. Охватывающая их "скобка" $((X_1 \& X_7) \& (X_3 \& X_5))$ должна быть реализована на втором слое. Не далее чем на этом же, т.е. на невыходном, слое может быть реализована и скобка $(X_{10} \& X_{13})$.

Предположим, что сквозная нумерация позволяет выделить подмножества нейронов, принадлежащих одному слою или, в общем

случае, имеющих одинаковую максимальную длину *статической цепочки*, как показано на рис. 11.2. Действительно, нейроны 1-12 образуют первый слой нейросети, нейроны 13-21 – второй слой, третий слой – выходной. Тогда и в описывающей нейросеть *матрице следования* S , наличие которой лишь подразумевается ввиду ее громоздкости, можно отчеркнуть строки и столбцы нейронов, принадлежащих, в данном случае, одному слою. (Напомним, что в матрице S связи обозначаются значениями весов, первоначально равными нулю.)

Построим матрицу S_1 (рис. 11.3) на основании выражения, определяющего решение R_1 .

Для ее построения необходимо из матрицы S исключить строки (и столбцы), которые соответствуют переменным, не участвующим в построении решения R_1 , а также – представительство других решений на выходном слое.

Приступим к реализации первой "скобки" третьего уровня вложенности.

В столбцах, соответствующих событиям X_7 и X_{11} , ведя поиск в пределах нейронов первого слоя, найдем первую строку, которая содержит максимальное число нулей, указывающих наличие связей. Такая строка, где нули находятся во всех выделенных столбцах, соответствует нейрону 3. В матрицах S и S_1 меняем эти нули на единицы – формируем единичные веса связей. В матрице S при строке, соответствующей нейрону 3, указываем число введенных единиц, $m_3 = 2$. В записи выражения меняем "скобку" на имя нейрона, реализующего ее. Запись принимает вид

$$(3 \& (X_3 \& X_5)) \& (X_{10} \& X_{13}) \rightarrow R_1.$$

Перейдем к реализации "скобки" $(X_3 \& X_5)$ того же уровня вложенности.

преобразуется:

$$(3 \& 1) \& 6 \rightarrow R_1.$$

Анализируя столбцы, соответствующие нейронам 1 и 3, находим нейрон 13 второго слоя, реализующий скобку $(3 \& 1)$, отмечаем значение $m_{13} = 2$. Запись выражения преобразуется к виду

$$13 \& 6 \rightarrow R_1.$$

Анализируя столбцы, соответствующие нейронам 6 и 13, включив в рассмотрение нейроны выходного слоя, находим первую строку, содержащую максимальное число нулей. Такая строка соответствует нейрону 15. Однако она содержит нуль лишь в столбце нейрона 6. Меняем этот нуль в матрицах S и S_1 на единицу, полагаем $m_{15} = 1$. Запись выражения преобразуется к виду

$$13 \& 15 \rightarrow R_1.$$

Затем реализуем единственную "скобку" нулевого уровня вложенности с помощью нейрона R_1 . Трассировка первого решения закончена. В результате ее введены отмеченные на рисунке связи $X_3 \rightarrow 1$, $X_5 \rightarrow 1$, $X_7 \rightarrow 3$, $X_1 \rightarrow 3$, $X_{10} \rightarrow 6$, $X_{13} \rightarrow 6$, $1 \rightarrow 13$, $3 \rightarrow 13$, $6 \rightarrow 15$, $13 \rightarrow R_1$, $15 \rightarrow R_1$.

Для трассировки решения

$$((X_1 \& X_7) \& (X_4 \& X_6)) \& (X_2 \& X_{12} \& X_{14}) \rightarrow R_2$$

построим матрицу следования S_2 (рис. 11.4). Для этого исключим из матрицы S все нейроны-рецепторы, не участвующие в приведенной выше записи выражения, а также неиспользуемые нейроны выходного слоя. Затем с помощью одного просмотра сверху вниз исключим представительство тех нейронов, где число единиц в строках меньше соответствующего значения m .

При анализе "скобки" $(X_1 \& X_7)$ обнаруживаем в столбцах,

соответствующих нейронам X_1 и X_7 , две единицы в строке, соответствующей нейрону 3, при том, что $m_3 = 2$. Запись выражения преобразуется к виду

$$(3 \ \& \ (X_4 \ \& \ X_6)) \ \& \ (X_2 \ \& \ X_{12} \ \& \ X_{14}) \ \rightarrow \ R_2.$$

В строке, соответствующей X_4 и X_6 , нет строки, содержащей единицы во всех позициях. Находим первую строку, соответствующую нейрону первого слоя и содержащую максимальное число нулей при нулевом значении m . Такая строка, содержащая нули во всех позициях, соответствует нейрону 2. В матрицах S и S_2 меняем выделенные нули на единицы, фиксируем значение $m_2 = 2$. Запись выражения преобразуется:

$$(3 \ \& \ 2) \ \& \ (X_2 \ \& \ X_{12} \ \& \ X_{14}) \ \rightarrow \ R_2.$$

В соответствии с уровнем вложенности присутствующие в записи "скобки" могут быть реализованы как на первом, так и на втором слое нейросети. В столбцах, соответствующих X_2 , X_{12} , X_{14} , нет строки нейрона первого или второго слоя, содержащей единицы во всех столбцах. Находим первую строку, содержащую максимальное число нулей при нулевом значении m . Такая строка соответствует нейрону 7. Меняем выделенные нули на единицы, фиксируем значение $m_7 = 2$. Запись выражения принимает вид

$$(3 \ \& \ 2) \ \& \ (X_2 \ \& \ 7) \ \rightarrow \ R_2. \quad (11.2)$$

Преобразование первой скобки производится с помощью нейрона 14 аналогично предыдущему. Фиксируем значение $m_{14} = 2$ и формируем новую запись:

$$14 \ \& \ (X_2 \ \& \ 7) \ \rightarrow \ R_2.$$

Выделяем столбцы, соответствующие нейронам X_2 и 7, и так как в этих столбцах нет строки, соответствующей нейронам второго слоя и содержащей единицы во всех позициях (нейрон 7 принадлежит первому слою и может участвовать в термах, создаваемых на втором

слое), находим первую строку, соответствующую нейрону второго слоя и содержащую максимальное число нулей при нулевом значении m . Такая строка соответствует нейрону 16, который становится преемником лишь нейрона 7. Меняем выделенный нуль на единицу, фиксируем значение $m_{16} = 1$. Запись выражения принимает вид

$$14 \ \& \ (X_2 \ \& \ 16) \ \rightarrow \ R_2.$$

Теперь выделяем столбцы нейронов X_2 и 16. Находим в них строку с максимальным числом нулей при нулевом значении m не далее, чем на втором слое нейронов. Такая строка соответствует нейрону 9. Меняем выделенный нуль на единицу, полагаем $m_9 = 1$. Запись принимает вид

$$14 \ \& \ (9 \ \& \ 16) \ \rightarrow \ R_2.$$

Выделяем столбцы, соответствующие нейронам 9 и 16. В них находим строку с максимальным числом нулей и при нулевом m не далее чем на втором слое нейронов. Такая строка соответствует нейрону 18.

Однако прежде чем использовать этот нейрон, заметим, что здесь мы впервые сталкиваемся с необходимостью введения дополнительной связи. Ведь нейрон 16 принадлежит "предельному" слою для реализованного им терма. Его объединение с нейроном 18 возможно лишь на слое более высокого уровня, что не позволяет реализовать все выражение. Нейрон 9 принадлежит первому слою, т.е. к предыдущему по отношению к нейрону 16. Его объединение с нейроном 16 с помощью непосредственной связи $9 \rightarrow 16$ обеспечивает правильную реализацию "скобки".

Таким образом, сформулируем правило: если "скобка" объединяет нейроны, среди которых один принадлежит слою максимального уровня, допустимого для формирования терма, а другие нейроны, образующие эту же "скобку", принадлежат предыдущему слою, то необходимо ввести дополнительные связи, соединяющие эти нейроны с первым, если такие связи не были заданы, и присвоить им единичный вес.

весов связей $X_1 \rightarrow 5, X_{10} \rightarrow 5, X_8 \rightarrow 8, 5 \rightarrow 17, 8 \rightarrow 17, 17 \rightarrow R_3$.

Произведем трассировку решения R_4 , записав определяющее его выражение

$$(X_9 \ \& \ (X_1 \ \& \ X_{10} \)) \ \& \ (X_2 \ \& \ X_{15} \) \ \rightarrow \ R_4 \ .$$

Матрица S_4 представлена на рис. 11.6.

X_1																	
X_{10}																	
X_8																	
4	0																
5	1	1															
8		0	1														
10			0														
11			0														
12																	
17					1	1											
18						0											
19						0	0	0									
20						0	0	0	0								
21							0	0	0								
R_3											1						

Рис. 11.5. Матрица следования S_3 после трассировки решения R_3

Выделим столбцы, соответствующие нейронам X_1 и X_{10} , и находим в них строку, соответствующую нейрону 5, в которой единицы записаны во всех позициях, $m_5 = 2$. Заменим "скобку" именем нейрона 5:

$$(X_9 \ \& \ 5) \ \& \ (X_2 \ \& \ X_{15} \) \ \rightarrow \ R_4 \ .$$

"Скобку" $(X_2 \ \& \ X_{15} \)$ реализуем с помощью нейрона 11, и запись принимает вид

$$(X_9 \ \&5) \ \&11 \rightarrow R_4.$$

Анализ первой скобки приводит к единичному весу связи $X_9 \rightarrow 10$, а затем $10 \rightarrow 19$. Запись выражения принимает вид

$$(19\&5) \ \&11 \rightarrow R_4.$$

Дальнейший анализ "скобки" свидетельствует о том, что нейрон 19 принадлежит слою максимального уровня, на котором эта "скобка" может

быть реализована. Нейрон 5 принадлежит предыдущему слою. Тогда по правилу, изложенному выше, вводим дополнительную связь $5 \rightarrow 19$. Полагаем $m_{19} = 2$. Запись выражения принимает вид

$$19\&11 \rightarrow R_4.$$

Полученная "скобка" может быть реализована на выходном слое нейронов. Анализируем столбцы, соответствующие нейронам 11 и 19, находим в них первую строку с максимальным числом нулей при нулевом значении m , соответствующую нейрону 20, и вводим единичный вес связи $11 \rightarrow 20$. Запись выражения принимает вид

$$19\&20 \rightarrow R_4.$$

Однако анализ столбцов, соответствующих нейронам 19 и 20, свидетельствует об отсутствии связей этих нейронов с нейроном R_4 . Введем дополнительные связи $19 \rightarrow R_4$ и $20 \rightarrow R_4$.

Трассировка решения R_4 закончена.

X_7																			
X_{10}																			
X_2																			
X_9																			
X_{15}																			
4	0																		
5	1	1																	
9			1																
10			0	1															
11			1	0	1														
12			0	0	0														
18								0											
19								1	0	1	0								
20								0	0	1	0								
21								0	0	0	0								
R_4														0	1	1			

Рис. 11.6. Матрица следования S_4 после трассировки решения R_4

Для трассировки решения R_5 составим матрицу S_5 (рис. 11.7) и запишем в первоначальном виде логическое выражение

$$X_2 \& (X_{11} \& X_{13}) \rightarrow R_5.$$

Выделяем столбцы, соответствующие нейронам X_{11} и X_{13} . В них нет строки, соответствующей нейрону не более чем второго слоя и содержащей единицы. Находим первую строку, содержащую максимальное число нулей при нулевом значении m . Эта строка соответствует нейрону 12. Меняем нуль на единицу, положив $m_{12} = 1$; запись выражения принимает вид

$$X_2 \& (12 \& X_{13}) \rightarrow R_5.$$

Продолжаем анализ "скобки", в результате чего запись выражения принимает вид

$$X_2 \& (21 \& X_{13}) \rightarrow R_5.$$

Вновь выделяем столбцы, соответствующие нейронам 21 и X_{13} и не обнаруживаем в них, в пределах второго слоя, строк, содержащих единицы или нули. Так как нейрон 21 принадлежит предельному для формируемого

терма слою, вводим дополнительную связь $X_{13} \rightarrow 21$. Запись выражения принимает вид

$$X_2 \& 21 \rightarrow R_5.$$

Дальнейшие преобразования приводят к использованию нейрона 9 и к введению дополнительных связей $9 \rightarrow R_5$ и $21 \rightarrow R_5$.

X_{13}	■								
X_2		■							
X_{11}			■						
4				■					
9	0	1			■				
12		0	1			■			
18					0		■		
21	1				0	1		■	
R_5							0	1	■

Рис. 11.7. Матрица следования S_5 после трассировки решения R_5

Трассировка нейросети закончена.

11.5. Алгоритм трассировки нейросети по описанию системы принятия решений в дистрибутивной форме

Рассмотренный выше пример иллюстрирует практически все ситуации, возникающие при обучении предварительно заданной нейросети. Они полностью определяют общий алгоритм ее трассировки по структурированной записи логических выражений СПР. Однако при формальном описании такого алгоритма следует предусмотреть некоторые вспомогательные преобразования матриц следования, упрощающие компьютерную реализацию. Их целесообразность

продемонстрирована в ранее рассмотренном в лекции 4 алгоритме *трассировки* по обобщенному эталону.

Следует напомнить предпосылки настоящего алгоритма, отличающие его от упомянутого.

1. *Логическое описание* СПР состоит из структурированных выражений, использующих скобки. Создаваемые нейросетевые подструктуры, объединенные связями с единичными весами, должны отображать порядок вложенности этих скобок. Это усложняет алгоритм *трассировки* вводом ограничений на длину статической цепочки возбуждения или на предельный слой нейронов при формировании подструктур нейросети – "скобок" в логической записи.
2. *Дистрибутивная форма* записи логических выражений обеспечивает единственность вхождения переменных, что значительно упрощает создание подструктур нейросети.

В остальном изложенный ниже алгоритм основан на схожих действиях, что и упомянутый.

Алгоритм.

1. Подготовительные действия.
 - Используя сквозную нумерацию нейронов, разобьем весь диапазон номеров скрытых нейронов на множества по их принадлежности одному слою, если заданная сеть "слоистая", или по одинаковым максимальным длинам *статических цепочек* возбуждения (что для удобства также условно отнесем к одному слою). Таким образом, с учетом нейронов выходного слоя, будет установлен ранг r (в частности, число слоев) нейросети.
 - Найдем максимальное значение s уровня вложенности скобочных выражений ("скобок") в дистрибутивном описании СПР.

Заданную для *трассировки* -обучения нейросеть следует считать корректной, если выполняется соотношение $s \leq r$.

2. Строим *матрицу следования* S , отображающую заданную нейросеть. Элемент этой матрицы на пересечении строки α и столбца β пуст, если в заданной нейросети отсутствует синапсическая связь $\beta \rightarrow \alpha$. Если такая связь существует, этот элемент имеет значение 0 – первоначально заданного *веса синапсической связи*. (В процессе *трассировки* некоторые веса приобретают единичное значение.) Каждую строку матрицы S , соответствующую *скрытому нейрону*, сопроводим признаком m_α , который указывает число единиц, введенных в процессе *трассировки*.
3. Организуем перебор записей всех выражений $i = 1, \dots, P$ из описания вида (11.1). Для каждой записи выполняем пп. 4-15. Если запись исчерпана, *трассировка* i -го решения заканчивается. По исчерпанию записей всех выражений *трассировка* нейросети заканчивается.
4. Фиксируем первоначальную запись i -го выражения и строим *матрицу следования* S_i исключением из матрицы S строк и столбцов, которые соответствуют переменным, не входящим в запись, а также всем решениям, кроме R_i .
5. В матрице S_i последовательно, сверху вниз, вычеркиваем строки (и соответствующие столбцы) которые содержат число ранее введенных единиц, меньшее значения m , указанного при строке.
6. Организуем просмотр выделенной записи и находим очередное внутрискобочное выражение ("скобку") самого высокого уровня вложенности. Если "скобок" больше нет, *трассировка* решения закончена, выполняем п. 3.
7. Ставим в соответствие уровню s^* вложенности скобки множество нейронов слоя $r = s^* + 1$, который является предельным для формирования терма – нейрона, реализующего (концентрирующего в себе возбуждение) данную "скобку".
8. Выделяем множество столбцов матрицы S_i , которые соответствуют нейронам, образующим "скобку".
9. В совокупности выделенных столбцов находим строку, во всех позициях которой стоят единицы. Если такая строка найдена, переходим к выполнению п. 15.
10. В совокупности выделенных столбцов находим первую строку, в пределах, допустимых для данной "скобки", содержащую

максимальное число нулей.

11. Если нейрон, соответствующий найденной строке, принадлежит предельному для данной "скобки" слою, записываем единицы (возможно, введя дополнительные связи с единичными весами) во все позиции, соответствующие выделенным столбцам. Переходим к выполнению п. 14.
12. Если этот нейрон не принадлежит предельному слою, меняем найденные нули на единицы (присваиваем связям единичный вес). Переходим к выполнению п. 14.
13. Если в выделенных столбцах не найдена строка с единицами и нулями, находим первый из представленных в S_i нейрон предельного слоя и по всем позициям выделенных столбцов записываем единицы. То есть вводим дополнительные связи с единичными весами.
14. Отображаем введенные изменения в матрице S . Количество введенных единиц указываем в признаке m модифицированной строки, если она не соответствует нейрону выходного слоя.
15. В записи исследуемого выражения меняем использованные нейроны (или всю "скобку"), передавшие возбуждение нейрону со скорректированными связями, на имя этого нейрона. Из матрицы S_i исключаем соответствующие этим нейронам, отображенным в "скобке", строки и столбцы. Переходим к выполнению п. 6.

Описание алгоритма закончено.

В заключение необходимо подчеркнуть, что корректное представление системы принятия решений на этапе ее логического описания определяет правильную "работу" нейронной сети. На этом этапе должно быть предложено простейшее описание, исключающее противоречивость и обеспечивающее эффективную нейросетевую реализацию.

В рассматриваемом в предыдущих лекциях примере допущена небрежность при формальном описании системы в терминах математической логики. Она привела к неоднозначности выдаваемого решения при задании ситуаций на основе булевых переменных. Ошибка, скрытая в логической схеме системы (можно считать, – умышленная), с неизбежностью сказалась на правильности принятия решений. Это еще раз свидетельствует о прямой связи логического и

нейросетевого представлений, о том, что *логическое описание* полностью определяет как структуру, так и правильное функционирование нейросети.

Однако отметим, что нейросеть на основе обобщения типа используемых данных (преобразования булево в вещественное) представляет собой и обобщение логической схемы, что приближает ее к воплощению "природных" механизмов ассоциативного мышления.

Моделирование ассоциативного мышления фактически эквивалентно созданию таблиц, фиксирующих опыт вида "если . . . , то . . . ", т.е. связывающих возникающие ситуации с принимаемым по ним решениям. Распространение возбуждения в нейросети с помощью передаточной функции является эффективным механизмом и оригинальным методом интерполяции (экстраполяции) опыта, т.е. нахождения по максимальной величине возбуждения наиболее близкого решения либо определения математического ожидания исхода с помощью значений возбуждения нескольких нейронов выходного слоя.

Благодаря логическому методу построения математический аппарат нейронных сетей прост и доходчив, разработка нейросети недорога, она не требует специальных знаний пользователя. В то же время нейросеть легко допускает включение в рассмотрение новых факторов, изменение решений и т.д.

Тот факт, что двухслойные сети достаточны для построения систем принятия решений, а также возможно широкое применение однослойных сетей, может быть учтен при разработке или программировании универсального нейрокомпьютера.

В представленном здесь алгоритме *трассировки* нейросети с заданными связями при дистрибутивном описании единичные веса вводятся в строгом соответствии с логической схемой СПР. В этом отличие данного алгоритма от алгоритма, изложенного в лекции 4. При этом используемым ресурсом являются все *скрытые нейроны*, независимо от структуры транзитивных связей.

В то же время построение обученной нейросети по дистрибутивному описанию СПР в том случае, когда связи между нейронами предварительно не установлены, представляется более перспективным,

т.к. "исправлять" уже введенные связи гораздо труднее, чем вводить их заново.

Анализ примеров позволяет с уверенностью говорить о высокой актуальности СПР, реализуемых с помощью однослойных и совершенных нейросетей, что значительно облегчает решение задачи *трассировки* -обучения.

- 1) Напоминаем, что здесь не рассматриваются обратные связи

Методика построения системы принятия решений на основе логической нейронной сети

Выводы предыдущих лекций объединяются в общую методику построения системы принятия решений, в основе которой лежит логическая нейронная сеть. Освещаются все этапы построения такой сети.

"Поразмыслив аккуратно,

Я избрал себе дорожку

И иду себе без шума,

Понемножку, понемножку!"

А.К. Толстой. "Благоразумие"

12.1. Основные этапы работ

На рис. 12.1 представлены основные этапы построения СПР.

Ниже подробно рассмотрены эти этапы на основе обобщения всего предшествующего материала.

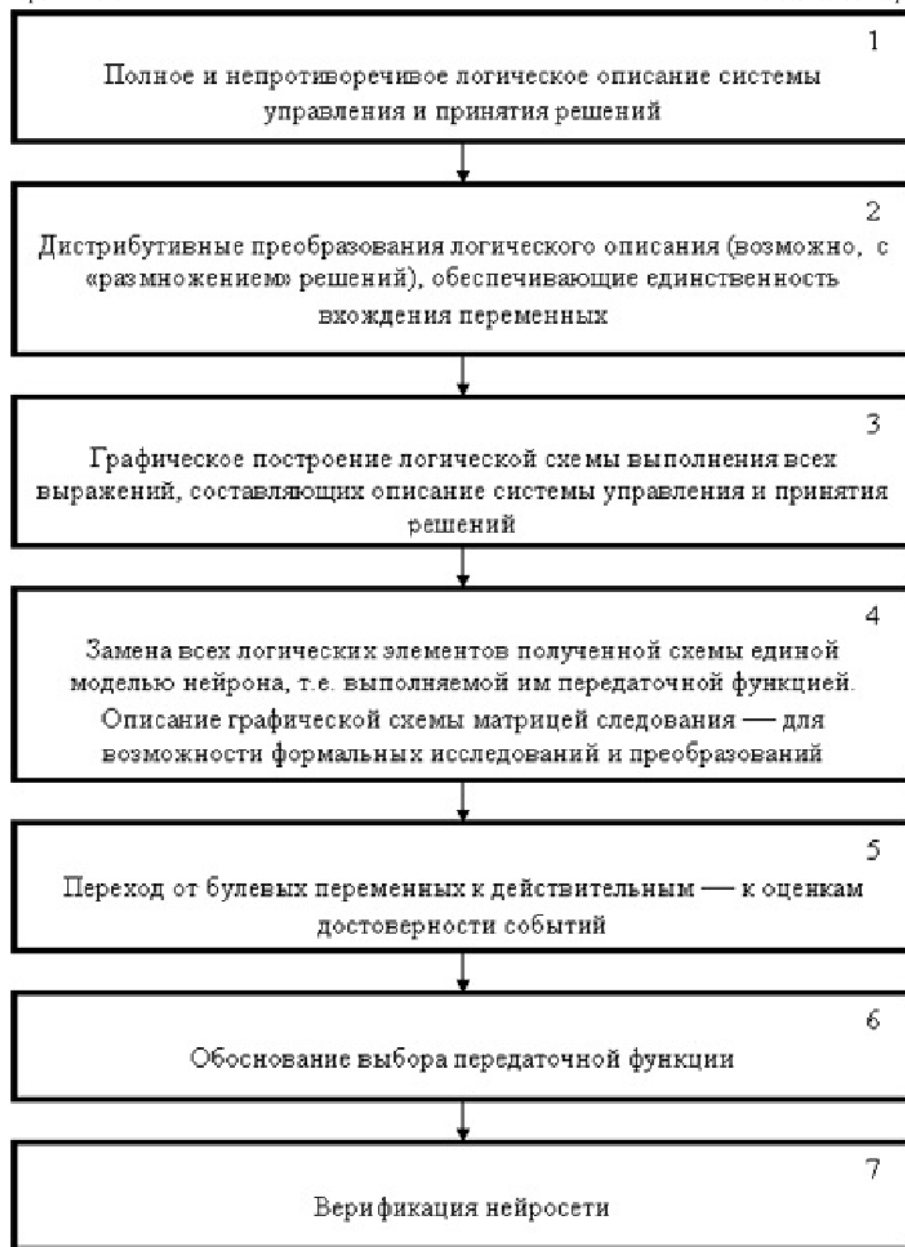


Рис. 12.1. Этапы разработки СПР

12.2. Логическое описание системы принятия решений

Ассоциативному мышлению адекватны логические нейронные сети,

создаваемые на основе *логического описания* системы управления или принятия решений в терминах алгебры высказываний. Если СПР функционально полностью определена и может быть описана в терминах математической логики, то по этому описанию целесообразно строить уже обученную нейронную сеть. Несложность такого построения способствует тому, что при модификации СПР обученная нейросеть может быть построена заново. В то же время правомерен подход, при котором нейросеть заданной структуры обучается для конкретного применения или переобучается. Алгоритмы такого обучения представлены в [лекциях 4](#) и [11](#).

Структура обученной логической нейросети полностью совпадает со структурой логической схемы системы управления и принятия решений.

Составление полного и непротиворечивого *логического описания* – первый этап выполнения методики построения логической нейронной сети для СПР.

В [лекции 1](#) показывалось, что в терминах алгебры высказываний (математической логики) СПР описывается *системой логических выражений* вида

$$\begin{aligned}
 & F_1(X_1, \dots, X_n) \rightarrow R_1 \\
 & \dots\dots\dots (12.1) \\
 & F_S(X_1, \dots, X_n) \rightarrow R_S,
 \end{aligned}$$

где $X_i, i = 1, \dots, n$, - логические переменные-высказывания о наступлении событий, принимающие значения ИСТИНА (1), если событие наступило, и ЛОЖЬ (0) в противном случае; $F_j, j = 1, \dots, S$, - логические функции указанных переменных-высказываний, значение ИСТИНА которых определяет принимаемое решение R_j .

Для *непротиворечивости* и *полноты* СПР необходимо и достаточно, чтобы для каждого набора высказываний не более чем одна логическая функция в (12.1) принимала значение ИСТИНА и чтобы не существовало такого набора переменных-высказываний на основе

факторного пространства событий, для которого в системе (12.1) все логические функции имеют значение ЛОЖЬ.

При логическом описании СПР используются *операции конъюнкции* и *дизъюнкции*, но не используется операция отрицания. Это объясняется тем, что в интересах корректности и *полноты* отображения предмета исследований в СПР каждый задействованный фактор должен учитываться в рамках модели *исчерпывающего множества событий*. Учитываются все возможные значения или состояния этого фактора. Каждый фактор А представлен множеством состояний (событий, значений, диапазонами значений)

$$A = \{A_1, \dots, A_N\} .$$

Это множество формируется так, что выполняется условие нормировки

$$\sum_N P_i = 1,$$

где P_i - предполагаемая пользователем (!) вероятность (*достоверность*) наступления события A_i .

Например, при построении системы штрафования водителей за превышение скорости V представляется логичным охват всего возможного диапазона предполагаемого превышения, например:

$$90 \text{ км/ч} < V \leq 120 \text{ км/ч}, 120 \text{ км/ч} < V \leq 160 \text{ км/ч}, \\ V > 160 \text{ км/ч} .$$

Такое описание учитываемого фактора, в данном случае – скорости, не оставляет открытым вопрос "Что делать, если скорость принадлежит неучтенному диапазону изменения?"

При рассмотренном способе корректного и полного учета всех возможных событий отрицание некоторого события означает возможность альтернативного выполнения (в смысле логической операции ИСКЛЮЧАЮЩЕЕ ИЛИ) всех других событий из *исчерпывающего множества*

$$\overline{A_i} = A_1 \vee \dots \vee A_{i-1} \vee A_{i+1} \vee \dots \vee A_N$$

12.3. Дистрибутивные преобразования логического описания

Вторым этапом выполнения методики, в силу специфики дальнейшего перехода от булевых переменных к *достоверности* событий, является *дистрибутивное преобразование* логических функций (12.1) к виду, который обеспечивает единственность вхождения всех переменных, участвующих в одном логическом выражении.

Это преобразование выполняется с помощью свойства дистрибутивности конъюнкции по отношению к дизъюнкции и дизъюнкции по отношению к конъюнкции, что составляет суть эквивалентного *дистрибутивного преобразования* (12.1).

Правила эквивалентных преобразований логических выражений широко известны. Однако указанная специфика свойства дистрибутивности гарантирует возможность с помощью последовательного вынесения за скобки обеспечить единственность вхождения каждой используемой переменной в отдельное логическое выражение.

Необходимость такого преобразования обусловлена заменой *операций конъюнкции* и *дизъюнкции* их приближенным "суррогатом" - *передаточной функцией*. Практически такая функция основана на суммировании (взвешенных) сигналов на входе нейрона. Тогда неоднократность вхождения одних и тех же сигналов становится недопустимой.

Однако приведенный ранее пример показывает, что не всегда подобное преобразование приводит к успеху. Тогда используется прием "*размножения*" *решений*, основанный на том, что логическое выражение разбивается на составляющие выражения, объединенные операцией дизъюнкции. Для всех таких выражений, входящих в описание вместо прежнего, определяется одно общее решение.

В результате таких преобразований каждое выражение *логического описания* в общем случае представляет собой скобочную запись с более чем одним уровнями вложенности.

Однако указанное свойство *передаточной функции*, сводящей две операции к одной, порождает возможность существенного упрощения нейросети, а именно, - сведение ее к однослойной.

Ранее было показано, что с помощью приема "*размножения*" решений такое сведение всегда возможно. Однако оно приводит к "дроблению" сети, к размножению путей получения одних и тех же решений, к перегрузке сети заданной фиксированной структуры на этапе ее обучения.

В то же время многие реальные процессы по самому смыслу своему (экономические системы, управляющие, технологические и др.) адекватны не просто однослойным логическим нейронным сетям, но даже сетям совершенным.

Анализ на возможность построения однослойной нейросети следует проводить на этом же этапе выполнения методики до выполнения *дистрибутивных преобразований*.

Самый простой способ такого анализа заключается в следующем. Для каждого логического выражения в (12.1) выпишем без повторения множества используемых в них переменных (сформируем обобщенные эталоны). Проведем совместный анализ обобщенных эталонов, составленных по всем логическим выражениям, и установим, отсутствует ли полное вхождение одного из таких эталонов в другой? Если такое вхождение отсутствует (пересечение обобщенных эталонов допустимо), то возможно построение однослойной нейросети, т.е. непосредственное соединение связями нейронов-рецепторов с нейронами выходного слоя. Более того, если все обобщенные эталоны содержат одинаковое число переменных, возможно построение совершенной нейронной сети.

Если некоторый обобщенный эталон входит в состав другого (пример такой СПР приведен в [лекции 9](#)), необходимо полное воспроизведение структуры логических связей, что приводит к многослойной нейросети.

12.4. Построение логической схемы

На третьем этапе графически, в соответствии со *схмотехническим*

подходом, применяемым в технологии построения электронных схем, составляется логическая схема выполнения всех выражений (12.1), преобразованная на втором этапе. Логические функции реализуются с помощью *конъюнкторов* и *дизъюнкторов*, связанных между собой в соответствии с представленными логическими функциями. При этом скобочная запись активно диктует структуру и взаимодействие логических элементов.

Такое представление соответствует общему виду (12.1). При допустимости построения однослойной нейросети, как говорилось выше, схема представляет собой структуру, при которой входы, на которые подаются переменные, непосредственно связаны с выходами, соответствующими решениям.

12.5. Построение матрицы следования для нейронной сети

Четвертый этап позволяет ввести в рассмотрение неопределенность и заключается в замене *конъюнкторов* и *дизъюнкторов* единой моделью нейрона, реализующего одну из простейших *передаточных функций*. Такая замена позволяет перейти от задания булевых переменных на входе (есть событие – нет события) к вероятности (*достоверности*) этих событий, то есть к заданию предполагаемых действительных оценок наступления этих событий.

За каждым событием оказывается закрепленным нейрон входного (*рецепторного*) слоя. Если событие отображает значение некоторого фактора, например температуры, то весь диапазон значений этого фактора разбивается на достаточные по точности малые диапазоны. Тогда при запросе к нейросети (в рабочем режиме) событие оценивается *достоверностью* того, что значение фактора принадлежит некоторому (возможно, нескольким) диапазону.

Такое разбиение областей значений факторов производится на этапе *логического описания* СПР так, чтобы впоследствии, на данном этапе выполнения методики, за каждым диапазоном оказался закрепленным нейрон-рецептор. Тогда события задаются *достоверностью* принадлежности значений факторов одному или более диапазонам

изменения.

В графическом исполнении нейросеть неудобна в работе. *Матрица следования*, эффективность которой для представления и обработки параллельных процессов доказана, позволяет строить простые алгоритмы и программы для выполнения всех операций над нейронами и связями нейросети.

В *матрице следования* за каждым нейроном закрепляется строка и столбец с номером нейрона. В строке указывается значение веса той связи, которая исходит от некоторого нейрона в данный.

12.6. Выбор и обоснование передаточной функции нейрона

На пятом этапе производится обоснование выбора *передаточной функции*.

Передаточная функция должна лишь условно, с наименьшей трудоемкостью воспроизводить аналог *операций конъюнкции* и *дизъюнкции*. Нет необходимости в моделировании реальных процессов, происходящих в нейроне. (Известно, что для функционирования и обеспечения жизнеспособности в "живом" нейроне происходит около 240 химических реакций.) Более того, ведь нас интересует только логика мышления, а не физические и химические механизмы ее реализации, сопровождающиеся сложными переходными процессами в отдельном *нейроподобном элементе*. (Это замечание можно считать критикой в адрес неуместных попыток воспроизведения т.н. сигмоидных функций при решении задач логического характера, хотя применение этих функций эффективно при решении многих "нейроподобных" задач.)

Требования к *передаточной функции*:

- эти функции в области преодоления порога должны быть *монотонно возрастающими* по каждому сигналу на входе нейрона;
- не должно быть "угасания" сигнала возбуждения при его прохождении по сети;
- сигналы возбуждения на выходном слое должны быть четко

- различимы по величине для различных эталонных ситуаций;
- должен быть примерно равным диапазон изменения величин возбуждения нейронов выходного слоя, закрепленных за разными решениями.

В лекции 9 приведен ряд *передаточных функций*, тривиальных по выполнению и используемых в примерах СПР. Однако в приведенных примерах возникает проблема *корректировки весов* связей нейронов выходного слоя для приведения величины возбуждения этих нейронов к одному диапазону изменения.

Более того, нейросеть, реализующую СПР, необходимо рассматривать как одно звено в цепочке логического вывода, вырабатывающее информацию для последующих звеньев цепочки. Это позволяет строить сложные, многоуровневые СПР, как цепочки логических выводов, реализующие индуктивное и дедуктивное мышление.

Конкретно это означает, что величины возбуждения нейронов выходного слоя должны, как и величины возбуждения рецепторов, принадлежать диапазону $[0, 1]$, моделируя все ту же *достоверность* выводов.

Представляется перспективной и подлежащей дальнейшему исследованию *передаточная функция 5*:

$$V := \frac{1}{n} \omega_i V_i$$

$V := \text{if } V \geq h \text{ then } V \text{ else } 0$, где n – количество активных входов нейрона

Легко видеть, что величина возбуждения каждого нейрона сети удовлетворяет указанному диапазону.

12.7. Верификация нейро сети

Последний, шестой этап выполнения методики заключается в *верификации* логической нейросети. Рассматривая точные, определенные ситуации как комбинации булевых значений, отражающих наступление событий, для которых составлялась

логическая схема, необходимо удостовериться, что максимальное возбуждение нейрона выходного слоя соответствует предполагаемому решению. Правильная работа нейросети может быть подтверждена и в том случае, если с учетом *достоверности* событий (действительных переменных) заранее известно предпочтительное решение. Соответствующий нейрон выходного слоя должен максимально возбуждаться.

После *верификации* логическая нейронная сеть может эксплуатироваться в рабочем режиме.

Таким образом, в основе логической нейросети лежит запоминание опыта на базе связей вида "если ..., то ...". Принятие решений производится по принципу "на что более всего похожа данная ситуация" или на основе оценки среднего. Входные ситуации задаются предполагаемыми оценками *достоверности* отдельных событий. Это позволяет использовать неполные и недостоверные данные.

Продолжая обобщение, следует отметить:

1. Моделирование ассоциативного мышления является основой создания таблиц, фиксирующих опыт вида "если ..., то ...", т.е. связывающих возникающие ситуации с принимаемыми по ним решениями. Распространение возбуждения в нейросети с помощью *передаточной функции* является эффективным механизмом интерполяции (экстраполяции) опыта, нахождения наиболее близкого решения, а также нахождения математического ожидания исхода.
2. Благодаря логическому методу построения, математический аппарат нейронных сетей становится простым и доходчивым способом реализации "бесформульных" вычислений. Разработка нейросети недорога, она не требует специальных знаний пользователя. В то же время нейросеть легко допускает обучение и развитие, включение новых факторов, изменение решений и т.д.
3. Проведенные исследования позволяют сформулировать основные положения методики построения системы управления или принятия решений на основе логической нейронной сети. Основные этапы разработки обученной нейросети для такой системы: логическое описание, составление соответствующей ему

логической схемы - получение структуры нейросети, переход от булевых переменных к действительным - к *достоверности* событий, обоснованию применяемой *передаточной функции, верификации* нейросети по известному опыту.

4. Применение логических нейронных сетей позволяет создавать доступные широкому пользователю компьютерные системы мониторинга, управления и принятия решений во всех сферах экономической и социальной деятельности. Рассмотренные в следующих лекциях проекты таких систем в области оценки финансов и риска, в управлении и диагностике, в искусстве и развлечениях показывают универсальность подхода, возможность переориентирования нейросетей на новые применения. В особенности это касается наиболее распространенных совершенных нейросетей – однослойных, в каждом решении использующих все факторы.

Нейросетевые технологии в экономике и бизнесе

Рассматривается принципиальная возможность применения совершенных логических нейронных сетей для банковского мониторинга и в системах оценки риска. На основе обобщения предлагаются принципы разработки унифицированного программного приложения для широкого круга пользователей в различных направлениях деятельности.

"Щоев ... (Углубленно.) Пускай теперь наука трудится, а человек около нее как на курорте".

А. Платонов. "Щарманка"

13.1. Табличный метод - основа искусственного интеллекта

Выражаем признательность и уважение к идеям, высказанным Н.М. Амосовым и Д.А. Поспеловым, которые, как нам кажется, опираются на принципы мозга, "не вынутого из черепной коробки", т.е. полностью следуют основам искусственного человеческого интеллекта. Мы не всезнайки, постигшие истину в последней инстанции. Не трогая мощный пласт духовных знаний, мы обращаемся к мозгу лишь как к средству мышления, столь же простому на логическом уровне, как молоток и лопата.

При этом мы сознаем, что, в общем-то, принципы мозга известны и активно используются. Да вся наша жизнь основана на накоплении опыта "если ..., то ...", "что это такое и к чему это ведет", на интерполяции и экстраполяции (предвидении) этого опыта (а то и просто гадании). Мы пользуемся незримыми таблицами в нашей памяти, принудительно и вольно заполняемыми за партой, за рулем, с министерским портфелем и без него, - крутя головой на шумной улице, за книгой, у станка и у мольберта. Мы учимся, учимся всю жизнь, от школьника, проводящего бессонные ночи за букварем, до умудренного опытом профессора - мастера спорта по волейболу. Ибо с теми же таблицами мы связываем не только принятие решений, но и движение, ходьбу, игру в мяч.

Выходит, не требуя духовной пищи, с помощью таких таблиц, отражающих факты "посылка - следствие", надежно связанных с мышечно-желудочно-кишечным трактом, можно безбедно прожить свою жизнь, не испытывая гнета творческих позывов?.. Не будем здесь трогать категории духовные, моральные, нравственные. Не коснемся всеислия посредственности, так, в целом, и проводящей жизнь. Ведь нас интересуют современные информационные технологии - максимально приземленные и формализованные.

Если противопоставить *ассоциативному мышлению* математический счет, то справедлив вопрос: каков вес счета в жизни человека? Как шло развитие, когда человек вообще не умел считать? (Кстати, вспомним тезис Д. Менделеева: "Наука начинается тогда, когда начинают считать.") На одном ассоциативном мышлении, развивая опыт и умея интерполировать и экстраполировать его? Можно спросить читателя: сколько раз сегодня вы считали? Но зато вы водили автомобиль, играли в теннис, торопились на автобус, рассчитывая (не тот смысл слова!) успех. Представляете, сколько бы вам пришлось считать (да еще где взять алгоритм?), для того чтобы поднять ногу для постановки на тротуар, минуя бордюр? Нет, мы не считаем, и это, пожалуй, основная сторона нашей интеллектуальной жизни, даже в науке и бизнесе. Механизмы чувствований, интуиции, автоматизма, которые мы, не в силах объяснить, адресуем подкорковому мышлению, на деле являются нормальными механизмами ассоциативного мышления с помощью таблиц базы знаний.

И главное - мы делаем это быстро! Как же нам не задуматься, как не "заглянуть в зеркало", пытаясь постичь и воспроизвести?

Наука и техника обладают колоссальным опытом построения сложных систем только лишь на основе алгоритмических, математических методов расчета, исключая какие бы то ни было методы ИИ. Алгоритмы вывода и наведения ракет-перехватчиков, расчет и выдача команд целеуказания и управления на борт по измеренным отклонениям впечатляют по объему вычислений, по частоте обновления данных и в конечном итоге - по требованиям к производительности вычислительных средств. И мы все более понимаем, что моделирование мозговых процессов, воспроизводящее методы ассоциативного мышления, открывает самые широкие возможности,

присущие живым организмам. Это и высокий универсализм, и высокое быстроедействие.

Природа реализует самые простые принципы, утверждая тезис, что "гениальное - просто". Сложность нагнетается количеством и структурированием простого. Мозг на логическом уровне, как мы поняли, действует на основе простых логических элементов типа "если . . . , то . . . ", "посылка - следствие", приемлемых даже для простейших. Огромное количество таких элементов в принципиально столь же простом взаимодействии, соединяясь и выстраиваясь в логические цепочки, воспроизводя принципы иерархии и рекурсии, образуют сложные выводы, тотчас же обращенные для новых выводов. (Здесь можно философски порассуждать о механизмах обеспечения эволюции, от структур простейших до человеческих и далее, о законе перехода количества в качество. Но это слишком далеко нас заведет и утомит в беспросветных спорах . . .)

Но, пожалуй, основное достоинство мозга - в способности параллельного выполнения сложных логических предикатов и последовательного выполнения длинных логических цепочек. Такое выполнение достигается, во-первых, при одновременной обработке всех сигналов, поступивших на рецепторы, и, во-вторых, при реализации каждым нейроном передаточной функции для поступивших величин сигналов. Передаточная функция является суррогатом логических функций булевых переменных, обращенных для обработки действительных переменных, для воплощения т.н. нечеткой логики.

Важно подчеркнуть, что информация перерабатывается в нейросети не непосредственно как таковая, а опосредствовано, на уровне оценок, на уровне ее воздействия на величину возбуждения рецепторов или нейронов *входного слоя* подструктур.

А как же наше формальное мышление, а как то, что И. Павлов назвал второй сигнальной системой, а как конструирование алгоритмов и правил счета? Мы полагаем, что это - все более и более высокие уровни того же логического мышления на основе правил вывода, наложение нейронной сети и продолжение логических цепочек, попутно приводящих к развитию образной памяти, - продукт роста в процессе Развития. Мы полагаем это вполне материально воплощенным и

потому реализуемым искусственно, подвластным моделированию и воспроизведению.

Хотя можно содрогнуться от подозрений в собственный адрес в сухости и заформализованности. Нет, наше Я, душевность и духовность вызывают в нас трепет и чувство непознаваемости, которое свойственно каждому нормальному человеку, строящему модель своего мировоззрения и постигающему смысл своего прихода в жизнь. Отметим могучие пласты человеческой деятельности, основанные на образном, эмоциональном, духовном мышлении. Это - искусство и творчество, основанные на красоте, любви, добре . . . Именно здесь мы начинаем чувствовать всю беспомощность своих построений . . .

И загасив острое желание углубиться в дебри философского наукоподобия, мы формулируем достаточный, "сегодняшний" принцип нейросети как элемента ИИ.

1. Мы должны признать, что основа имитации нейроструктуры мозга - это метод табличной интерполяции.
2. Таблицы заполняются или по известным алгоритмам счета, или экспериментально, или экспертами.
3. *Нейросеть* обеспечивает лишь высокую скорость обработки таблиц - за счет возможности лавинообразного распараллеливания.
4. Кроме того, *нейросеть* допускает вход в таблицу с неточными и неполными данными, обеспечивая приблизительный ответ по принципу максимальной или средней похожести.
5. Задача нейросетевой имитации мозга заключается не в преобразовании самой исходной информации, а в преобразовании оценок этой информации, в подмене информации величинами возбуждения рецепторов, искусно распределенных между видами, типами, параметрами, диапазонами их изменения или отдельными значениями.
6. Точно так же нейроны *выходного слоя* каждой подструктуры своим возбуждением указывают на закрепленные за ними решения. (Где носители этих записей?) В то же время, эти величины возбуждения на правах исходной опосредствованной информации должны быть готовы к использованию в следующем звене "длинной" логической цепочки, без внешнего вмешательства

в рабочем режиме.

13.2. Мониторинг банковской системы

"Где денег взять? Весь разорился я,

Все рыцарям усердно помогая.

Никто не платит".

А.С. Пушкин. "Скупой рыцарь"

В [28] приводится пример блистательного применения *самоорганизующихся карт* Кохонена (*SOM* - Self-Organizing Map [2] для исследования *банковской системы* России в 1999-2000 годах. В основе *мониторинга* лежит рейтинговая оценка. Исследование строится на автоматическом выполнении одной процедуры: по многомерному вектору параметров банков¹⁾ на "плоском" экране компьютера высвечивается результат кластеризации - отнесение банка к различным категориям преуспевания. Для отдельного исследования различных факторов необходимо выстраивать свои *SOM*. Предсказание можно строить лишь на основе анализа временного ряда оценок *SOM*. Новые *SOM* необходимо строить и для продления цепочки выводов, подключив новые данные "со стороны", например - данные политического характера.

Такой подход, несомненно, эффективен и результативен. Но представляется, что по сравнению с потенциалом мозговых нейроструктур он сдерживает размах и смелость мысли, не позволяет "тянуть" *длинные цепочки* "посылка - следствие", совмещать прогноз с анализом, оперативно учитывать складывающуюся ситуацию и вводить в рассмотрение новые факторы и опыт экспертов. Мы должны согласиться с тем, что мозгу все это подвластно, и вновь обратиться к его структурам, предлагая проект программных средств системы *мониторинга*.

13.2.1. Структура нейросети и способы обучения

Логические функции, лежащие в основе мониторинга, преимущественно основаны на конъюнкции логических значений, утверждающих определенные диапазоны изменения параметров или показателей банков. В общем виде такая логическая функция имела бы вид

$$N \wedge \{ \text{значение } i\text{-го показателя банка принадлежит диапазону } [a, b] \} \rightarrow \{ \text{указатель значения рейтинга} \}$$

В [28] представлены такие показатели:

- *собственный капитал*;
- *сальдированные активы*;
- *ликвидные активы*;
- *обязательства до востребования*;
- *вклады населения*;
- *коэффициент ликвидности*;
- *бюджетные средства*.

Фантазируя, мы могли бы развить эту систему показателей:

- *объем инвестиций в бурно развивающуюся экономику*;
- *объем прибыли*;
- *прошлый рейтинг* и значение миграции;
- *отчисления в фонд поддержки науки и образования*;
- *налоговые отчисления*;
- *отчисления в пенсионный фонд*;
- *отчисления в благотворительный и культурный фонды*;
- *участие в программах ЮНЕСКО и т.д.*

Такой простой вид логической функции, при переходе в область действительных переменных, говорит о достаточности однослойной нейросети (рис. 13.1), содержащей входной слой рецепторов и выходной слой, указывающий на результаты мониторинга.

При создании *входного слоя* необходимо учитывать не только текущие показатели, но и, например, динамику изменения *рейтинга - рейтинга* за прошлые периоды времени.

Выходной слой должен отражать не только *рейтинг*, но и экспертные рекомендации, а также другие решения и выводы.

Целесообразен простейший вид обучения (т.е. построения базы знаний, какой является наша сеть), который сопровождает концепцию создания нейросети "под задачу": непосредственное введение связей в совершенной нейросети оператором-исследователем "вручную" - от рецепторов к нейронам *выходного слоя* в соответствии с причинно-следственными связями. Тем самым сеть создается уже обученной.

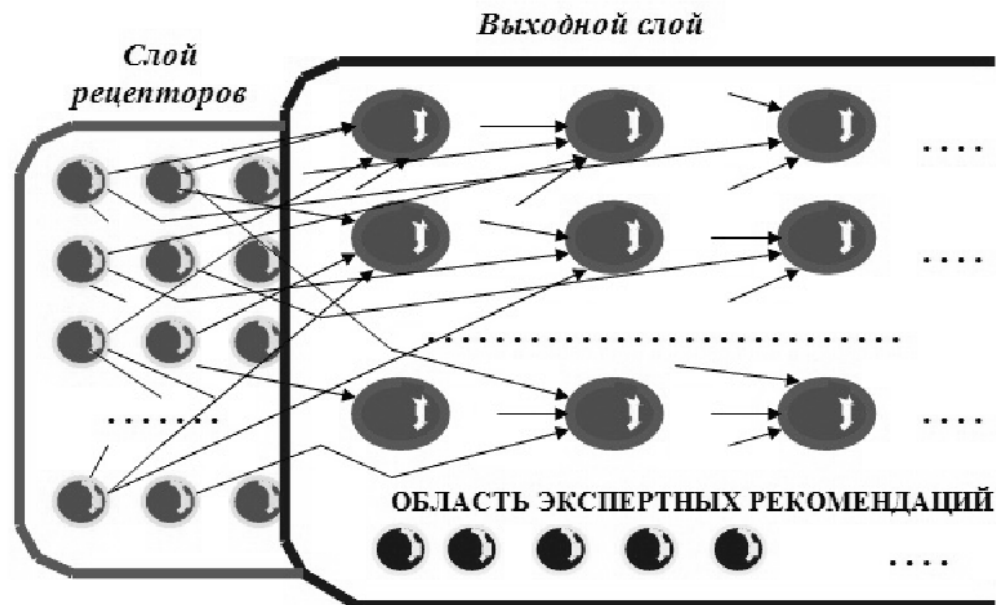


Рис. 13.1. Структура нейросети

Тогда передаточная функция должна быть тоже простейшей и основанной на суммировании величин возбуждения на входе нейрона, умноженных на вес связи (функция 5, лекция 9):

$$V := k \sum_j \omega_j V_j;$$

$$V_j = \begin{cases} V, & \text{если } V > h \\ 0, & \text{в противном случае} \end{cases}$$

Вес ω связей все же целесообразен (по сравнению с грубым заданием всех весов равными единице) в связи с возможным желанием оператора или эксперта в разной степени учитывать влияние различных показателей.

Порог h отсекает заведомо "плохие" выводы, упрощая дальнейшую обработку (например, нахождение среднего). Коэффициент приведения k обусловлен следующим. Максимальное значение V может достигать n - числа показателей. Чтобы свести значение *рейтинга* в некоторый приемлемый диапазон, например в $[0, 1]$, значения возбуждения надо преобразовать в этот диапазон, положив, например, $k=1/n$. Кроме того, этим коэффициентом, как и весами, можно регулировать значимость факторов в соответствии со сделанным ниже замечанием.

Принятые выше решения позволяют оперативно вносить изменения и уточнения оператором-экспертом-пользователем, развивать сеть, вводя новые факторы и учитывая опыт. Для этого оператору достаточно, "топнув" мышкой, выделить рецептор, а затем - нейрон *выходного слоя*, и связь установлена! Осталось только на основе предположения (ясновидения), осуществляя эксперимент, приблизительно установить вес введенной связи из диапазона $(0, 1]$.

И здесь мы должны сделать Очень Важное Замечание (ОВЗ), распространенное на весь материал и адресованное очень внимательному читателю.

Ранее, рассматривая обучение, мы четко классифицировали исходные эталонные ситуации, считая достоверность каждого компонента равной единице. Проводя затем трассировку, мы, прокладывая динамические *пути возбуждения*, также полагали веса связей равными единице (или некоторому максимальному постоянному значению). Но ведь можно сразу дать возможность (дать дополнительную степень свободы) учителю учитывать факторы в той степени, с теми весами, с которыми он пожелает! То есть допустить, что разные факторы в разной степени влияют на результат, и такое влияние заложить на этапе обучения

принудительно!

Например, известно, что накануне войны население в огромном количестве закупает мыло, спички и соль. Значит, следя за этим фактором, можно предсказывать скорое наступление войны. Создавая *нейросеть* для предсказания исторических или социальных событий, следует выделить один или несколько рецепторов, возбуждение которых соответствует разному уровню закупок мыла, соли и спичек одновременно. Возбуждение этих рецепторов должно передаваться, влиять (наряду с другими факторами) на степень возбуждения нейрона *выходного слоя*, "отвечающего" за заявление: "Скоро война!".

Однако мы вправе считать, что интенсивная закупка мыла, спичек и соли - необходимое, но не такое уж достаточное условие наступления войны. Оно может свидетельствовать, например, о бурном возрождении туризма в районе Главного Кавказского Хребта. В словах "не такое уж" заключается смысл нечеткой логики [30], позволяющей учитывать не непреложность влияния события, не булеву переменную "да - нет", а некоторое промежуточное, неопределенное, взвешенное предположение типа "влияет, но не так уж прямо, что уж обязательно . . . ". Поэтому мы сразу можем связи (все или некоторые), исходящие из данного (данных) рецептора, положить равными некоторой предполагаемой величине, меньшей единицы, корректируемой впоследствии, отражающей влияние возбуждения этого рецептора на вывод. В этом и состоит главный смысл коэффициента k .

Таким образом, одновременная закупка мыла, соли и спичек будет учитываться дважды: уровень закупки будет отображаться в уровне возбуждения соответствующих этим уровням рецепторов, а степень влияния этой закупки на вывод "Скоро война!" будет учитываться с помощью *весов синаптических связей*.

Согласитесь, что при построении одноуровневых сетей такой учет напрашивается, реализуется предельно просто, что мы и рекомендуем делать.

Конец ОВЗ.

13.2.2. Структура экрана рецепторов

Экран рецепторов (проект) показан на [рис. 13.2.](#)

Основную часть его составляет "окно прокрутки", через которое можно просматривать и задавать состояние *рецепторного слоя*, несомненно, не способного поместиться на статическом экране.

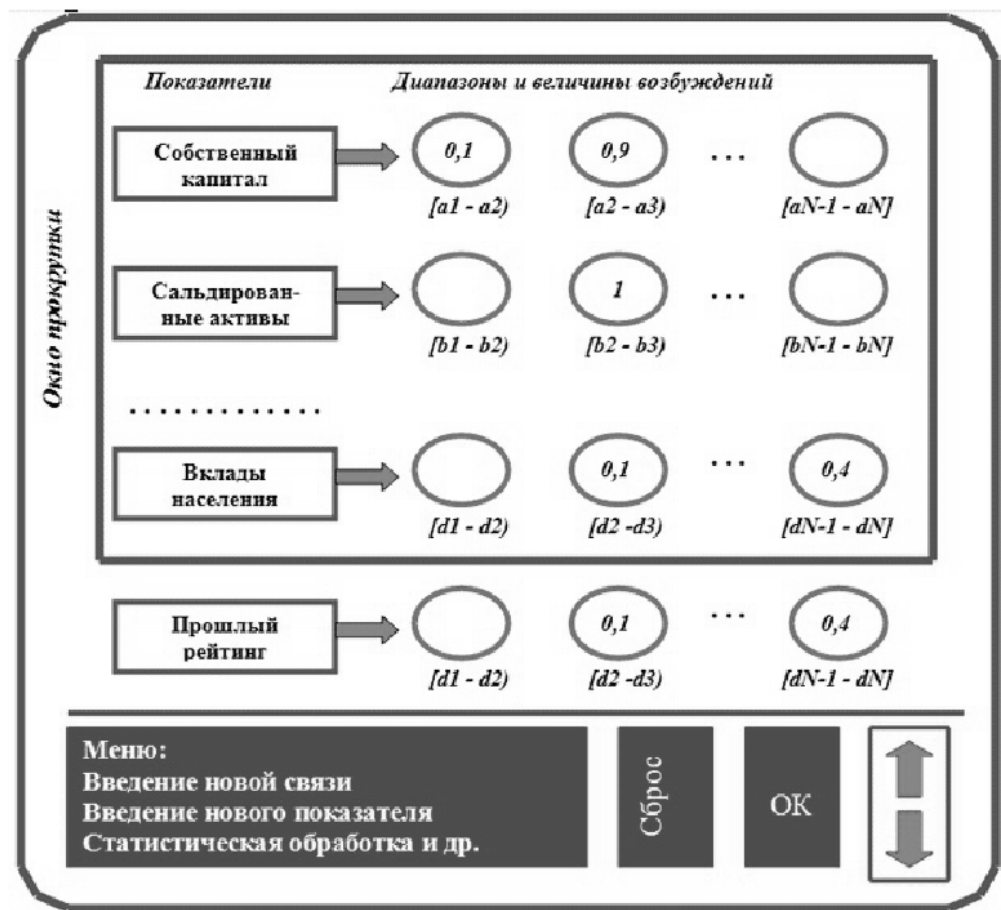


Рис. 13.2. Экран рецепторов

В "окне прокрутки" указаны показатели и их оценочные значения (в диапазоне $[0, 1]$), задаваемые соответствующим рецепторам. Эти значения - вероятностные, по достоверности, интуитивные, экспертные. Они могут предполагать охват нескольких рецепторов. Например, оценка того, что *собственный капитал* составляет не то 24, не то 34, не то 42 тыс. у. е., но скорее, все-таки, 24, может привести к приблизительной оценке задаваемых величин возбуждения, равных 0,6,

0,2 и 0,2 соответственно рецепторов, "отвечающих" за диапазоны (20-25], (30-35], (40-45].

На экране отображены статически (без прокрутки) задаваемые показатели, такие как *рейтинг* в результате прошлых измерений, выборочные "прошлые" показатели, а также показатели политической, социальной и экономической конъюнктуры. (Их обилие и развитие могут все-таки потребовать прокрутки.)

Должно быть отображено управление прокруткой, а также меню основных действий. Эти действия следующие:

- переход на экран *выходного слоя* ;
- статистическая обработка результатов (предполагает переход к выходному экрану);
- введение новой связи;
- введение нового рецептора;
- введение нового нейрона *выходного слоя* (предполагает переключение экранов);
- введение нового показателя и т.д.

13.2.3. Структура экрана выходного слоя

Экран *выходного слоя* (рис. 13.3) может отображать систему концентрических (вложенных) прямоугольников или других плоских фигур, отражающих распространение *рейтинга* по убыванию. В центре экрана яркими точками отражены самые преуспевающие банки или предполагаемые идеальные образы. Каждый элемент экрана жестко закреплен за нейроном *выходного слоя*. В результате *мониторинга* может максимально возбудиться нейрон, соответствующий эталону, что должно быть понятно отображено. Однако скорее всего высветится точка экрана, не совпадающая ни с каким эталоном и являющаяся промежуточной или усредненной. (Эталоны (см. ниже) отображают все категории банков, отнюдь не только самые преуспевающие.)

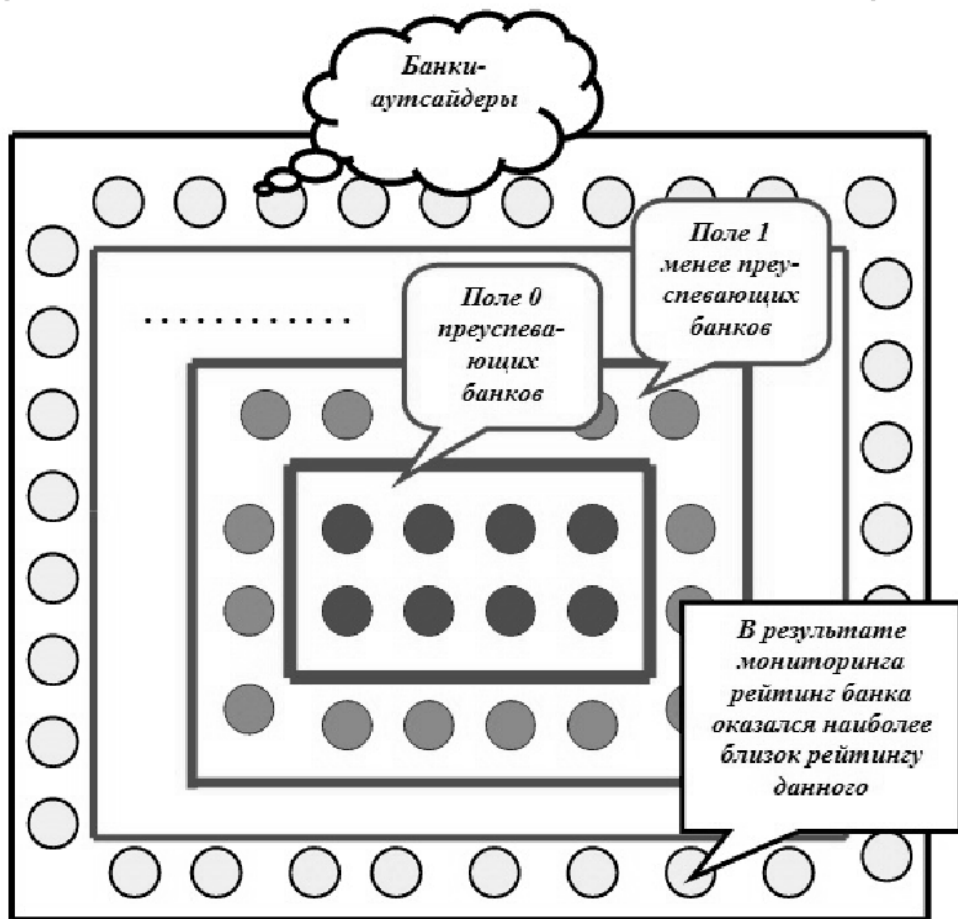


Рис. 13.3. Экран выходного слоя

Несомненно, должно быть отображено меню, позволяющее выполнить операции усредненной оценки *рейтинга*, демонстрации категории преуспевания, выдачи сигналов предупреждения, текстов-заключений, рекомендуемых стратегий развития, сохранения данных для дальнейшего развития и т.д.

13.2.4. Обучение нейросети

Для обучения нейросети на основе экспертных оценок должны быть известны диапазоны допустимых параметров, которые позволяют считать банк идеально преуспевающим, имеющим максимальный

рейтинг. Фиксируя несколько точек, координаты которых (множества значений параметров) удовлетворяют допустимым значениям *рейтинга* для известных или предполагаемых (с учетом возможных вариантов) банков, мы можем получить несколько идеальных представителей. Соответствующие им нейроны, т.е. элементы *экрана выходного слоя*, можно выделять произвольно, рассредоточивая по области экрана. Желательно, чтобы эталоны с большим *рейтингом* "располагались" ближе к центру.

Далее переходят к подобному же заполнению охватывающего прямоугольника, на основе следующей рейтинговой категории и т.д. - до банков-аутсайдеров.

Для проведения подобной работы эксперты должны предварительно сформировать таблицу, подобную табл. 13.1.

Таблица 13.1.

Показатели				Рейтинг	Эталоны
a_{11}	a_{12}	...	a_{1n}	1	Идеальные банки
a_{21}	a_{22}	...	a_{2n}	0,9	
...	
a_{r1}	a_{r2}	...	a_{rn}	0,86	
$a_{r+1,1}$	$a_{r+1,2}$...	$a_{r+1,n}$	0,85	Менее преуспевающие банки
...	
a_{s1}	a_{s2}	...	a_{sn}	0,8	
...	
a_{u1}	a_{u2}	...	a_{un}	0,7	Банки-аутсайдеры
...	
$a_{u+v,1}$	$a_{u+v,2}$...	$a_{u+v,n}$	0,1	

При нейронах, отображающих банки, запоминаются (и записываются при символах этих нейронов на экране) величины возбуждения (*рейтинги*), как говорилось выше.

13.2.5. Методика мониторинга

Обученная система, которая может поступить в распоряжение пользователя после усилий высококвалифицированных экспертов - экономистов и политиков, - готова к использованию в рамках *CASE-технологии* (CASE - Computer Aided Software Engineering; короче - средства программного обеспечения).

Однако, предварительно сняв блокировку вполне лицензированного продукта, пользователь реализует свое право на дополнительное обучение, уточнение (например, весов связей - для усиления или ослабления влияния некоторых показателей на основе собственного опыта), введение дополнительных показателей, - для эксперимента на свой риск и ответственность, и т.д.

Теперь, предположим, он хочет исследовать ситуацию, сложившуюся вокруг банка "Инвест-Туда-И-Обратно". Естественно, что никакие разведанные не в состоянии принести ему сколько-нибудь удовлетворительной информации о целесообразности собственных вложений. И он приступает к скрупулезному сбору данных, в результате чего получает приблизительные, вероятные (как ему кажется), разноречивые характеристики, которые решает вложить в модель.

Скользя "мышкой" по экрану *рецепторов*, пользователь задает значения их возбуждения, исходя из вполне достоверных данных, но иногда, - учитывая варианты "или - или" (частично возбуждая разные рецепторы), иногда, - по наитию, иногда, - просто пропуская показатели, не в силах преодолеть допустимую границу порядочности. Такие показатели, как прошлый *рейтинг* и миграция, он пока не знает, но предполагает полученный сейчас результат использовать в дальнейшем.

Введя данные, обозрев их последний раз, он со вздохом "топает" по клавише "ОК". Тотчас вспыхивает экран *выходного слоя*. Яркая точка где-то вблизи области аутсайдеров красноречиво свидетельствует о защите священного гражданского права ненасильственного выбора решения о целесообразности вложения праведно накопленного капитала.

Координаты этой точки на экране определяются по известной (ранее нами неоднократно используемой) формуле нахождения среднего по координатам "засветившихся" нейронов тех банков, которым близок

контролируемый банк, и по величинам их возбуждения. Но по этим же формулам на основе *рейтингов* "засветившихся" банков отыскивается *рейтинг* исследуемого банка!

Пользователь может принять решение о дополнении базы знаний (и, следовательно, нейросети) информацией о новом банке. Это особенно целесообразно, если совет экспертов подверг существенной критике получившийся результат и указывает тем самым на ошибку нейросети. Для этого достаточно воспользоваться опцией "Дополнить", в результате выполнения которой инициируется диалог компьютера с пользователем:

"Вы хотите изменить *рейтинг*?" - "Да"

"Новое значение *рейтинга*?" - "..."

"Сохранить!"

Тогда (один вариант) нейрон *выходного слоя* с найденными координатами закрепляется за новым банком. Устанавливаются его связи с теми рецепторами, которым было сообщено возбуждение при вводе информации о банке. Вес каждой связи полагается равным введенной пользователем величине возбуждения соответствующего нейрона-рецептора. Теперь база знаний дополнена наподобие пристрелянных установок артиллерийской батареи после поражения очередной цели.

Однако значительное принудительное изменение *рейтинга* должно приводить (второй вариант) к перемещению "засвеченной" точки в область банков, обладающих соответствующим уровнем *рейтинга*. То есть должно производиться перезакрепление нейрона *выходного слоя* за банком. Это также устанавливается в результате того же диалога компьютера с пользователем.

13.2.6. Корректировка и развитие

Выше мы уже говорили о необходимости и возможности постоянного уточнения и развития нейросети. Можно изменять представление о продвинутой банка-эталона - реального примера или

несуществующего идеала. Новыми такими примерами можно дополнить базу знаний - данную *нейросеть*. Можно корректировать веса связей как веса отдельных показателей. Этим будет скорректировано влияние конкретного показателя на выходной результат. Можно вводить новые показатели с их весами. Можно вводить в рассмотрение новые решения и устанавливать степень влияния на него тех же или новых показателей. Можно приспособить *нейросеть* для решения некоторых смежных задач, учитывая влияние отдельных показателей на миграцию банков (переход с одного рейтингового уровня на другой) и т.д.

Наконец, можно, приобретя данный программный продукт с отлаженным доступом и сервисом, с развитым набором функций преобразования сети, переделать *нейросеть* для решения совершенно другой задачи, например - для увлекательной игры в "железнодорожную рулетку", которую мы рассмотрели ранее.

13.3. Система оценки странового риска

Как видно из рассмотренного примера, используемые методы анализа и прогнозирования экономических показателей (в частности, биржевых сделок), таких как котировки и объемы продаж активов на фондовом рынке, выбор наилучшего проекта строительства, определение стратегии международных торговых сделок и т.д., основаны на анализе многофакторных моделей. Разнообразии факторов политического, экономического и социального характера, а также такие конкретные показатели, как уровень инфляции, кредитные ставки банков, уровень безработицы, обменные валютные курсы, изменение объемов производства, денежная эмиссия и эмиссия ценных бумаг и др., представляют значительные трудности при формальном описании задачи, а главное - при определении зависимостей прогнозируемого результирующего показателя (*котировка*, объем продаж, цена и др.) от совокупности этих факторов. Даже предположение о линейности или мультипликативности не уменьшают неопределенности при аналитическом построении таких зависимостей. Таким образом, задача анализа и прогнозирования экономических показателей относится к классу трудно формализуемых [12].

Для построения адекватной модели требуется представительная

статистика по всем факторам–аргументам. Статистические исследования накопленного опыта являются основой построения экспертных систем, являющихся ни чем иным, как *системами принятия решений*, моделирующими *ассоциативное мышление*. Это, в свою очередь, определяет целесообразность нейросетевого воплощения.

Исследование различных значений всех используемых факторов при априорном предположении о характере передаточной функции (можно с уверенностью считать, что такие функции, как 3 или 5, адекватны процессу ассоциативного выбора решений в данном применении) приводит к построению совершенной нейросети.

Входной (рецепторный) слой этой сети составляют нейроны, закрепленные за дискретными значениями или диапазонами изменения значения факторов. Нейроны *выходного слоя* соответствуют принимаемым решениям или оценкам ситуации. При этом возможна *корректировка весов* связей в соответствии со значимостью, степенью влияния конкретного фактора на принимаемое решение.

Возбуждение рецепторов задается в виде оценки достоверности принадлежности значения фактора данному дискретному значению или данному диапазону. Эта оценка производится на основе индивидуального опыта пользователя и является скорее интуитивной. Ряд факторов может быть оценен на основе официальной или конфиденциальной информации.

Рассмотрим на примере подход к проектированию нейросети, предназначенной для простейшей оценки странового риска.

Риск – *относительный размер* возможного ущерба или выигрыша, соотнесенный с вероятностью возникновения некоторого события в результате принятия того или иного решения.

Страновой риск возникает в связи с национальными особенностями функционирования государства, с которым предприятие или банк соотносят свою деятельность.

Экономическое воздействие на страновой риск в основном оказывают следующие факторы, на основе которых мы попытаемся спроектировать *нейросеть*, пронумеровав их:

1. Долговременное уменьшение роста валового национального продукта.
2. Быстрый рост цен на продукцию.
3. Увеличение импорта по сравнению с экспортом (отрицательное внешнеторговое сальдо).
4. Уменьшение доходов от экспорта.
5. Инфляция.
6. Резкое увеличение внешней задолженности.
7. Снижение объема обслуживания внешнего долга.
8. Уменьшение иностранных резервов по сравнению с импортом.
9. Криминальная ситуация в финансовой сфере.
10. Военные действия или наличие террористических организаций.

Очевидно, сеть должна быть восприимчива к развитию, ибо перечисленные факторы не исчерпывают всех возможных.

На рис. 13.4 создаваемая *нейросеть* представлена своим отображением на экране монитора компьютера и предполагает работу пользователя с ней.

Не следует рассчитывать на то, что ниже будут представлены хотя бы коэффициенты, с которыми факторы влияют на результат. Еще раз подчеркнем, что выявление зависимостей этого влияния – задача длительного исследования моделируемой системы, процесс проб и ошибок, набор статистики и анализ общественного и исторического опыта, выделение подструктур, для которых известны методы обсчета, - т.е. длительный процесс обучения учителя. *Нейросеть* "сама" (пока!) не мыслит. Она лишь является инструментом хранения опыта и быстрого извлечения рекомендаций по принципу "на что более всего это похоже". В этом смысле "хорошо" обученная *нейросеть*, т.е. сформированная на основе достаточно точного учета многих факторов, позволяет пользователю, не столь глубоко сведущему, принимать правильные решения при помощи коллективного опыта коллег и при неточных данных, утверждая, разумеется, тезис о том, что чудес не бывает.

Задача пользователя заключается в оценке достоверности каждого диапазона (или дискретного значения) изменения меры каждого фактора. Эти оценки (на рисунке - кружки) он заносит в "окна" соответствующих нейронов-рецепторов. Запись на основе

приблизительности предполагаемых оценок может производиться в несколько окон одного фактора, как показано на рисунке, приводя в возбуждение несколько рецепторов этого фактора.

Значение риска либо текст, объясняющий ситуацию и даже содержащий рекомендации, формируется в соответствующем окне.

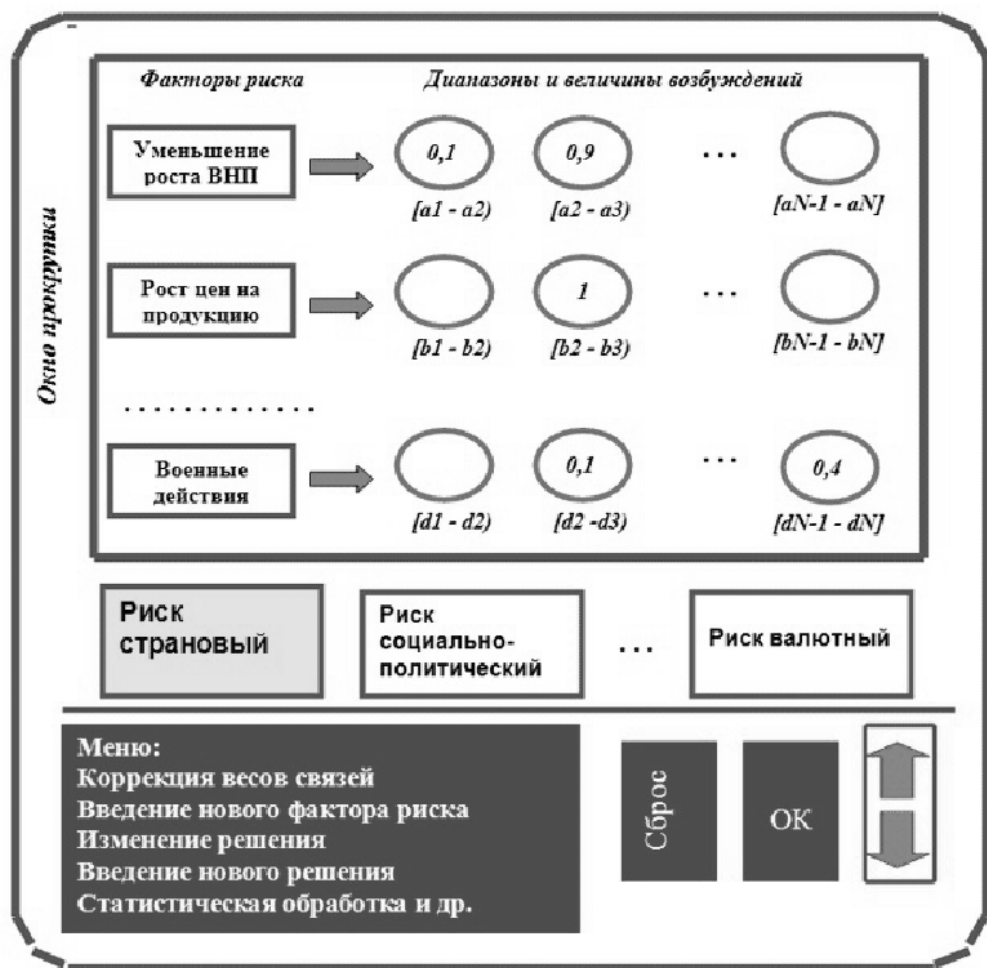


Рис. 13.4. Экран системы оценки риска

Одна *нейросеть*, в основном из-за большого количества факторов ориентированная на одну категорию рисков, может давать менее точные оценки рисков других категорий (не все факторы могут быть учтены). Однако возможна разработка нейросети, ориентированной на

все известные категории рисков, требующей энциклопедического, динамически уточняемого в процессе эксплуатации объема знаний для обучения (уточнения структуры связей и их весов).

Напомним, что обучение нейросети, т.е. построение связей между множеством диапазонов (значений) оценки факторов и множеством значений рисков, производится на основе экспертных оценок, а также в результате длительного личного и коллективного опыта. Пользователь может корректировать связи нейросети и веса этих связей в процессе эксплуатации.

Рассмотренные в лекции проекты требуют реализации в виде унифицированного стандартного программного приложения, которое, предоставляя широкий сервис пользователю, может быть легко переориентировано для многих применений. Следует подчеркнуть, что рассмотрены совершенные логические нейронные сети, которые, по-видимому, характерны для экономических задач. Это тем более утверждает целесообразность включения аппарата логических нейронных сетей в состав средств *CASE-технологии*.

1) Кстати, авторы обращают внимание на то, что нейросетевые технологии позволяют строить наглядные функции многих переменных, как бы преобразуя многомерное пространство в одно-, двух- или трехмерное

Нейросетевые модели пошаговой оптимизации, маршрутизации и тактических игр

Рассматривается применение технологии логических нейронных сетей при построении транспортных моделей, в которых реализуется пошаговая маршрутизация. Общность подхода для решения класса задач пошаговой оптимизации демонстрируется на возможном применении в моделях тактических игр.

"Будут игры беспредельные,

В упоительности цельные..."

К. Бальмонт. "Будут игры беспредельные..."

14.1. Логическая нейронная сеть - средство пошагового принятия решений

Задачи *пошаговой оптимизации* составляют широкий класс задач исследования операций. Минимизация (максимизация) значения критериальной функции достигается в результате выполнения нескольких последовательных шагов обработки исходных данных. Это - многочисленные задачи нахождения оптимальных стратегий управления, таких как вывод космического объекта в заданную точку, минимизация длины пути следования в *транспортной сети* и др. Как правило, такие задачи решаются методами динамического программирования, требующими обратного и прямого проходов. Однако схему решения задачи *линейного программирования*, а также транспортной задачи, также можно свести к схеме *пошаговой оптимизации*, если она отображает процесс параллельного или последовательного смещения в соседнюю вершину многогранника допустимых решений, с меньшим значением целевой функции.

То же можно сказать о решении задачи целочисленного линейного программирования (особенно - параллельного) и многих задач *нелинейного программирования*, а также задач, решаемых методом "ветвей и границ".

Однако высокая сложность задач указанного типа выдвигает проблему запоминания и использования опыта, т.е. применения элементов обучения.

При решении задач оперативного управления и планирования, по-видимому, нецелесообразно каждый раз, например, прокладывать маршрут следования груза по железной дороге с учетом огромного числа динамически возникающих факторов. Конечно, пользуются простыми эвристическими алгоритмами управления, достаточно детализированными и децентрализованными для возможности учета постоянно изменяющихся условий и обстановки.

В то же время *пошаговая оптимизация* примитивно вырождается в последовательно принимаемое решение вида "я нахожусь в состоянии X ; куда двигаться (что делать) дальше?" Такая простая схема движения к оптимуму и порождает возможность априорного расчета оптимальных стратегий изменения состояния системы, для того чтобы в рабочем режиме по параметрам целевой функции и по текущему состоянию системы находить запомнившийся предпочтительный переход в соседнее состояние, уменьшающий значение целевой функции.

В терминах динамического программирования это означает, что обратный проход - первый акт решения задачи - может быть выполнен заранее, вне рабочего режима системы управления. На его основе происходит обучение системы. В рабочем режиме по текущим исходным данным и по промежуточному состоянию системы определяется ее целесообразный, заранее обоснованный переход в следующее промежуточное или конечное состояние. Это - прямой проход.

Такая схема соответствует и идее ситуационного управления, и рассмотренной ранее схеме нейросетевой реализации управления.

К стратегии *пошаговой оптимизации* следует отнести и *тактические игры*, заключающиеся в последовательном выполнении противниками (преимущественно, двумя) действий, приводящих к минимизации некоторой целевой функции, например функции потерь. Здесь перед отдельным игроком стоит проблема выбора наилучшего хода для сложившейся ситуации.

Для этого, несомненно, могут каждый раз рассчитываться все

возможные варианты ходов с возможными ответными ходами. Может быть использован и теоретический опыт. Однако такой анализ требует огромной производительности вычислительных средств. Он недостаточно оперативен.

Здесь решение находится на основе анализа действий опытного игрока, помнящего эффективный выход из множества сложившихся ситуаций и эксплуатирующего свои способности ассоциативного мышления. Используется и коллективный опыт, теоретически обобщенный.

Автоматизация подобного анализа может быть произведена на основе разработки "подсказчика", хранящего в памяти большое количество ситуаций и рекомендующего следующий ход в соответствии с опытом экспертов, с историей и с теоретическим анализом.

Все сказанное выше определяет целесообразность реализации с помощью логической нейронной сети.

14.2. Нейросетевая транспортная модель динамической маршрутизации

"- Вперед! - скомандовал сам себе бравый солдат Швейк. - Долг зовет. Я должен попасть в Будейовицы.

Но по несчастной случайности, вместо того чтобы идти от Противина на юг - к Будейовицам, стопы Швейка направились на север - к Писеку".

Я. Гашек. "Похождения бравого солдата Швейка"

Транспортная сеть представляет собой конечное множество пунктов (узлов), соединенных между собой линиями связи. Ограничения на *топологию связей* отсутствуют.

Случайным образом или по некоторому закону (например, по расписанию) в сети возникают заявки на транспортное перемещение, на передачу сообщений или, в общем случае, - на движение объекта из пункта отправления к пункту назначения. Такими пунктами может быть любая пара пунктов сети. Заявки требуют прокладки маршрутов следования. Маршрут может предполагать как непосредственное

перемещение из пункта отправления в пункт назначения (если существует связь), так и транзитное перемещение через промежуточные пункты.

При моделировании целесообразно наблюдать единичный акт перемещения по каждому маршруту: из одного пункта в другой. Это позволяет применять динамическое управление реализацией маршрутов и изменять условия продвижения от пункта отправления к пункту назначения.

Модель ориентирована на решение задачи "проникновения" из пункта отправления в пункт назначения кратчайшим или менее загруженным доступным путем, т.е. за минимально возможное время. Решение этой задачи характерно для *маршрутизации* в компьютерной сети, включая Интернет, для почтовой связи, оптимизации грузовых перевозок всех видов, нефтяных и газовых магистралей и др. В то же время с помощью этой модели может быть составлено оптимальное транспортное расписание, например, движения поездов.

Для проекции сети на плоскость введем систему координат (x, y) , позволяющую однозначно идентифицировать (адресовать) каждый из N пунктов сети. Координаты могут быть декартовыми, географическими и др. В соответствии с их назначением, знание их точного значения не обязательно.

Пусть Δx - разность координаты x пункта назначения и пункта нахождения (отправления или промежуточного пункта), из которого следует произвести шаг перемещения - смещение; Δy - аналогичная разность координаты y .

Для полной реализации модели достаточна однослойная логическая нейронная сеть (рис. 14.1).

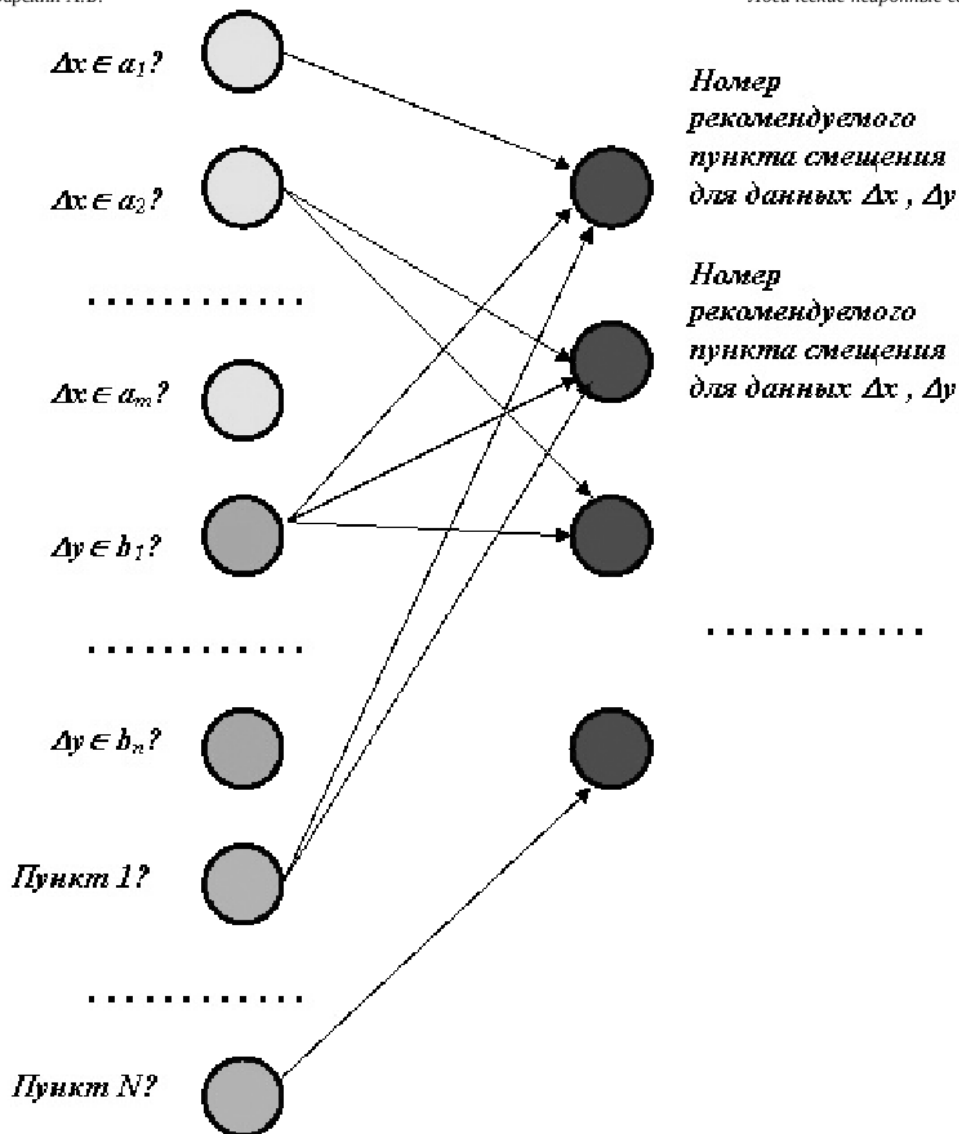


Рис. 14.1. Нейросеть для пошаговой маршрутизации

Разобьем весь интервал изменения Δx для данной транспортной сети на отрезки a_1, \dots, a_m . За каждым отрезком закрепим нейрон рецепторного слоя. Возбуждение этого нейрона определяется достоверностью принадлежности найденного текущего значения Δx соответствующему отрезку.

Весь интервал возможного изменения Δy также разобьем на отрезки b_1, \dots, b_n . За каждым отрезком закрепим нейрон-рецептор. Его возбуждение определяется достоверностью того, что текущее значение Δy принадлежит соответствующему отрезку.

Закрепим N рецепторов за пунктами сети. Возбуждение этих рецепторов определяет пункт нахождения - пункт отправления или промежуточный пункт, из которого следует произвести смещение.

Выходной слой состоит из N нейронов. Их возбуждение определяет пункты, в которые необходимо или возможно произвести смещение.

Рекомендация по смещению не обязательно однозначна. Могут выдаваться варианты смещения, которые подлежат дополнительному анализу с учетом различных динамических факторов.

Синаптические связи вводятся так, чтобы каждое единичное возбуждение рецепторов всех элементов тройки $\{a_i, b_j, \langle k\text{-й пункт нахождения} \rangle\}$ приводил к максимальному возбуждению нейрона выходного слоя, называющего пункт дальнейшего смещения.

Передаточную функцию целесообразно выбрать как

$$V_i = \xi \left(\sum_j \omega_j V_j - h \right), \text{ где}$$

$$\xi(z) = \begin{cases} z, & \text{при } z \geq 0 \\ 0, & \text{при } z < 0 \end{cases}$$

Таким образом, первоначально по разностям координат x и y пункта отправления и пункта назначения отыскивается промежуточный пункт смещения. Для продолжения имитации движения возбуждается нейрон рецепторного слоя, закрепленный за полученным пунктом смещения, т.е. за новым пунктом нахождения. По его координатам и координатам пункта назначения вновь определяются Δx и Δy . Возбуждаются соответствующие этим разностям рецепторы, и для данного пункта нахождения вновь по максимальному возбуждению нейрона выходного слоя определяется следующий пункт смещения и т.д.

Координаты каждого полученного пункта смещения сравниваются с координатами пункта назначения, т.к. в случае их совпадения маршрут следования реализован.

Изменение весов связей и порогов позволяет ввести элементы динамического управления движением. Так, увеличение порога нейрона выходного слоя целесообразно связать с нагрузкой соответствующего пункта (узла). Величина возбуждения нейрона, указывающего на этот пункт, может при этом зависеть, например, от количества занятых путей станции (при "железнодорожной" интерпретации) или от количества занятых каналов обслуживания (базы данных).

Такое регулирование величины возбуждения нейронов выходного слоя позволяет строить *транспортные сети* с использованием резервных, обходных путей, т.к. исходные построения обученной нейросети можно производить с учетом не одного, а более пунктов смещения. Тогда пункт смещения, "возбужденный" более других, может быть выбран для этого смещения.

Продвижение по сети с каждого пункта должно уменьшать значение соответствующего ему порога (на выходном слое) в зависимости от "физического смысла" задачи.

Регулировка синаптических весов нейрона выходного слоя может быть связана с загруженностью пути, линии передачи данных и других средств перемещения в *транспортной сети*. Такая регулировка требует введения дополнительного контроля состояния связи, соединяющей пункты. Синаптический вес такой связи должен снижаться при имитации движения. С этим снижением должен быть связан счетчик времени, в течение которого связь невозможна. Например, интервал следования поездов по одному пути обусловлен, в частности, необходимым расстоянием между поездами. В общем случае необходимо учитывать пропускную способность линий связи. По истечении определенного времени (когда счетчик "обнуляется") вес связи увеличивается до максимального (скорее всего, до единицы). Таким образом, даже если величина возбуждения нейрона, указывающего на пункт смещения, высока и позволяет принять движущийся объект, учитывается состояние средства связи.

Так как работа модели тактируется, то при запретах на смещение движущихся объектов нейросеть не подвергается изменению с увеличением модельного времени на такт. Уменьшаются лишь значения счетчиков задержки. При переходе к следующему такту модельного времени вновь исследуется возможность дальнейшего продвижения объектов с новыми значениями синаптических весов и порогов нейронов выходного слоя.

С помощью подобных счетчиков можно организовывать задержки в пунктах сети для имитации остановок движения, например, поездов.

При построении *обученной нейронной сети*, моделирующей *транспортную сеть*, целесообразно усилить влияние разностей координат пункта назначения и пункта отправления таким образом, чтобы учесть меньшее количество промежуточных пунктов следования. То есть рекомендуемый пункт очередного смещения для "дальних" маршрутов может и дальше отстоять от пункта назначения или промежуточного пункта следования. Например, скорый поезд, следующий в Астрахань, первую остановку делает в Рязани, и прокладывает ему маршрут, указывая Малаховку как пункт смещения, уместный для электропоезда, вряд ли целесообразно.

14.3. Пример модели транспортного маршрутизатора из центрального пункта отправления

Рассмотрим пример транспортного маршрутизатора, планирующего движение из одного, центрального пункта к периферийным пунктам со специальной *топологией связей*. *Транспортная сеть* представлена на рис. 14.2, где все пункты заданы своими координатами (x, y) в системе координат, связанной с центром.

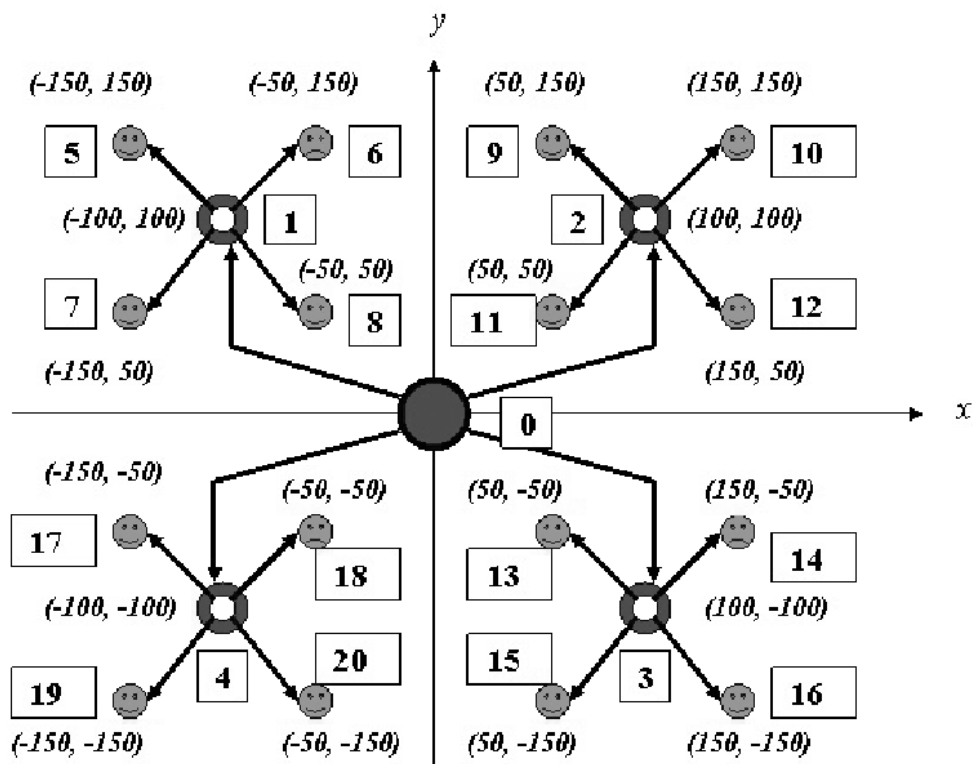


Рис. 14.2. Транспортная сеть

По величине и знаку разности координат пункта назначения и исходного или промежуточного пункта нахождения с помощью простейшей однослойной нейросети (рис. 14.3) может быть найден пункт смещения, т.е. тот пункт, в который следует переместиться для дальнейшего следования. Целесообразно использовать рассмотренную ранее передаточную функцию, положив все веса равными единице, а пороги - равными двум.

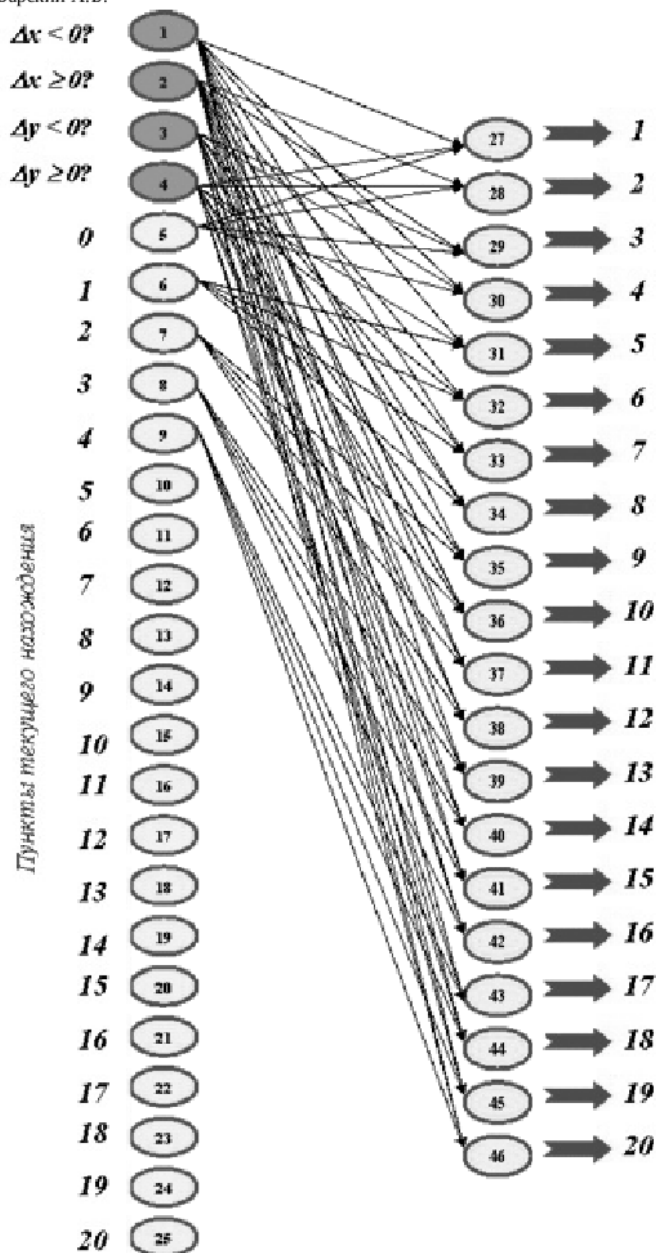


Рис. 14.3. Транспортная нейросеть

Как рассматривалось выше, нейросеть отображает рекомендуемые смещения для всех возможных ситуаций, складывающихся на основе текущего нахождения объекта и соотношения между его координатами и

координатами пункта назначения.

Если координаты найденного пункта смещения совпадают с координатами пункта назначения, маршрут реализован.

В противном случае отыскиваются разности координат пункта назначения и найденного пункта смещения. Они определяют единичное возбуждение соответствующих рецепторов входного слоя. Так же возбуждается рецептор, соответствующий найденному пункту смещения. Эти три возбужденных рецептора вновь определяют новый пункт смещения.

Так продолжается до совпадения координат пункта смещения с координатами пункта назначения. В данном примере потребуется сделать не более двух шагов.

Например, при движении из пункта 0 по адресу пункта 6 с координатами (-50, 150) оказывается, что $\Delta x < 0$, $\Delta y \geq 0$. Тогда нейрону 1 сообщается значение возбуждения, равное 1. Такое же значение сообщается нейрону 4, а также нейрону 5, соответствующему центральному пункту. Тогда $v_{27} = 1$, $v_{28} = \dots v_{46} = 0$. То есть найден промежуточный пункт 1. Тогда возбуждение нейрона 6 полагается равным единице и устанавливается, что относительно пункта 1 координаты пункта назначения определяют неравенства $\Delta x \geq 0$, $\Delta y \geq 0$. Значит, следует сообщить единичное возбуждение нейронам 2 и 4. Возбуждение всех других рецепторов полагается нулевым. При расчете передаточной функции для всех нейронов выходного слоя выдается рекомендация следования в пункт 6, т.к. единичное возбуждение приобретает нейрон 32. Маршрут составлен: 0 -> 1 -> 6.

Нейросеть, представленная на [рис. 14.3](#), предполагает одностороннее перемещение - от центра к периферии. Поэтому конечные пункты оказались "зависшими", не влияющими на дальнейшее смещение. Однако легко развить эту нейросеть (нецелесообразно усложнять рисунок), предположив возможность возвращения в центральный пункт, или движения через центральный пункт, если задан маршрут из пункта отправления одного сектора в пункт назначения другого сектора. Внутри сектора движение может осуществляться через его локальный

центральный пункт также на основе анализа указанной выше разности координат.

Таким образом, моделирование сложной *транспортной сети* любой природы с помощью логической нейронной сети оказывается исключительно простым и универсальным.

14.4. Моделирование реального времени движения в сети железных дорог для оценки пропускной способности, эффективности и оптимизации расписания

Проектируемая модель строится на основе логической нейронной сети, представленной на рис. 14.3, и должна удовлетворять следующим требованиям:

- реализация дискретного шага модельного времени для имитации движения в реальном времени;
- одновременное обслуживание многих маршрутов;
- отсутствие ограничений выбора маршрутов внутри сети;
- приоритетное обслуживание маршрутов;
- учет загруженности станций;
- учет загруженности магистралей и порядка следования по ним;
- возможность выбора резервных или дублирующих путей следования;
- возможность предусмотренных остановок, выполнения скоростного режима или расписания движения;
- фиксация характеристик пути движения из пункта отправления к пункту назначения для формирования оптимального расписания;
- учет возможностей эксперимента и развития: добавление новых станций, увеличение их пропускной способности, увеличение пропускной способности магистралей и др.;
- модель должна быть универсальной, нетрудоемкой при разработке, быстродействующей и обладать дружественным интерфейсом.

На рис. 14.4 приведена укрупненная схема решения задачи

моделирования как задачи обоснования решений о развитии железнодорожной сети с помощью модельного эксперимента, а также выбора оптимального расписания движения.

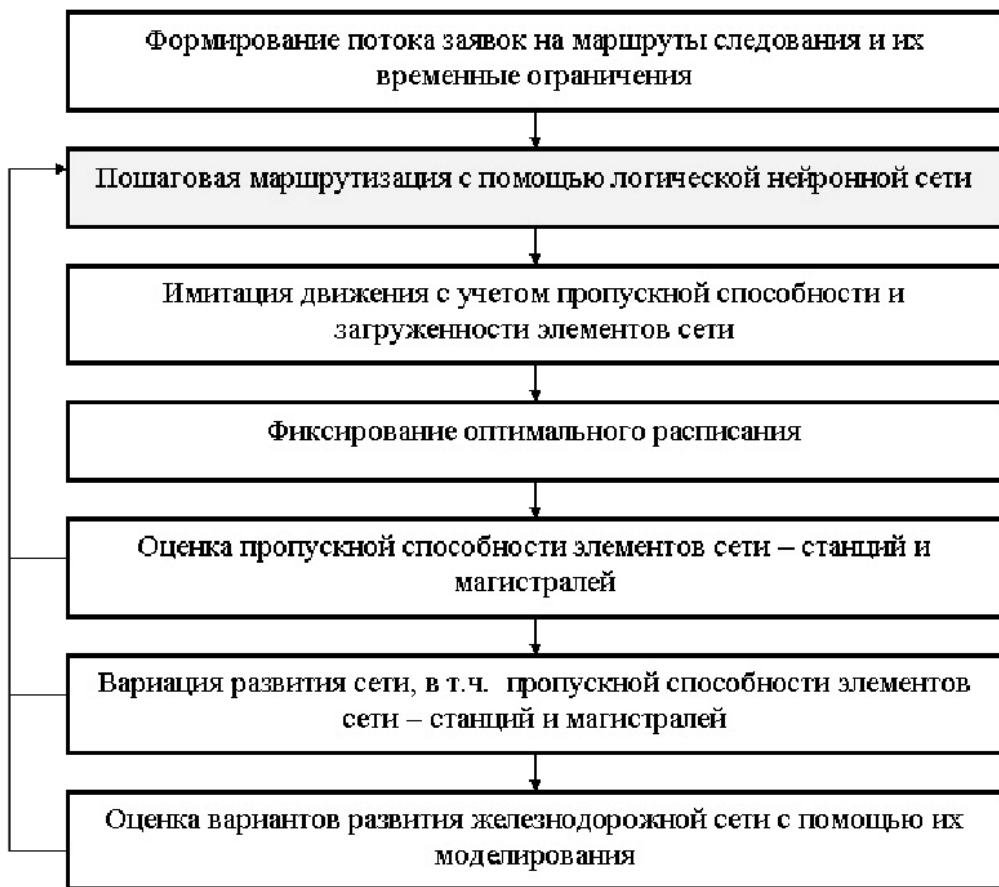


Рис. 14.4. Схема проведения исследований с помощью модели железнодорожной сети

В ее основе лежит однослойная логическая нейронная сеть (рис. 14.3), определяющая очередное перемещение движущегося объекта (поезда) в соответствии с текущей ситуацией. Дополнительные построения позволяют учитывать текущие возможности станций, занятость магистралей и интервал движения по ним, необходимость пребывания на станции не менее положенного времени.

Блок-схема проектируемой модели представлена на [рис. 14.5](#).

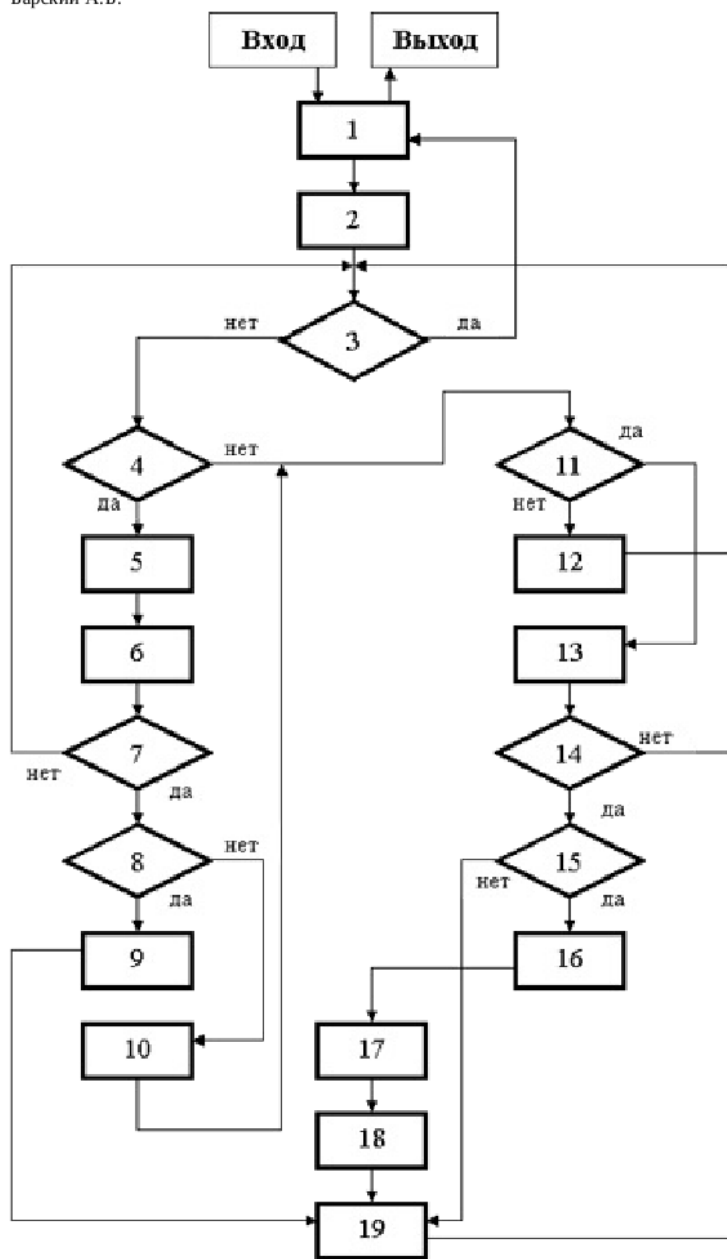


Рис. 14.5. Схема модели

Блок 1 реализует функции управления, связи с пользователем и формирования транспортного потока. Каждая заявка на выполнение маршрута предполагает задание координат пункта отправления и пункта

назначения, время начала движения, приоритет маршрута (литер).

В блоке 2 реализован счетчик модельного времени. Осуществляется переход к следующему такту работы модели, т.е. реализуется условная дискретность, адекватная режиму реального времени.

Блок 3 управляет циклом обработки всех выполняющихся маршрутов. Если все маршруты в такте обработаны, допускается возможность генерации (в блоке 1) новых маршрутов. Исследование маршрутов в порядке их приоритета обеспечивает преимущественное, первоочередное использование ресурсов – узлов и магистралей.

В блоке 4 проверяется, движется ли объект по магистрали, или находится на станции.

Имитация реального времени движения по магистрали, связывающей два узла, производится с помощью счетчика, связанного с объектом при его движении. Счетчик учитывает длину пути, скорость и расстояние между станциями, переведенные в количество тактов движения. Уменьшение этого счетчика на единицу в блоке 5 имитирует движение.

Так как объект занял магистраль, то для соблюдения интервала движения повышается *вес синаптической связи*, имитирующей эту магистраль. По мере продвижения объекта этот вес повышается (восстанавливается до единичного значения) в блоке 6 по мере продвижения.

В блоке 7 проверяется, достиг ли указанный счетчик нулевого значения? Это означает прибытие поезда на станцию.

Если достиг (блок 8), проверяется, конечный ли это пункт следования. Если да, маршрут полностью выполнен и осталось обработать информацию о его следовании (блок 19).

Если достигнутая станция не является конечной, в блоке 10 взводится счетчик времени (числа тактов) предусмотренной остановки на данной станции. Производится увеличение порога передаточной функции, соответствующей этой станции, т.к. загрузка этой станции увеличилась. Это должно быть учтено при рассмотрении возможности ее использования другими маршрутами.

Блок 11 выполняется в случае нахождения объекта движения на станции. В нем проверяется, исчерпан ли счетчик "остановки".

Если этот счетчик не равен нулю, в блоке 12 производится его уменьшение на единицу и осуществляется переход к анализу следующего маршрута (блок 3).

Если указанный счетчик исчерпан, объект (поезд) готов к дальнейшему движению. В блоке 13 производится расчет передаточной функции (раздел 2) для всех нейронов выходного слоя. Максимальная величина возбуждения определяет следующий пункт смещения объекта в сети.

Однако ранее увеличенные пороги передаточной функции, соответствующей пункту смещения, могут привести к тому, что ни один нейрон выходного слоя не возбудится. Это говорит о необходимости дополнительной задержки на станции, что проверяется в блоке 14.

Если нейросеть указала пункт смещения, в блоке 15 проверяется: позволяет ли загрузка магистрали (вес соответствующей связи) начать движение?

Если нет, в блоке 16 увеличивается значение счетчика тактов времени пребывания на станции.

При положительном результате анализа в блоке 17 рассчитывается и "взводится" счетчик движения объекта по данной магистрали.

В блоке 18 снижается *вес синаптической связи*, ведущей из пункта пребывания к пункту смещения.

В блоке 19 производится фиксация всей информации о маршруте для составления расписания и временной диаграммы движения.

Таким образом:

1. Аппарат логических нейронных сетей в наибольшей степени адекватен ассоциативному мышлению человека, реализующего последовательное принятие решения для его дальнейших действий на основе изменения ситуации.
2. *Транспортные сети*, к которым следует отнести как

информационные, так и железнодорожные, реализуют маршруты следования, которые можно рассматривать как последовательный выбор пункта смещения, исходя из текущего места нахождения и конечной цели.

3. Применение элементов искусственного интеллекта - ассоциативного мышления - позволяет запоминать целесообразное смещение в различных ситуациях и строить простейшие модели следования в железнодорожных сетях.
4. Так как эти действия целенаправленны и оперативны, то имитация природных механизмов "бесформульных" вычислений, как при моделировании сложных систем, так и в основе управления ими, позволяет существенно расширить возможности используемых вычислительных средств.
5. Это демонстрирует предлагаемая схема универсальной модели движения в *транспортной сети*, учитывающая динамику приоритетного следования многих объектов в реальном времени, их совместное влияние на пропускную способность и эффективность, как время и стоимость доставки грузов. Модель допускает нетрудоёмкое введение и исследование вариантов, что обеспечивает нахождение оптимальных решений при проектировании сетей железнодорожного транспорта.

14.5. Нейросетевой "подсказчик" в тактической игре

"Остап ... подошел к одноглазому, сидевшему за первой доской, и передвинул королевскую пешку с клетки e2 на клетку e4".

И. Ильф, Е. Петров. "Двенадцать стульев"

Схема *пошаговой оптимизации* наилучшим образом ложится на схему игры, где последовательные действия одного или нескольких игроков приводят к успеху, обусловленному правилами.

Единичное действие, чаще всего называемое ходом, должно либо статистически, либо комбинационно приводить к увеличению "качества" или к уменьшению "штрафа" на пути к победе (или к поражению). И здесь большое значение имеет не только длительный анализ всех возможных продолжений, грозящий цейтнотом, но и огромный опыт и фактические знания, переродившиеся в интуицию и

позволяющие действовать механически в условиях блиц-турнира. Следовательно, должны быть реализованы механизмы запоминания и извлечения опыта и знаний в пошаговых действиях. Основным средством такой реализации является нейросеть - как природная, так и искусственная.

Рассмотрим игру в шахматы. Аналогом пункта в *транспортной сети* здесь является позиция на шахматной доске, состоящей из 64 клеток. Каждая клетка может быть пустой или иметь значение символа занимаемой фигуры.

То есть каждая клетка i может принимать значение из множества $\{\emptyset, \text{пешка белая, ладья белая, конь белый, слон белый, ферзь белый, король белый, пешка черная, ладья черная, конь черный, слон черный, ферзь черный, король черный}\}$. При этом позиции являются симметричными относительно цвета фигур. Игроку, прибегающему к услугам "подсказчика", главное - указать: "фигуры мои - фигуры противника". Играть можно "самому с собой", как бы поворачивая доску после очередного хода. (Если "подсказчик" играет сам с собой, логично предположить, что такая игра всегда будет сводиться к ничьей?)

Тогда рецепторный слой однослойной логической нейронной сети должен состоять (рис. 14.6) из 64 групп нейронов. Каждая группа закреплена за одной клеткой и, в свою очередь, состоит из 13 нейронов-рецепторов. Каждый рецептор закреплён за одним из возможных значений клетки.

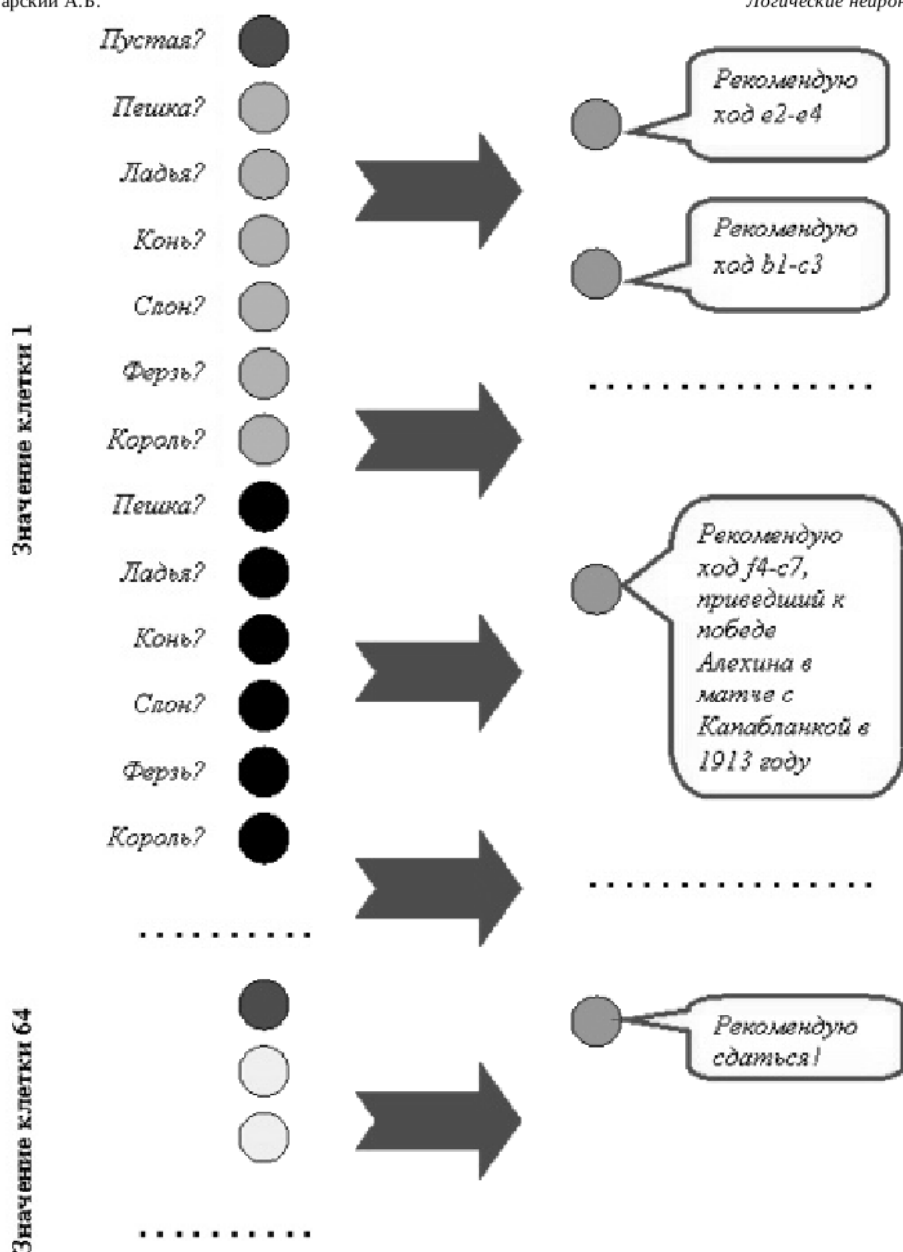


Рис. 14.6. Нейросетевой "подсказчик"

Ясно, что в подавляющем числе случаев значение клетки достаточно задавать с помощью единичного значения возбуждения единственного рецептора из тринадцати, соответствующих этой клетке. Так можно

задавать позицию для определения следующего хода. Однако можно предусмотреть и неполную, предполагаемую с некоторой достоверностью информацию о значении клетки. Это справедливо, например, для случая игры "вслепую", тем более в сеансе одновременной игры, когда детали ситуации на отдельной доске могут быть забыты.

Целесообразно использовать приведенную выше целевую функцию с единичными синапсическими весами. Значения порогов несущественны; они могут быть положены равными нулю.

Комбинации возбуждения рецепторов должны приводить, в соответствии с непосредственными связями, к максимальному значению возбуждения того нейрона выходного слоя, с которым связан текст - рекомендация следующего хода. Текст может включать исторические ссылки, комментарии, мультимедийные эффекты и др.

Как говорилось, возможно попеременное обращение игроков к *нейросети*-*"подсказчику"*. Тогда все нечетные обращения соответствуют белым, черные - черным. Однако интереснее игра живого шахматиста с компьютером.

Конечно, анализ колоссального опыта гроссмейстеров и литературы по теории шахмат не способен по всем возможным ситуациям и для белых и (симметрично) для черных определить абсолютно правильные ходы.

Да и объем информации колоссален! Останутся шахматные позиции без рекомендаций. Здесь необходимо исследовать, насколько указание нейрона, наиболее возбужденного, может быть принято в качестве совета, - то есть насколько это хотя бы статистически соответствует правильному решению или, по крайней мере, не приводит к снижению качества. Следует ли *"учить"* нейросеть решению по данной комбинации или достаточно использовать ее способности ассоциативного мышления?

На этом пути может производиться совершенствование *"подсказчика"*, что повышает интерес именно игры человека с машиной.

Основы "живого" моделирования

На основе предположения, что логические нейронные сети адекватно отображают принципы работы мозга, обсуждается проблема моделирования "живых" объектов — как натурального, так и компьютерного. Для компьютерного моделирования предлагаются математические основы имитации движения и деформации объектов в трехмерной памяти, отображающей пространственное их размещение. Обсуждается возможность построения и применения реагирующих объектов для оперативного отображения ситуаций в сложных и территориально распределенных управляющих системах, а также в искусстве.

"Мы, собаки, в смысле нервной восприимчивости в семь и еще много раз тоньше людей... Я чувствую тайными, неведомыми, дрожащими путями, как их души краснеют, бледнеют, трепещут, завидуют, любят, ненавидят".

А.И. Куприн. "Мысли Сапсана о людях, животных, предметах и событиях"

15.1. Какую задачу мы хотим решить?

Рассмотрим принципы построения управляемых видимых объектов в *трехмерной памяти* компьютера. Для имитации их движения используем "*мышцы*" и *шарниры*. Мышцы воспринимают команды на сокращение. Предложим *эмпирические формулы* для переноса содержимого ячеек памяти при имитации движения. Движение может задаваться программно. Однако подсоединение объектов к *логической нейросети* открывает большие возможности непредсказуемости реакции объектов на внешние факторы. Это, в свою очередь, открывает перспективу применения *реагирующих объектов* в сфере игр, развлечений, сценических постановок, в *интеллектуальных системах отображения* и в других видах искусства и бизнеса.

Во всех предыдущих лекциях в качестве примеров возможного применения обсуждается построение систем с элементами искусственного интеллекта, которые можно отнести к области

искусства и развлечений. Фантазия легко связывает создание монстров, объектов компьютерных игр, имитацию их реакции на внешнюю среду или другие предлагаемые им объекты и ситуации с такими "мероприятиями", как парк фантазмагорий, Disneyland, цирк марионеток, распознавание "свой — чужой", сетевые зрелищно-рекламные демонстрации и т.д. Такие системы могут представлять как искусство, так и бизнес.

Существуют два направления реализации указанных систем ИИ: натурное и компьютерное.

В современном парке (культуры и отдыха) можно в натуральном виде встретить не одно существо, мало отличающееся от живого и совершающее действия и движения по заложенной программе. Технологии их построения можно считать отработанными, имитация движений, включая работу "мышечного" аппарата, вполне совершенна. Отсутствие элементов ИИ, предполагающих непредсказуемость реакции, — основной недостаток таких объектов.

Компьютерная реализация сегодня ограничена мультимедийными средствами двумерной и трехмерной графики. Возможности создаваемых этими средствами систем и образов огромны. Достаточно отметить такое средство, как 3D-MAX, позволяющее создавать целые кинофильмы. Однако основной недостаток всех известных систем кроется в априорном знании возможных сцен, сглаживание перехода через которые в реальном времени имитирует движение. Это подобно табличному заданию огромного количества значений функции многих переменных при игнорировании непосредственного расчета.

Непредсказуемость сцен, возникающая при имитации реакции моделируемого объекта на внешние воздействия, может потребовать огромного, практически нереализуемого числа предполагаемых возможных сцен.

В этом случае желательно при построении объекта (монстра, человека и др.) снабжать его управляемыми органами движения (модификации, деформации, имитирующей улыбку, движение руки и т.д.), подобными *мышцам*. Их необходимо соединить с управляющим элементом — жестко заданной программой. Однако такая программа может предполагать анализ ситуации или внешних воздействий. Это уже

напрямую требует применения аппарата *логических нейронных сетей*, одновременно и в разной степени реализующих ряд отношений вида "если ..., то ...", что наиболее близко к поведению живого существа.

Таким образом, построение реального объекта в памяти компьютера, а не его математического образа, задание ряда рычагов — "*мышц*", которые способны деформировать или видоизменять объект в соответствии с формируемыми извне командами, использующими коэффициент сжатия, позволяют имитировать любое, в том числе непредсказуемое, движение объекта. (Ниже будет отмечена необходимость механизмов перемещения и вращения.)

Становятся видными новые возможности для создания кинофильмов, сценических постановок и других видеосценариев.

Построение реальных объектов требует имитации объемной, *трехмерной памяти* и введения основных процедур ее преобразования.

Заметим, что проблемы компьютерной реализации "живого" моделирования целиком включают в себя проблемы натурального моделирования, оставляя второму лишь проблемы химического, технологического и электротехнического (возврат к релейно-контактным схемам?) свойства, что, как сказано выше, сейчас успешно решается.

15.2. Моделирование трехмерной памяти

Поскольку компьютер, реализующий "живую" модель, несомненно, должен обладать рядом свойств, обнаруживаемых в процессе дальнейшего изложения, справедливо назвать его видеонейрокомпьютером (ВНК). Это тем более справедливо, что обработка большого числа ячеек памяти предъявляет те же требования к распараллеливанию, что и обработка большого числа нейронов сети.

Отметим, что современные персональные компьютеры (PC) и рабочие станции, реализующие системы мультимедиа, вполне способны выполнять функции такого *нейрокомпьютера*. Однако потребность интеграции памяти нескольких компьютеров может потребовать применения локальной вычислительной сети, удовлетворяющей

требованиям работы в реальном времени. И уж без сомнения, *параллельная вычислительная система SPMD-архитектуры* [9] с большой оперативной памятью в наибольшей степени адекватна рассматриваемой задаче.

Назовем ячейку, занятую элементом объекта, клеткой этого объекта. Как видно далее, клетка минимально должна содержать информацию о цвете (пиксель), коэффициенте освещенности, указателе вершины стека.

Пусть адресное пространство ВНК при сквозной линейной адресации ячеек определяется N разрядами. Адрес разбивается на четыре части (рис. 15.1).

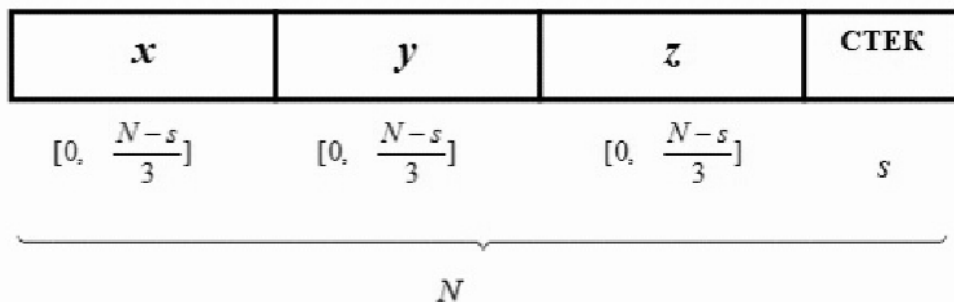


Рис. 15.1. Преобразование линейного адреса в трехмерный

Здесь x , y , z — координаты клетки в *трехмерной памяти*; s разрядов отводятся для адресации стека, сопровождающего каждую клетку. Стек необходим для сохранения информации в случае сжатия объекта — для более точного его восстановления. Таким образом, адресное пространство хранения клеток объекта или целой сцены составляет 2^{N-s} , что говорит о необходимом весьма большом значении N . Значение s практически может быть не более трех.

Трехмерная память — это перспективная разработка. Она предполагает непосредственный (невиртуальный) доступ пользователя на этапе программирования как к памяти, так и к адресам данных при указанной структуре адреса.

При моделировании *трехмерная память* в виртуальной памяти современного компьютера задается трехмерным массивом $S[0:P;$

$0:P; 0:P]$ переменных $S[i, j, k]$, где значения индексов являются координатами соответственно x, y, z . Тогда задача деформации или перемещения объектов преобразуется в задачу нахождения новых значений этих индексов для каждой переменной, являющейся клеткой объекта — для выполнения соответствующего переноса.

При этом линейный адрес переноса или текущего нахождения клетки объекта находится так:

$$\langle S[i, j, k] \rangle = C + i + jP + kP^2.$$

Здесь C — базовый адрес массива.

Стек моделировать нецелесообразно. Информация, потерянная при сжатии объекта, может быть восстановлена с помощью интерполяции.

15.3. "Мышца" и ее сокращение

Пусть объект в трехмерном пространстве задан своей оболочкой. Каждый элемент оболочки является клеткой. Оболочка формируется при создании объекта. В общем случае объект может быть заполнен значащей информацией и внутри, если его вид в разрезе интересует пользователя.

"Мышца" (далее кавычки опустим) как рычаг управления деформацией объекта задается пользователем-разработчиком в центральной системе координат координатами начала и конца, а также точкой неподвижности, относительно которой возможно сокращение *мышцы*. Точка неподвижности может совпадать с началом или концом *мышцы*. (Конечно указанные точки обозначаются мышью; координаты при этом выставляются автоматически.) По умолчанию точка неподвижности является серединой *мышцы*. *Мышца* не обязательно связывает точки оболочки. В общем случае она может располагаться внутри и даже вне объекта, принадлежа ему и влияя на его деформацию.

Мышцы объекта являются невидимыми, входящими в список *мышц* данного объекта и допускающими реакцию на приказы извне.

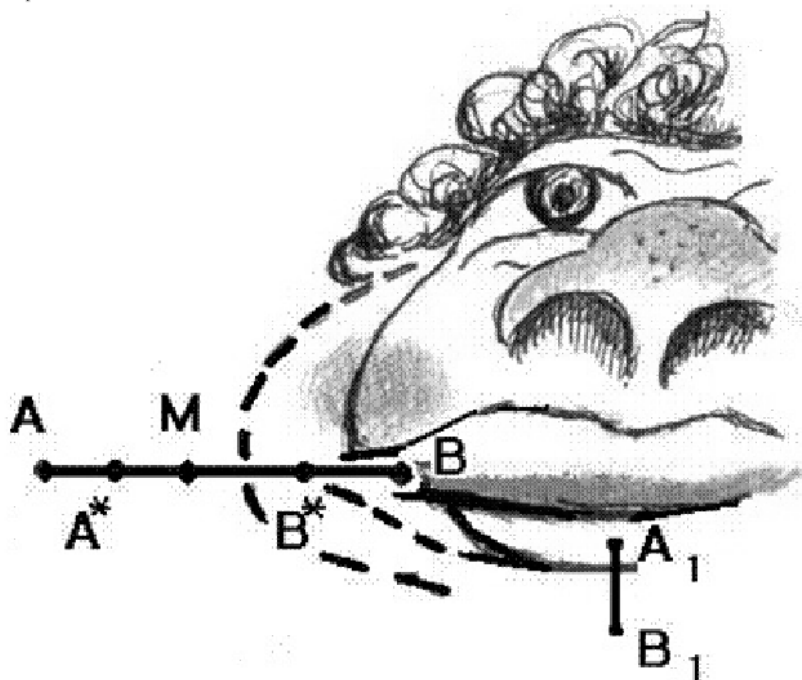


Рис. 15.2. Применение мышцы

Таким образом, *мышца* (рис. 15.2) — это невидимый отрезок AB , связывающий точки $A(x_1, y_1, z_1)$ и $B(x_2, y_2, z_2)$, отображающий некоторое геометрическое место клеток объекта. Этот отрезок связан с объектом и может сокращаться при подаче управляющих импульсов и восстанавливаться при их отсутствии. Такое изменение отрезка производится относительно некоторой принадлежащей ему точки $M(x_M, y_M, z_M)$. Сокращение *мышцы* сопровождается пространственным переносом (уплотнением) клеток объекта, которые принадлежат (с точностью до дискретности адресации) отрезку, обозначающему эту *мышцу*.

Параметрическая система уравнений, описывающих отрезок AB , имеет вид

$$\begin{aligned} x &= k(x_2 - x_1) + x_1 \\ y &= k(y_2 - y_1) + y_1 \\ z &= k(z_2 - z_1) + z_1 \\ 0 &< k < 1. \end{aligned} \quad (15.1)$$

Напоминаем, что координаты точек объекта совпадают с адресами памяти, по которым находятся соответствующие клетки, так что систему уравнений (15.1) с точностью до целочисленной дискретности адресации можно рассматривать относительно *трехмерных адресов*. Это и подразумевается в дальнейших выкладках.

Пусть l — коэффициент сжатия *мышцы*. Тогда отрезок AM необходимо преобразовать в отрезок A^*M с длиной $l \times |A^*M|$, а отрезок MB — в отрезок MB^* с длиной $l \times |MB^*|$.

Рассматривая по каждой координате отдельно, убеждаемся, что отрезок $x_1 x_M$ необходимо преобразовать в отрезок $x_1^* x_M$ с длиной $l(x_M - x_1)$, отрезок $x_M x_2$ необходимо преобразовать в отрезок $x_M x_2^*$ с длиной $l(x_2 - x_M)$. Аналогично отрезок $y_1 y_M$ должен быть преобразован в отрезок $y_1^* y_M$ с длиной $l(y_M - y_1)$ и т.д.

Это, в частности, означает, что клетка с координатой x_1 должна быть переслана в ячейку, определяемую соответствующей составляющей адреса $l(x_M - x_1)$ и т.д.

Однако изменение адреса пересылки клеток по разным координатам различно, что определяется положением *мышцы* в пространстве. Для пояснения этого положения и для формирования алгоритма сжатия *мышцы* рассмотрим пример.

Пусть 045 451 714 — *трехмерный адрес* точки A в восьмеричной системе счисления. Это означает, что $A = (x_1, y_1, z_1) = (045, 451, 714)$. Аналогично, пусть $B = (x_2, y_2, z_2) = (055, 471, 720)$, а неподвижная точка находится в середине отрезка AB , $M = (x_M, y_M, z_M) = (051, 461, 716)$. Зададим $l = 0,6_{10}$ (в десятичной системе счисления).

Адресное расстояние между точками A и M составляет $\Delta A_x = 4$, $\Delta A_y = 10 = 8_{10}$, $\Delta A_z = 2$. Максимальное адресное расстояние по всем координатам $\Delta A = 8_{10}$.

Тогда очевидно, что если по y , определяя точку переноса, следует шагать с переадресацией, определяемой коэффициентом сжатия, то при том же числе переносов по x следует производить переадресацию в два раза реже, а по z — в четыре раза. Тогда по некоторым координатным адресам переноса возможна неоднократная запись. Однако, поскольку происходит сокращение и диапазона изменения координаты y , по некоторым адресам переноса возможна неоднократная запись. Здесь и приходит на помощь стек, позволяющий сохранить информацию для последующего восстановления (расслабления) *мышцы*.

Пусть i — параметр цикла переноса точек отрезка AM для его преобразования в отрезок A^*M , $i = 1, \dots, 11$. Здесь граничное значение цикла определяется максимальным значением точек переноса по всем координатам, равным $\Delta A + 1$. Программно перенос производится с конца отрезка к его началу — чтобы не уничтожить еще не перенесенную информацию. Тогда адреса переноса отдельно по координатам находятся как

$$\begin{aligned} x_{\text{пер}} &= x_M - il \frac{\Delta A_x}{\Delta A} \\ y_{\text{пер}} &= y_M - il \frac{\Delta A_y}{\Delta A} \\ z_{\text{пер}} &= z_M - il \frac{\Delta A_z}{\Delta A}. \end{aligned} \quad (15.2)$$

В указанном порядке рассчитаем адреса переноса отдельно по координате x , не допуская при округлении выхода адресов из адресного пространства *мышцы*:

$$\begin{aligned} 051 - 1 \times 0,6_{10} \times 0,5_{10} &= 051 \\ 051 - 2 \times 0,6_{10} \times 0,5_{10} &= 050 \\ 051 - 3 \times 0,6_{10} \times 0,5_{10} &= 050 \\ 051 - 4 \times 0,6_{10} \times 0,5_{10} &= 050 \\ 051 - 5 \times 0,6_{10} \times 0,5_{10} &= 047 \\ 051 - 6 \times 0,6_{10} \times 0,5_{10} &= 047 \\ 051 - 7 \times 0,6_{10} \times 0,5_{10} &= 046 \\ 051 - 10 \times 0,6_{10} \times 0,5_{10} &= 046 \end{aligned}$$

$$051 - 11 \times 0,6_{10} \times 0,5_{10} = 046$$

Рассчитаем координату y :

$$461 - 1 \times 0,6_{10} \times 1 = 460$$

$$461 - 2 \times 0,6_{10} \times 1 = 460$$

$$461 - 3 \times 0,6_{10} \times 1 = 457$$

$$461 - 4 \times 0,6_{10} \times 1 = 456$$

$$461 - 5 \times 0,6_{10} \times 1 = 456$$

$$461 - 6 \times 0,6_{10} \times 1 = 455$$

$$461 - 7 \times 0,6_{10} \times 1 = 454$$

$$461 - 10 \times 0,6_{10} \times 1 = 453$$

$$461 - 11 \times 0,6_{10} \times 1 = 452$$

Адреса переноса по координате z :

$$716 - 1 \times 0,6_{10} \times 0,25_{10} = 716$$

$$716 - 2 \times 0,6_{10} \times 0,25_{10} = 716$$

$$716 - 3 \times 0,6_{10} \times 0,25_{10} = 716$$

$$716 - 4 \times 0,6_{10} \times 0,25_{10} = 716$$

$$716 - 5 \times 0,6_{10} \times 0,25_{10} = 715$$

$$716 - 6 \times 0,6_{10} \times 0,25_{10} = 715$$

$$716 - 7 \times 0,6_{10} \times 0,25_{10} = 715$$

$$716 - 10 \times 0,6_{10} \times 0,25_{10} = 715$$

$$716 - 11 \times 0,6_{10} \times 0,25_{10} = 715$$

Найдены адреса, по которым необходимо осуществить пересылку при сокращении левой части *мышцы*.

Аналогично, для правой части *мышцы*, для отрезка МВ, найдем адресные расстояния $\Delta A_x = -4$, $\Delta A_y = -10 = -8_{10}$, $\Delta A_z = -2$, $\Delta A = -8_{10}$. По (15.2) для адресов переноса по x , y , z находим

$$051 + 1 \times 0,6_{10} \times 0,5_{10} = 051 \quad 461 + 1 \times 0,6_{10} \times 1 = 462 \quad 716 + 1 \times 0,6_{10}$$

$$051 + 2 \times 0,6_{10} \times 0,5_{10} = 052 \quad 461 + 2 \times 0,6_{10} \times 1 = 463 \quad 716 + 2 \times 0,6_{10}$$

$051 + 3 \times 0,6_{10} \times 0,5_{10} = 052$	$461 + 3 \times 0,6_{10} \times 1 = 463$	$716 + 3 \times 0,6_{10} \times 1 = 716$
$051 + 4 \times 0,6_{10} \times 0,5_{10} = 052$	$461 + 4 \times 0,6_{10} \times 1 = 464$	$716 + 4 \times 0,6_{10} \times 1 = 716$
$051 + 5 \times 0,6_{10} \times 0,5_{10} = 053$	$461 + 5 \times 0,6_{10} \times 1 = 464$	$716 + 5 \times 0,6_{10} \times 1 = 716$
$051 + 6 \times 0,6_{10} \times 0,5_{10} = 053$	$461 + 6 \times 0,6_{10} \times 1 = 465$	$716 + 6 \times 0,6_{10} \times 1 = 716$
$051 + 7 \times 0,6_{10} \times 0,5_{10} = 053$	$461 + 7 \times 0,6_{10} \times 1 = 465$	$716 + 7 \times 0,6_{10} \times 1 = 716$
$051 + 10 \times 0,6_{10} \times 0,5_{10} = 054$	$461 + 10 \times 0,6_{10} \times 1 = 466$	$716 + 10 \times 0,6_{10} \times 1 = 716$
$051 + 11 \times 0,6_{10} \times 0,5_{10} = 054$	$461 + 11 \times 0,6_{10} \times 1 = 466$	$716 + 11 \times 0,6_{10} \times 1 = 716$

Таким образом, найдены адреса ячеек памяти, куда следует пересылать информацию при сокращении *мышцы*. Необходимо выяснить, какие точки отрезка, отображающего *мышцу*, необходимо пересылать по найденным адресам.

Свяжем перебор искомых точек с тем же параметром цикла организации пересылки

$$x = x_M - i \frac{\Delta A_x}{\Delta A} y = y_M - i \frac{\Delta A_y}{\Delta A} z = z_M - i \frac{\Delta A_z}{\Delta A}. \quad (15.3)$$

Найдем отдельно для x , y , z отрезка AM адреса пересылаемых точек:

$051 - 1 \times 0,5_{10} = 050$	$461 - 1 \times 1 = 460$	$716 - 1 \times 0,25_{10} = 716$
$051 - 2 \times 0,5_{10} = 050$	$461 - 2 \times 1 = 457$	$716 - 2 \times 0,25_{10} = 715$
$051 - 3 \times 0,5_{10} = 047$	$461 - 3 \times 1 = 456$	$716 - 3 \times 0,25_{10} = 715$
$051 - 4 \times 0,5_{10} = 047$	$461 - 4 \times 1 = 455$	$716 - 4 \times 0,25_{10} = 715$
$051 - 5 \times 0,5_{10} = 046$	$461 - 5 \times 1 = 454$	$716 - 5 \times 0,25_{10} = 715$
$051 - 6 \times 0,5_{10} = 046$	$461 - 6 \times 1 = 453$	$716 - 6 \times 0,25_{10} = 714$
$051 - 7 \times 0,5_{10} = 045$	$461 - 7 \times 1 = 452$	$716 - 7 \times 0,25_{10} = 714$
$051 - 10 \times 0,5_{10} = 045$	$461 - 10 \times 1 = 451$	$716 - 10 \times 0,25_{10} = 711$
$051 - 11 \times 0,5_{10} = 045$	$461 - 11 \times 1 = 450$	$716 - 11 \times 0,25_{10} = 711$

Для отрезка MB находим

$051 + 1 \times 0,5_{10} = 052$	$461 + 1 = 462$	$716 + 1 \times 0,25_{10} = 716$
$051 + 2 \times 0,5_{10} = 052$	$461 + 2 = 463$	$716 + 2 \times 0,25_{10} = 716$
$051 + 3 \times 0,5_{10} = 053$	$461 + 3 = 464$	$716 + 3 \times 0,25_{10} = 716$

$051 + 4 \times 0,5_{10} = 053$	$461 + 4 = 465$	$716 + 4 \times 0,25_{10} = 716$
$051 + 5 \times 0,5_{10} = 054$	$461 + 5 = 466$	$716 + 5 \times 0,25_{10} = 716$
$051 + 6 \times 0,5_{10} = 054$	$461 + 6 = 467$	$716 + 6 \times 0,25_{10} = 716$
$051 + 7 \times 0,5_{10} = 055$	$461 + 7 = 470$	$716 + 7 \times 0,25_{10} = 716$
$051 + 10 \times 0,5_{10} = 055$	$461 + 10 = 471$	$716 + 10 \times 0,25_{10} = 7$
$051 + 11 \times 0,5_{10} = 055$	$461 + 11 = 471$	$716 + 11 \times 0,25_{10} = 71$

"Склеим" координатные части адресов и обозначим ее окончательно в двух столбцах: в левом — пересылку для левой части *мышцы*, в правом — для правой.

$(050\ 460\ 716) \Rightarrow 051\ 460\ 716$	$(052\ 462\ 716) \Rightarrow 051\ 462\ 716$
$(050\ 457\ 715) \Rightarrow 050\ 460\ 716$	$(052\ 463\ 717) \Rightarrow 052\ 463\ 716$
$(047\ 456\ 715) \Rightarrow 050\ 457\ 716$	$(053\ 464\ 717) \Rightarrow 052\ 463\ 716$
$(047\ 455\ 715) \Rightarrow 050\ 456\ 716$	$(053\ 465\ 717) \Rightarrow 052\ 464\ 716$
$(046\ 454\ 715) \Rightarrow 047\ 456\ 715$	$(054\ 466\ 717) \Rightarrow 053\ 464\ 717$
$(046\ 453\ 714) \Rightarrow 047\ 455\ 715$	$(054\ 467\ 720) \Rightarrow 053\ 465\ 717$
$(045\ 452\ 714) \Rightarrow 046\ 454\ 715$	$(055\ 470\ 720) \Rightarrow 053\ 465\ 717$
$(045\ 451\ 714) \Rightarrow 046\ 453\ 715$	$(055\ 471\ 720) \Rightarrow 054\ 466\ 717$
$(045\ 450\ 714) \Rightarrow 046\ 452\ 715$	$(055\ 471\ 720) \Rightarrow 054\ 466\ 717$

В строках 2 и 3, в 6 и 7, а также в 8 и 9 видно формирование стеков при неоднократной записи в одну клетку.

Однако вся проделанная выше работа по сжатию *мышцы* в действительности касается только клеток со значащим содержимым. Это означает, что если объект задан своей оболочкой, то переносу подвергаются только клетки оболочки. *Мышца* чаще всего является "пустым" объектом, и смещаются только точки А и В. Переносить ее клетки следует в таком случае лишь тогда, когда она принадлежит оболочке или пересекается с ней.

Сокращение *мышцы* можно использовать для деформации объекта на стадии его создания. После этого *мышца* может быть исключена из списка объекта или переопределена вновь в измененных границах.

Таким образом, каждый перенос клетки *мышцы* или объекта в целом сопровождается анализом на значимость этой клетки.

15.4. Сокращение "твердого" объекта

Рассмотрим взаимодействие *мышцы* с тем объектом, к которому она приложена. Сокращаясь, она должна увлекать клетки объекта или только его оболочку, имитируя сокращение не только *мышцы*, но и всего объекта.

Пусть, как и ранее, задано значение $0 \leq \lambda \leq 1$, приводящее к сокращению объекта в направлении *мышцы* АВ (рис. 15.3). Объект должен преобразиться, как показано на рисунке пунктиром. *Мышца* описывается системой параметрических уравнений (15.1). М, как и прежде, — неподвижная (относительно неподвижного объекта) точка *мышцы*. Однако эта точка теперь определяет плоскость М, перпендикулярную *мышце*, относительно которой слева и справа все значащие клетки объекта должны сместиться в направлении к этой плоскости, т.е. параллельно *мышце*, с коэффициентом смещения (сжатия), равным λ .

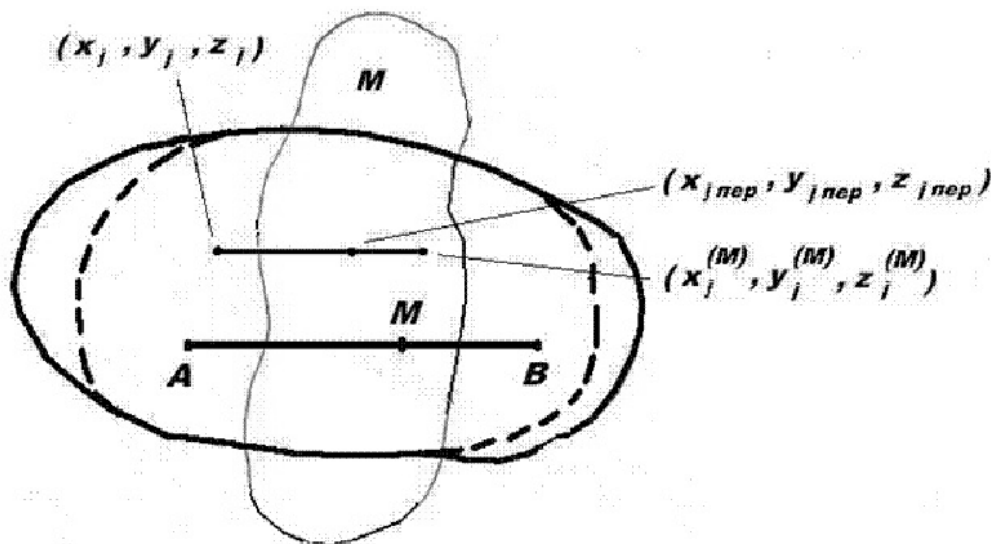


Рис. 15.3. Сокращение твердого тела

Плоскость М описывается уравнением

$$(\xi - x_M)(x_1 - x_M) + (\psi - y_M)(y_1 - y_M) + (\zeta - z_M)(z_1 - z_M) = 0$$

Пусть точка (x_j, y_j, z_j) — текущая клетка объекта с адресом $x_j y_j z_j$, подлежащая указанному переносу. Найдем координаты $x_j^{(M)}, y_j^{(M)}, z_j^{(M)}$ ее проекции на плоскость M . Так как точка с этими координатами принадлежит плоскости M и минимизирует квадрат расстояния точки (x_j, y_j, z_j) до этой плоскости, задача такой условной минимизации формулируется так:

найти значение $x_j^{(M)}, y_j^{(M)}, z_j^{(M)}$, удовлетворяющее требованию

$$(x_j^{(M)} - x_j)^2 + (y_j^{(M)} - y_j)^2 + (z_j^{(M)} - z_j)^2 \rightarrow \min$$

при условии

$$(x_j^{(M)} - x_M)(x_1 - x_M) + (y_j^{(M)} - y_M)(y_1 - y_M) + (z_j^{(M)} - z_M)(z_1 - z_M) = 0.$$

Решим данную задачу, используя уравнение Лагранжа

$$(x_j^{(M)} - x_j)^2 + (y_j^{(M)} - y_j)^2 + (z_j^{(M)} - z_j)^2 + \lambda[(x_j^{(M)} - x_M)(x_1 - x_M) + (y_j^{(M)} - y_M)(y_1 - y_M) + (z_j^{(M)} - z_M)(z_1 - z_M)] \rightarrow \min$$

Дифференцируя по всем переменным, включая λ , и приравняв производные нулю, получим систему линейных уравнений

$$2(x_j^{(M)} - x_j) + \lambda(x_1 - x_M) = 0$$

$$2(y_j^{(M)} - y_j) + \lambda(y_1 - y_M) = 0$$

$$2(z_j^{(M)} - z_j) + \lambda(z_1 - z_M) = 0$$

$$(x_j^{(M)} - x_M)(x_1 - x_M) + (y_j^{(M)} - y_M)(y_1 - y_M) + (z_j^{(M)} - z_M)(z_1 - z_M) = 0.$$

Решая систему, находим

$$x_j^{(M)} = 0,5\lambda(x_M - x_1) + x_j$$

$$y_j^{(M)} = 0,5\lambda(y_M - y_1) + y_j$$

$$z_j^{(M)} = 0,5\lambda(z_M - z_1) + z_j$$

$$\lambda = 2 \frac{(x_j - x_M)(x_1 - x_M) + (y_j - y_M)(y_1 - y_M) + (z_j - z_M)(z_1 - z_M)}{(x_1 - x_M)^2 + (y_1 - y_M)^2 + (z_1 - z_M)^2}.$$

Таким образом, получена *мышца* с началом в точке (клетке) с координатами x_j, y_j, z_j и с концом в точке $x_j^{(M)}, y_j^{(M)}, z_j^{(M)}$. Эта же точка является неподвижной. Тогда по (14.2) и (14.3) имитируется сжатие *мышцы*. Цикл по j охватывает все значащие клетки объекта.

15.5. Сокращение вязкого тела

Вязкое тело, внутри которого сокращается *мышца*, характеризуется "затуханием" величины смещения клетки объекта с увеличением ее расстояния до *мышцы* (рис. 15.4).

Найдем квадрат r^2 кратчайшего расстояния между произвольной текущей точкой объекта или только его оболочки (x_j, y_j, z_j) и мышцей. Это расстояние определяется с помощью проекции (x_{j0}, y_{j0}, z_{j0}) данной точки на *мышцу* (на рисунке — две точки, проектируемые правее A), или ее удаленностью от концов *мышцы* (на рисунке — точка левее A).

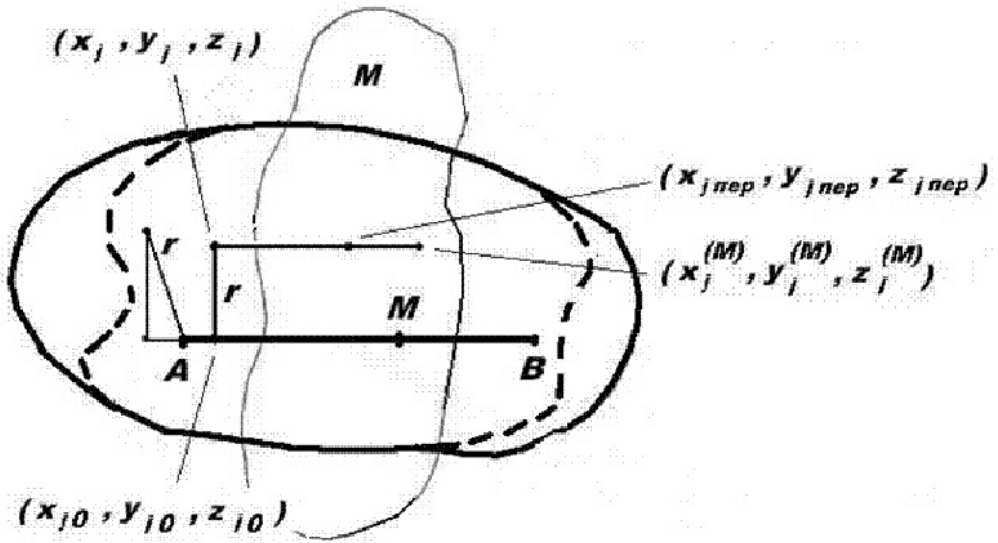


Рис. 15.4. Сокращение вязкого тела

Предполагая, что данная проекция принадлежит *мышце*, найдем с помощью параметрического описания (15.1) соответствующее ей значение k_{min} из соотношения

$$r^2 = (x_{j0} - x_j)^2 + (y_{j0} - y_j)^2 + (z_{j0} - z_j)^2 = (x_{j0} - x_1 - k(x_2 - x_1))^2 + (y_{j0} - y_1 - k(y_2 - y_1))^2 + (z_{j0} - z_1 - k(z_2 - z_1))^2.$$

Значение k_{min} , которое определяет клетку *мышцы*, соответствующую искомому минимальному расстоянию, вычисляется с помощью нулевого значения производной по k :

$$(x_{j0} - x_1 - k(x_2 - x_1))(x_2 - x_1) + (y_{j0} - y_1 - k(y_2 - y_1))(y_2 - y_1) + (z_{j0} - z_1 - k(z_2 - z_1))(z_2 - z_1) = 0$$

Откуда

$$k_{min} = \frac{(x_{j0} - x_1)(x_2 - x_1) + (y_{j0} - y_1)(y_2 - y_1) + (z_{j0} - z_1)(z_2 - z_1)}{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}.$$

Если выполняется условие $0 \leq k_{\min} \leq 1$, искомая проекция действительно принадлежит *мышце*. Тогда r^2 отыскивается в результате подстановки (15.7) в (15.6). Если $k_{\min} < 0$, ближайшей точкой *мышцы* является А. В этом случае

$$r^2 = (x_1 - x_j)^2 + (y_1 - y_j)^2 + (z_1 - z_j)^2.$$

Если $k_{\min} > 1$, ближайшей точкой *мышцы* является В:

$$r^2 = (x_2 - x_j)^2 + (y_2 - y_j)^2 + (z_2 - z_j)^2.$$

Чтобы определить направление переноса клетки (x_j, y_j, z_j) , необходимо по (15.5) найти ее проекцию на плоскость М. Тогда для нахождения адресов циклической пересылки значащих клеток объекта воспользуемся формулами

$$\begin{aligned} x_{j\text{пер}} &= x_{jM} - il\rho \left(1 - \frac{r^2}{R^2} \right) \frac{\Delta A_x}{\Delta A} \\ y_{j\text{пер}} &= y_{jM} - il\rho \left(1 - \frac{r^2}{R^2} \right) \frac{\Delta A_y}{\Delta A} \\ z_{j\text{пер}} &= z_{jM} - il\rho \left(1 - \frac{r^2}{R^2} \right) \frac{\Delta A_z}{\Delta A}. \end{aligned} \quad (15.8)$$

Здесь ρ — коэффициент вязкости и R^2 — заведомо большое число, ограничивающее значение r^2 , где смещение полностью затухает.

Перебор клеток объекта для пересылки выполняется по (15.3). Пересылка выполняется для обоих полупространств, разделяемых плоскостью М.

15.6. Сокращение вязкого тела с "выпучиванием"

Сокращение вязкой и упругой среды, такой как резина, губка, упитанные щеки и др., обычно сопровождается явлением, которое можно назвать

попыткой сохранения объема в результате *выпучивания*. При сокращении *мышцы* это означает, что перенос клеток объекта вдоль *мышцы* обретает перпендикулярную составляющую, тем меньшую, чем больше расстояние до *мышцы*.

Пусть, как и прежде, x_j, y_j, z_j — составляющие адрес координаты текущей клетки *вязкого объекта*, для которого выше сформированы выражения (15.8) для определения адреса переноса. Дополним их указанной перпендикулярной составляющей (рис. 15.5).

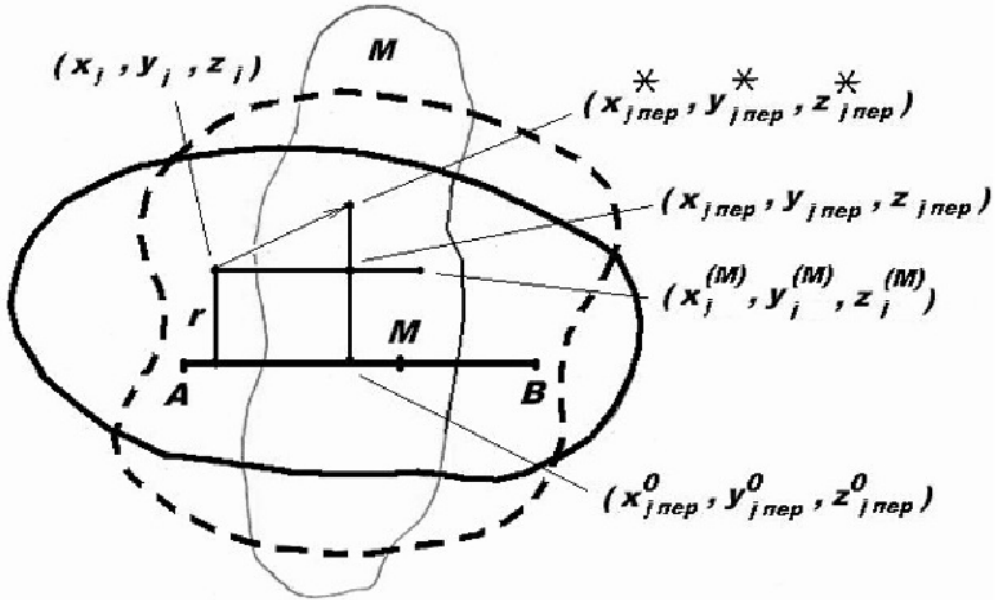


Рис. 15.5. Сокращение тела с "выпучиванием"

По (15.7), подставив вместо x_{j0}, y_{j0}, z_{j0} значение найденной по (15.8) точки переноса $(x_{j пер}, y_{j пер}, z_{j пер})$, найдем k_{min} и, не смотря на его значение, найдем проекцию $(x_{j пер}^0, y_{j пер}^0, z_{j пер}^0)$ на *мышцу* или ее продолжение влево или вправо.

(Заметим, что уравнения (1) при $0 \leq k \leq 1$ определяют собственно *мышцу*, другие значения $k \in [-\infty, 0], k \in [0, \infty]$ определяют точки прямой, которой принадлежит отрезок — *мышца*.)

Запишем систему параметрических уравнений прямой, проходящей через точки $(x_{j \text{ пер}}, y_{j \text{ пер}}, z_{j \text{ пер}})$ и $(x_{j \text{ пер}}^0, y_{j \text{ пер}}^0, z_{j \text{ пер}}^0)$:

$$\begin{aligned} x &= m(x_{j \text{ пер}}^0 - x_{j \text{ пер}}) + x_{j \text{ пер}} \\ y &= m(y_{j \text{ пер}}^0 - y_{j \text{ пер}}) + y_{j \text{ пер}} \\ z &= m(z_{j \text{ пер}}^0 - z_{j \text{ пер}}) + z_{j \text{ пер}}. \end{aligned} \quad (15.9)$$

Задав малое приращение $\Delta m > 0$, зависящее от $r_{\text{пер}}^2 = (x_{j \text{ пер}}^0 - x_{j \text{ пер}})^2 + (y_{j \text{ пер}}^0 - y_{j \text{ пер}})^2 + (z_{j \text{ пер}}^0 - z_{j \text{ пер}})^2$, и приняв $m = 1 + \Delta m$, получим окончательно точку переноса $(x_{j \text{ пер}}^*, y_{j \text{ пер}}^*, z_{j \text{ пер}}^*)$, отстоящую от точки переноса, найденной по (15.8), и смещенную в перпендикулярном направлении от *мышцы*.

Эмпирическую зависимость коэффициента смещения Δm от указанного выше расстояния можно выбрать как

$$\Delta m = l\rho \left(1 - \frac{r_{\text{пер}}^2}{R^2} \right). \quad (15.10)$$

Тогда из (9) находим

$$\begin{aligned} x_{j \text{ пер}}^* &= (1 + \Delta m)(x_{j \text{ пер}}^0 - x_{j \text{ пер}}) + x_{j \text{ пер}} \\ y_{j \text{ пер}}^* &= (1 + \Delta m)(y_{j \text{ пер}}^0 - y_{j \text{ пер}}) + y_{j \text{ пер}} \\ z_{j \text{ пер}}^* &= (1 + \Delta m)(z_{j \text{ пер}}^0 - z_{j \text{ пер}}) + z_{j \text{ пер}}. \end{aligned} \quad (15.11)$$

Таким образом, сокращая *мышцу* АВ, показанную на рис. 15.2, можно, например, изобразить то подобие улыбки, которая обозначена пунктиром. Введение других *мышц*, например, A_1B_1 под нижней губой, может значительно усилить правдоподобие улыбки.

15.7. Шарнирно-мышечное соединение

Координаты шарнира задаются условно (ячейка не занимается) точкой $O(x_0, y_0, z_0)$ внутри объекта (рис. 15.6). Мышца АВ связывает лучи (элементы скелета), исходящие из точки O . Сокращение мышцы должно вызывать видимость движений, характерных при ходьбе, движении рук и т.д. Это требует таких деформаций объектов, при которых его клетки, облегающие эти лучи, или только клетки оболочки, несущие в себе данные лучи, сближаются вместе с лучами, не приводя к дополнительной деформации.

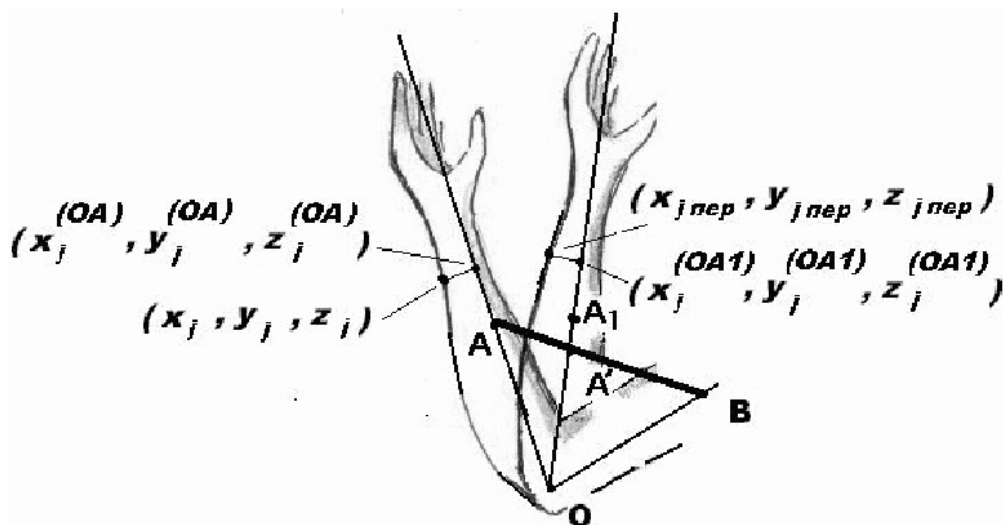


Рис. 15.6. Деформация объекта вокруг шарнира

Практический интерес представляет случай, когда неподвижная точка мышцы совпадает с одним из ее концов. Это соответствует направлению движения конечности относительно неподвижного тела. (Хотя следование известным соотношениям о количестве движения или третьего закона механики не представляет трудностей.)

По аналогии с (15.1), луч OA описывается параметрической системой уравнений

$$\begin{aligned} x &= k(x_1 - x_0) + x_0 \\ y &= k(y_1 - y_0) + y_0 \\ z &= k(z_1 - z_0) + z_0. \end{aligned} \quad (15.12)$$

Если $0 \leq k \leq 1$, то уравнения отображают отрезок OA . Нас же интересует вся прямая, исходящая из точки O .

Отобразим сокращение *мышцы* переносом клетки A в клетку A' (x_1', y_1', z_1') , в упрощенной форме воспользовавшись (15.2) и (15.3). Этим установим новое положение OA' луча OA .

Уравнения этого луча имеют вид

$$\begin{aligned} x &= k'(x_1' - x_0) + x_0 \\ y &= k'(y_1' - y_0) + y_0 \\ z &= k'(z_1' - z_0) + z_0. \end{aligned} \quad (15.13)$$

Если $0 \leq k' \leq 1$, то уравнения отображают отрезок OA' . Однако точка A' не отображает факт вращения луча OA вокруг точки O . При этом вращении точка A должна переместиться в точку A_1 , отстоящую от O на расстояние, равное длине отрезка OA . Найдем координаты этой точки на луче OA' из соотношения

$$(x_{11} - x_0)^2 + (y_{11} - y_0)^2 + (z_{11} - z_0)^2 = r^2 = (x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2.$$

В параметрической форме представления (12) точке (x_{11}, y_{11}, z_{11}) соответствует некоторое значение k_1' параметра. Подставим вместо координат этой точки их параметрическое представление:

$$(k_1')^2 ((x_1' - x_0)^2 + (y_1' - y_0)^2 + (z_1' - z_0)^2) = r^2.$$

Откуда

$$k_1' = \frac{r}{\sqrt{(x_1' - x_0)^2 + (y_1' - y_0)^2 + (z_1' - z_0)^2}} \quad (15.14)$$

Очевидно, $k_1' > 0$. Таким образом, точка A_1 переноса конца *мышцы* найдена.

Для синхронных действий при переносе всех клеток объекта, подлежащих перемещению вокруг шарнира, запишем уравнения луча OA_1 в зависимости от того же параметра k , определяющего положение (15.12) луча OA

$$\begin{aligned}x &= k(x_{11} - x_0) + x_0 \\y &= k(y_{11} - y_0) + y_0 \\z &= k(z_{11} - z_0) + z_0.\end{aligned}\quad (15.15)$$

Пусть (x_j, y_j, z_j) — точка (клетка), подлежащая переносу, в соответствии с сокращением *мышцы*, в точку с тем же радиусом относительно точки O , но с исключением других деформаций объекта. Чтобы найти координаты $(x_j^{(OA)}, y_j^{(OA)}, z_j^{(OA)})$ проекции этой точки на луч OA , решим задачу минимизации расстояния между этой точкой и лучом OA , описанным системой (15.12):

$$(x_j - k(x_1 - x_0) - x_0)^2 + (y_j - k(y_1 - y_0) - y_0)^2 + (z_j - k(z_1 - z_0) - z_0)^2 \rightarrow \min.$$

Приравняв производную по k нулю, получим

$$(x_j - k(x_1 - x_0) - x_0)(x_1 - x_0) + (y_j - k(y_1 - y_0) - y_0)(y_1 - y_0) + (z_j - k(z_1 - z_0) - z_0)(z_1 - z_0) = 0.$$

Отсюда

$$k_{(min)} = \frac{(x_j - x_0)(x_1 - x_0) + (y_j - y_0)(y_1 - y_0) + (z_j - z_0)(z_1 - z_0)}{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}.$$

Подставим (15.16) в (15.12) и найдем координаты $(x_j^{(OA)}, y_j^{(OA)}, z_j^{(OA)})$ искомой проекции.

Вычислим разности координат точки (x_j, y_j, z_j) и найденной ее проекции на луч OA , определяющие смещение этой точки:

$$\begin{aligned}\Delta x_j &= x_j - x_j^{(OA)} \\ \Delta y_j &= y_j - y_j^{(OA)} \\ \Delta z_j &= z_j - z_j^{(OA)}\end{aligned} \quad (15.17)$$

Подстановкой (15.15) в (15.14) найдем координаты $(x_j^{(OA1)}, y_j^{(OA1)}, z_j^{(OA1)})$ "образа" точки $(x_j^{(OA)}, y_j^{(OA)}, z_j^{(OA)})$ при повороте луча. Тогда координаты $x_{j \text{ пер}}, y_{j \text{ пер}}, z_{j \text{ пер}}$ переноса точки (x_j, y_j, z_j) отыскиваются так:

$$\begin{aligned}x_{j \text{ пер}} &= x_j^{(OA1)} + \Delta x_j \\ y_{j \text{ пер}} &= y_j^{(OA1)} + \Delta y_j \\ z_{j \text{ пер}} &= z_j^{(OA1)} + \Delta z_j.\end{aligned} \quad (15.18)$$

Организовав цикл по j - по всем точкам объекта (или оболочки) в окрестности луча OA , выполним требуемую имитацию движения.

15.8. Перемещение, поворот и вращение

До сих пор рассматривалось поведение объекта в единственной системе координат. Взаимодействие многих объектов требует наличия центральной системы координат, в которой происходят все перемещения объектов относительно друг друга, и систем координат, связанных с каждым объектом — объектовых систем координат. Объект создается в его объектовой системе координат. Его движения "относительно себя" — сокращение *мышц*, повороты, вращение — удобнее наблюдать в связанной с ним объектовой системе координат. Таким образом, центральная система координат должна быть связана со многими объектовыми системами возможностями пересчета. Объектовые системы не занимают неизменное место в ней, а могут перемещаться и поворачиваться вместе с объектами.

Это означает, что каждая n -я объектовая система координат связана с центральной системой матрицей пересчета:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos(x, x_n) & \cos(x, y_n) & \cos(x, z_n) \\ \cos(y, x_n) & \cos(y, y_n) & \cos(y, z_n) \\ \cos(z, x_n) & \cos(z, y_n) & \cos(z, z_n) \end{pmatrix} \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} + \begin{pmatrix} x_{n0} \\ y_{n0} \\ z_{n0} \end{pmatrix}$$

Здесь элементы матрицы — направляющие косинусы углов между осями, x_{n0} , y_{n0} , z_{n0} — координаты начала объектовой системы. Все указанные величины являются переменными. Только плавно изменяя их значения, можно добиться перемещения, поворота или вращения объекта среди других объектов общей сцены, связанной с центральной системой координат.

Отсюда важная рекомендация: чтобы имитировать относительное движение объектов, необходимо плавно, в реальном времени, давать приращения *направляющим косинусам* и смещениям так, чтобы сохранялись основные соотношения, связывающие декартовы системы координат. А именно, сумма квадратов элементов строк и столбцов должна быть равна единице.

15.9. Внешнее "механическое" воздействие на объект

Такое воздействие необходимо не только при определенном взаимодействии различных объектов, но и на уровне формирования, "ваяния" объекта. Идея такого ваяния заключается в следующем.

Пусть в память введена некоторая заготовка, имеющая наиболее близкую форму для желаемого объекта. (Например шар — для формирования головы.) Приближая к заготовке объект-"инструмент", например молоток, необходимо потребовать, чтобы при угрозе прикосновения к объекту, ближайшие его клетки отступали, и оболочка, продавливаясь, принимала форму проникающего инструмента. Эта операция напоминает ковку или штамповку. Таким многократным воздействием с разных сторон можно добиться любой формы объекта.

Можно допустить не только вдавливание инструмента в объект, но и

вытягивания близлежащей области, подобно нарыву. В этом случае после соприкосновения с объектом следует отводить инструмент от него. Близкая область оболочки должна вытягиваться вслед за инструментом.

Если не деформировать оболочку, а позволить инструменту проникать в объект, то подобное действие сравнимо с действием ножа. Таким образом от объекта могут отсекаться части, — для последующего уничтожения или для создания новых объектов.

Роль подобных инструментов могут играть объекты друг для друга. Это позволит, например, имитировать некоторые разрушительные события.

15.10. Построение объектов в трехмерной памяти

Для построения объектов целесообразно воспользоваться тремя координатными проекциями, подобно 3D-MAX. Сначала в одной из проекций рисуется пока неточный, предполагаемый разрез объекта. На других проекциях автоматически сформируются отрезки — боковые проекции этого разреза. На разрезе определяется точка для дальнейшей деформации объекта в двух других проекциях. Проекция этой точки высвечивается на двух других проекциях.

Подведя мышь к одной из проекций точки, следует повести ее (мышь) вверх или вниз, формируя выпуклость вслед за этим движением так, чтобы первоначально заданная плоская поверхность образовала выпуклость. Так сформируется объемное изображение. Если при этом придерживать клавишу control (или другим способом), плоская поверхность сохранится для замкнутости объема или для последующей аналогичной деформации в ту или другую сторону.

Повороты получившейся объемной фигуры позволяют производить указанным способом различные деформации.

Такой способ формирования объектов может быть признан основным. Однако выше указывалось на возможность "ваяния" с помощью временно создаваемых *мышц* и инструментов внешнего воздействия или проникновения.

15.11. Командно-программное управление объектами

Необходимо помнить, что объекты в *трехмерной памяти* создаются для их визуального восприятия. Система визуализации может базироваться на проецировании видимой поверхности всех объектов, включая фон, на плоскость $z = 0$ в центральной системе координат. Эта плоскость и представляет экран. Значит, каждая прямая, исходящая из точки $x_i, y_i, 0$ и перпендикулярная экрану, продолжается (здесь рассматриваем только прозрачную среду) до первого пересечения с одним из объектов или с фоном. Полученная точка обеспечивает изображение в данной текущей точке экрана.

Для получения различных срезов (подобно томографии) экран может формироваться и в других плоскостях вида $z = a > 0$.

Для получения различных ракурсов центральная система координат должна быть преобразована.

Введение *мышц, шарниров*, объектовых систем координат позволяет программировать действия объектов и их взаимное расположение. Программа создается на базе командного языка, содержащего команды вида

<сократить мышцу m с $l = 0,8$ >;
<повернуть объект P по матрице S >;
<приблизить объект P >;
<перенести объект P >;
<показать срез $z = 5$ > и т.д.

Таким образом, могут планироваться и исполняться сцены, основанные на имитации движения объектов, их мимики, преобразования и т.д.

Как каждая управляющая система, работа рассматриваемой системы тактируется. Поскольку видео — основное ее назначение, то тактовую частоту целесообразно совместить с частотой следования кадров. (Если хватит производительности! Иначе придется получать промежуточные кадры путем интерполяции.) Тогда программа ориентируется на имитацию движения в отдельном кадре в системе реального времени. Это значит, что если дан приказ сократить *мышцу* в 0,6 раза, то

движение растягивается во времени на несколько кадров. Например, в отдельном кадре *мышца* сокращается лишь в 0,98 раз. Тогда в течение нескольких кадров необходимое сокращение будет достигнуто. По-видимому, необходимо задавать то количество кадров, в течение показа которых движение должно быть выполнено.

Эмпирические зависимости для такого дробления целесообразно получить экспериментально.

Воспроизводя принципы живых организмов, необходимо учитывать, что *мышца* сокращается, пока и поскольку поступают импульсы — приказы на ее сокращение. Отсутствие таких импульсов служит расслаблению, восстановлению невозбужденной *мышцы*. Так, улыбка существует, пока существует ее причина. Отсутствие причин приводит к индифферентности. Это следует учитывать при программировании поведения объектов.

Программа управления объектами, как и каждая компьютерная программа, должна предусматривать альтернативные действия в случае выполнения предусмотренных условий, т.е. содержать команды условного перехода.

15.12. Логические нейронные сети в основе управления трехмерными компьютерными объектами

Занимаясь "живым" моделированием на основе компьютерных и информационных технологий, следует быть приверженным точным наукам, в которых любые построения начинаются с исчерпывающих, непротиворечивых определений.

Что есть жизнь? Какое существо можно назвать живым?

1. Живое существо адекватно реагирует на внешние раздражители.
2. Адекватность реакции живого существа подчиняется критерию "хорошо — плохо".

В качестве комментариев отметим, что даже внутренние болезни имеют внешние причины: экологические, политические, наследственные,

бактериологические и т.д. В то же время философски уместно говорить о жизни существа не в абсолютном аспекте, а в аспекте моделирования: вся жизнь представляет собой *игровую модель*, а задачей любого моделирования является минимизация целевой функции, в конечном итоге отделяющей понятие "хорошо" от понятия "плохо". К сожалению, для каждого индивидуума эти понятия различны и на нейросетевом (мозговом) уровне обусловлены обучением — генетическим или практическим.

Посчитаем достаточными положения 1 и 2 для настоящего уровня решения задачи "живого" моделирования. Однако мы понимаем, что достичь такого уровня обучения, когда система сама начинает осознавать, что хорошо, а что плохо, на основе индуктивного и дедуктивного мышления, вряд ли сейчас возможно практически. Мы вынуждены решать задачу-минимум, требуя, в основном для культурно-развлекательных целей, адекватности реакции. Степень этой адекватности в соответствии с означенным критерием мы уж предусмотрим сами.

Например, можно научить объект улыбаться, если он видит картинку с преобладанием розовых тонов, но гневаться, если на картинке превалирует коричневый цвет и т.д.

Таким образом, ближайшей практической задачей является закладка инструментальных основ. Позволим проблеме развиваться на базе нейросетевых технологий более глобально в дальнейшем.

Пусть сотворенный объект (рис. 15.7) в компьютерном или материальном воплощении наделен некоторым фоновым поведением, реализуемым программно. Например, такое поведение может быть аналогичным поведению зверя в клетке, нервно дефилирующего вдоль решетки. Целесообразно в программе использовать и элементы случайности на основе ДСЧ — датчика случайных чисел. Это может ввести разнообразие в фоновое поведение объекта: внезапный поворот головы, смена направления движения и т.д. Такое поведение уже само по себе является занимательным для зрителя.

Однако представим себе, что фоновая программа предусматривает периодическое прерывание для включения "внимания" объекта к внешней обстановке — к "картинке", циклически вводимой в его память

с помощью видеоввода.

Включается программа анализа "картинки", питающая *рецепторный слой* нейросети. Алгоритмы такого анализа, как подготовительного этапа решения задачи распознавания, представляют собой отдельное направление исследований. На этом этапе "картинка" отображается в *пространстве признаков*, обоснованных для распознавания и принятия решений. На самом простом практическом уровне может производиться подсчет различных оттенков цветов в элементарном квадрате "картинки", обнаружение резких цветовых границ, наличие линий, кругов определенного цвета и т.д. Такие идеи реагирования на цвет рассматриваются в лекции 16.



Рис. 15.7. Возможный объект натурального моделирования

Ограничиваясь только цветом, можно, в терминах алгебры высказываний, представить следующий пример алгоритма реакции объекта.

- "В квадрате i, j преобладает зеленый цвет" \wedge "отсутствуют резкие перемены"
-> "Запустить программу умиротворения";
- "В квадрате i, j преобладает зеленый цвет" \wedge "наблюдаются резкие перемены"
-> "Запустить программу тоски по лесным далям";
- "В квадрате i, j преобладает голубой цвет" \wedge "наблюдаются темные пятна"
-> "Запустить программу приветствия";
- "В квадрате i, j преобладает красный цвет" ->
-> "Запустить программу повторного обзора квадрата" \wedge "Запустить программу приветствия";
- "В квадрате i, j обнаружен круг с преобладанием телесного цвета" ->
-> "Запустить программу приветствия";
- "В квадрате i, j преобладает коричневый цвет" ->
-> "Запустить программу презрения";
- "В квадрате i, j преобладает желто-оранжевый цвет" ->
-> "Запустить программу радостного возбуждения" и т.д.

Тогда обученная для реализации данного алгоритма однослойная логическая нейронная сеть представлена на рис. 15.8.

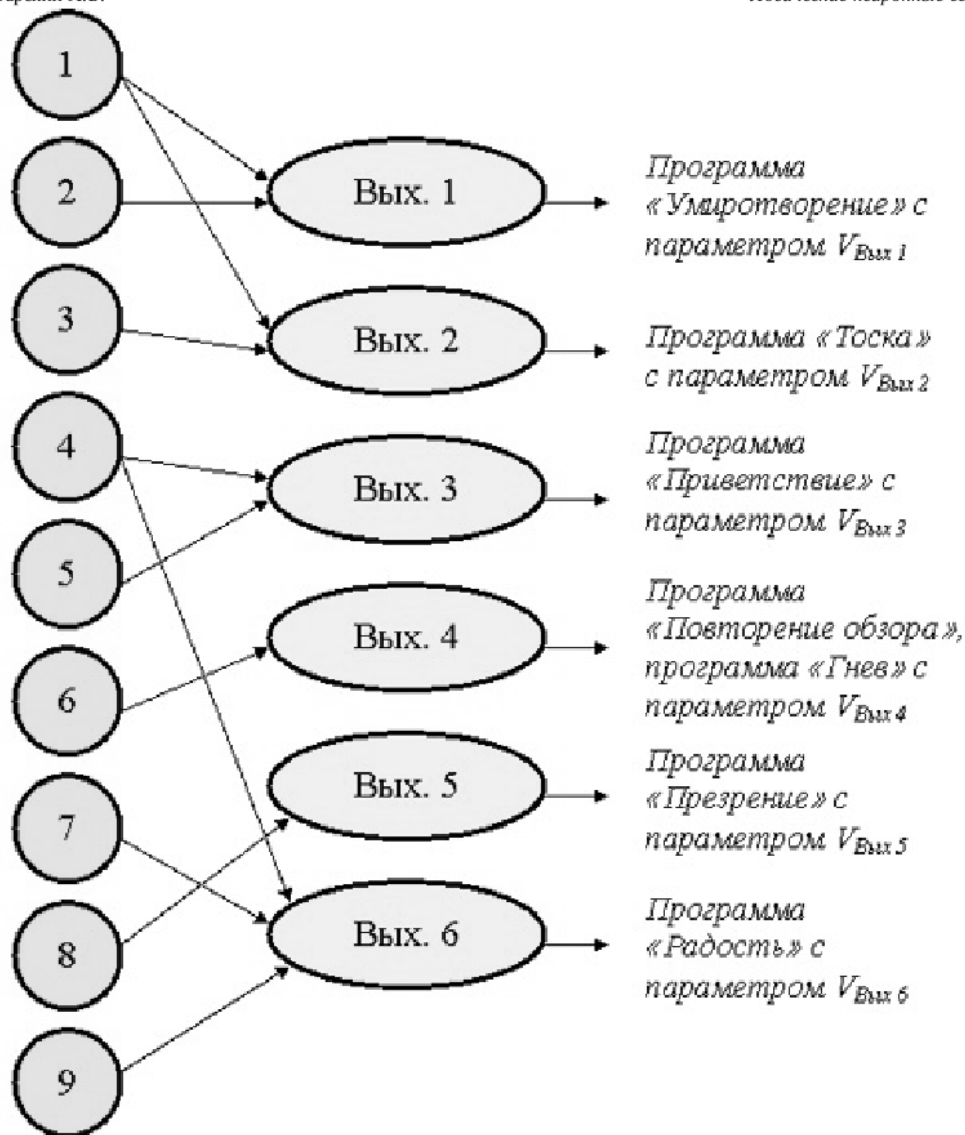


Рис. 15.8. Нейросеть для "живого" моделирования

Логическая нейронная сеть составлена по методике, обоснованной ранее. Это определило единичные значения весов синаптических связей, не указанных на рисунке.

Здесь нейрон-рецептор 1 принимает значение достоверности высказывания "В квадрате преобладает зеленый цвет" как величину возбуждения.

Рецептору 2 сообщается значение достоверности высказывания "Отсутствуют резкие переходы в оттенках".

Рецептор 3 принимает значение достоверности высказывания "Наблюдаются резкие переходы в оттенках".

Рецептор 4 возбуждается на величину достоверности высказывания "В квадрате преобладает голубой цвет".

Рецептор 5 принимает значение достоверности высказывания о схожести на полет птиц.

Рецептор 6 возбуждается на преобладание красного цвета.

Рецептор 7 возбуждается на величину достоверности обнаружения круга телесного цвета.

Рецептор 8 принимает значение достоверности преобладания коричневого цвета.

Рецептор 9 принимает значение достоверности преобладания желто-оранжевого цвета.

И так далее — по расширяемому на основе бесконечного совершенствования количеству используемых факторов.

Нейросеть строится обученной. При ее построении и в процессе ее развития с добавлением новых рецепторов и нейронов выходного слоя, ответственных за реакцию, вводятся связи на основе эталонных ситуаций. Это означает, что возбуждение связываемых при этом рецепторов равно единице. Иначе говоря, в логическом описании создаваемой системы исходные высказывания имеют значение ИСТИНА. Переход от булевых переменных к действительным, к достоверности высказываний, порождает модель *ассоциативного мышления*.

Практически за эту достоверность можно принимать частоту появления или "удельный вес" определенного цвета (соответствующих пикселей) в анализируемом квадрате. Подавая значения достоверности на *рецепторный слой*, с помощью передаточной функции находим величины возбуждения нейронов выходного слоя. Эти величины служат параметрами запускаемых программ.

Рекомендуемая передаточная функция имеет вид

$$V_{\text{Вых}} = \xi \left(\sum_v \omega_v V_v - h \right),$$

$$\left(\xi(s) = \begin{cases} s, & \text{при } s \geq 0 \\ 0, & \text{при } s < 0 \end{cases} \right).$$

Порог h целесообразно выбрать достаточно высоким, например $h = 0,5$, так, чтобы объект не выглядел слишком "нервным", возбуждаясь понапрасну из-за малой причины возбуждения.

Таким образом, "живое" моделирование — это экспериментальное направление, позволяющее проверить и осуществить все достижения на пути развития технологий искусственного интеллекта. Это реальный путь воплощения новых направлений в искусстве, в шоу-бизнесе, в учебной, игровой и развлекательной деятельности.

Вместе с тем, наряду с разработкой инструментария и постановкой задачи дальнейших исследований в области *логических нейронных сетей*, в их обучении и имитации дедуктивного и индуктивного мышления возникают сопутствующие задачи.

Прежде всего это задача оценки современных компьютерных возможностей реализации столь большого объема работ в реальном времени. Очевидно, что лишь высокопараллельная архитектура, подобная мозгу, может эффективно исполнять функции видеонейрокомпьютера.

Следует проанализировать эффективность применения современных суперкомпьютеров, таких, например, как Эльбрус-E2k.

Анализ приведенных выше математических построений говорит в пользу *SPMD*-архитектур, т.е. архитектур типа "одна программа —

много потоков данных". Актуальна постановка задачи разработки специальной параллельной приставки к персональному компьютеру (рабочей станции) — видео-нейрокомпьютера, "врезанного" в современные популярные операционные системы. Наконец, широкое привлечение сетевых технологий позволит создавать значительные распределенные сюжеты с коллективным участием многих взаимно влияющих "живых" объектов.

Реализация *трехмерной памяти*, при которой моделируемые объекты оказываются воплощенными не с помощью своего математического образа, а пространственно, также ставит новые технические задачи. Одна из них заключается в отображении содержимого памяти в некоторой экранной трехмерной среде. Это явилось бы новым решением задачи формирования стереоизображений.

15.13. Реагирующие объекты для систем интеллектуального отображения

15.13.1. Постановка задачи

Управление сложными системами с возможным участием операторов и диспетчеров предъявляет высокие требования к динамическому отображению их состояния для оперативного анализа ситуации и принятия решений. Задача оператора значительно усложняется в случае территориальной разобщенности средств системы, превращаясь в задачу многоуровневого контроля и диагностики. Возникает необходимость интеллектуальной надстройки всей системы отображения, контроля и принятия решений, производящей оперативную первичную обработку многообразной регистрирующей, отображающей и управляющей информации для предварительного, грубого отображения происходящего процесса. Задачей такого предварительного, поверхностного отображения является выяснение того, протекает ли процесс в пределах нормы, грозит ли уход за эти пределы, а также установление факта и причин тревоги.

Отображение должно быть наглядным, доступным и понятным широкому кругу наблюдателей, бесспорным и образным, а также не

должно требовать постоянных кропотливых усилий операторов. Важна и эстетическая составляющая.

Требование образности и эстетики, обеспечивающих наибольшее понимание, приводит к целесообразности использования *реагирующих объектов* — моделей живых существ или других "оживляемых" образов, по поведению которых можно судить о состоянии сложной системы.

Для такой предварительной, грубой оценки применимы принципы *ситуационного управления* [23]. Однако информацией для *реагирующего объекта* является чрезвычайно большое количество данных разнообразной природы и типов. Задача осуществления реакции становится трудно формализуемой. Ее решение возможно только с применением средств искусственного интеллекта. Множество зависимостей вида "если ..., то ..." в основе решения этой задачи приводит к выводу о целесообразности применения *логических нейронных сетей*.

Таким образом, на вход *реагирующего объекта* подается большое число разрозненных или связанных показателей состояния сложной системы, определяющих *факторное пространство*. Эти показатели, в соответствии с методикой, изложенной в лекции 12, разбиты по характеру влияния на значимые интервалы, отдельные объекты, дискретные значения или булевы переменные (вида "есть — нет"), позволяющие оценить достоверность значения каждого фактора, что может быть использовано для возбуждения *рецепторного слоя* нейросети. На основе накапливаемого опыта, экспертных оценок или теоретических исследований строится обученная, развиваемая в процессе эксплуатации, *логическая нейронная сеть* — основа поведения *реагирующего объекта*.

На данном этапе утверждения важности данного направления не следует брать на себя смелость заявления о *самообучении*, тем более — об автоматическом обобщении опыта, т.е. об имитации индуктивного мышления. Несомненно, это задачи будущего развития технологии применения *реагирующих объектов*.

Модели *реагирующих объектов* могут быть трех типов:

- натурная модель, выполненная, в частности, в соответствии с известными технологиями *робототехники* или театра кукол;
- компьютерная (электронная) модель, использующая графические технологии — двумерные и трехмерные;
- компьютерная (электронная) модель на основе клип-технологий.

Все три типа моделей предполагают связь нейронов выходного слоя с соответствующими программами (процедурами) имитации реакции объектов на ситуацию по принципу ассоциативного мышления. При этом величина возбуждения нейрона выходного слоя служит основным параметром соответствующей программы, определяющим реакцию. Этим обеспечивается возможность совмещения различных реакций, дающих некоторую результирующую.

Натурная модель, ввиду значительного применения элементов механики, электротехники и других технологий, по-видимому, имеет ограниченное применение в сфере развлечений или при медленно изменяющихся факторах (реакция на прогноз погоды) и др.

Электронная графическая модель при кажущейся доступности технологий требует критического рассмотрения и развития достаточно разнообразных современных средств для реализации динамики их поведения, как реакции на входные сигналы. В этом случае решающую роль играет проблема управляемой анимации в реальном времени — проблема "живого" моделирования.

Клип-модель, предполагающая оперативное вторжение в развитие сюжета, базируется на хранении большого количества вариантов — клипов, что может вызвать проблемы памяти, а также проблемы временных и технических объемов разработки.

Тем не менее все возможности должны быть экспериментально исследованы на моделях.

Помимо актуальной задачи *интеллектуального отображения* в сложных управляющих системах, применение *реагирующих объектов* может быть значительно более широким. Это — системы развлекательные, игровые, демонстрационные, рекламные, медицинской и технической диагностики, контроля

несанкционированного доступа и другие из сфер искусства, науки, техники, бизнеса.

15.13.2. Основы построения модели системы интеллектуального отображения для "плоской" анимации

Целью моделирования является:

1. Обоснование возможности создания на основе нейросетевой технологии *реагирующих объектов* — объектов, созданных в памяти компьютера, обрабатываемых средствами анимации по сценарию, ход которого зависит от внешней, динамически поступающей информации.
2. Демонстрация возможностей практического применения нетрудоемкого аппарата *логических нейронных сетей*.
3. Создание, исследование и развитие средств *плоской анимации*, динамически учитывающей внешние факторы.
4. Конкретизация области применения *реагирующих объектов*.
5. Конкретизация задач и возможностей применения существующих средств компьютерной графики для *интеллектуального отображения*.
6. Обоснование необходимости разработки средств построения *реагирующих объектов* в модели *трехмерной памяти* для повышения эффективности компьютерного и натурального их моделирования.

Основная часть экрана для работы пользователя с моделью *интеллектуального отображения* представлена на рис. 15.9.



Рис. 15.9. Экран модели

Основными элементами этого экрана являются два поля: поле показываемых (создаваемых) с помощью мыши образов (ПО) и поле реагирующего объекта (РО).

Если ПО пусто или контурно нарисованный в нем объект не узнан РО, последний находится в состоянии покоя, т.е. показа некоторого нейтрального или начального состояния, например первого показанного на рис. 2 "нулевого" кадра.

Если в реальном времени, независимо от текущего состояния РО, на ПО сформирован рисунок, похожий на один из эталонов, которые знает РО, то в соответствии с запомнившейся реакцией на этот образ РО начнет переходить в состояние, обусловленное ассоциациями. Например, результатом его реакции на показанное яблоко может быть кислая гримаса огорчения, как изображено на рисунке.

Однако этот переход может быть прерван показом нового образа на поле ПО. Тогда, если предъявленный объект узнан, переход в новое состояние осуществляется из достигнутого текущего состояния. Если

объект не узнан, но прошлое содержимое ПО потеряно, РО стремится вернуться в начальное состояние.

В ограниченной демонстрационной реализации *логическая нейронная сеть*, обученная на этапе ее построения, или совершенствуемая в статическом режиме, реализует алгоритм, аналогичный распознаванию букв по конфигурации засветки клеток экрана в соответствии с множеством известных эталонов, как рассмотрено в лекции 2.

Увеличение числа клеток экрана ПО (повышение разрешающей способности) служит увеличению возможного числа эталонов и вероятности распознавания.

Для имитации настроения РО разрабатывается база данных — набор *ключевых кадров*, например, показанных на рис. 15.10.

На основе этих кадров формируются *сюжетные ряды* — последовательности *ключевых кадров*, ведущие, при их реализации, к отображению определенных реакций объекта: к состоянию радости, огорчения и т.д.

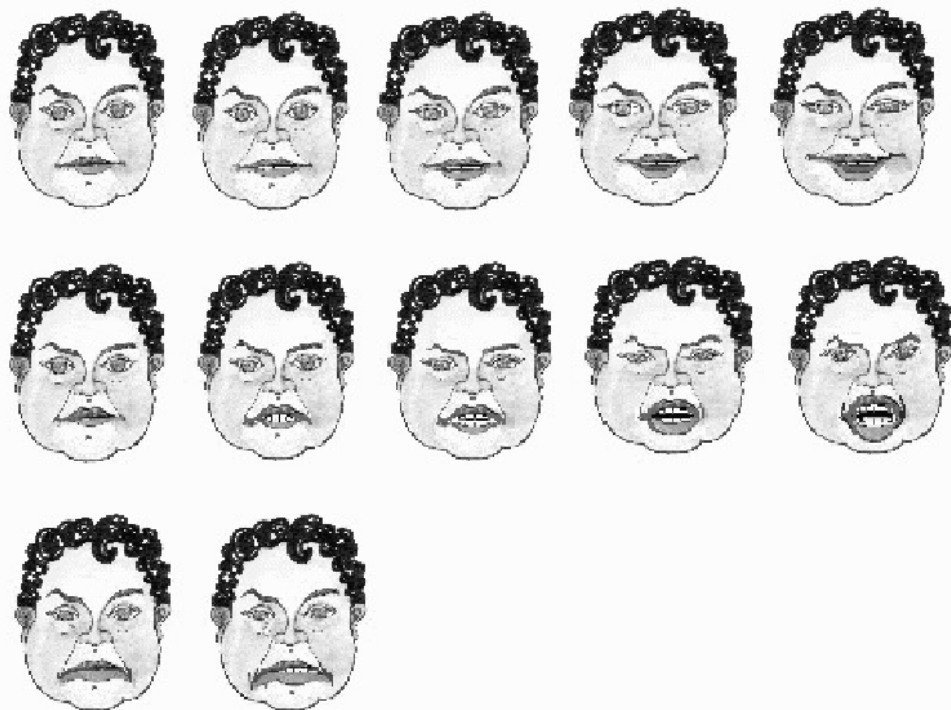


Рис. 15.10. Образы для комплектации сюжетных рядов

Динамическая подстановка *сюжетных рядов* в реальном времени, в соответствии с показанным рисунком и вызываемыми ассоциациями, является основной функцией модели, вызывающей изменяемую во времени реакцию объекта.

Система анимации на основе текущего *ключевого* или промежуточного кадра и в соответствии с текущим известным целевым *ключевым кадром* из *сюжетного ряда* осуществляет визуализацию с помощью расчета и показа промежуточных кадров с заданной частотой, обусловленной зрительным восприятием и производительностью компьютера.

Функциональная схема модели показана на [рис. 15.11](#).

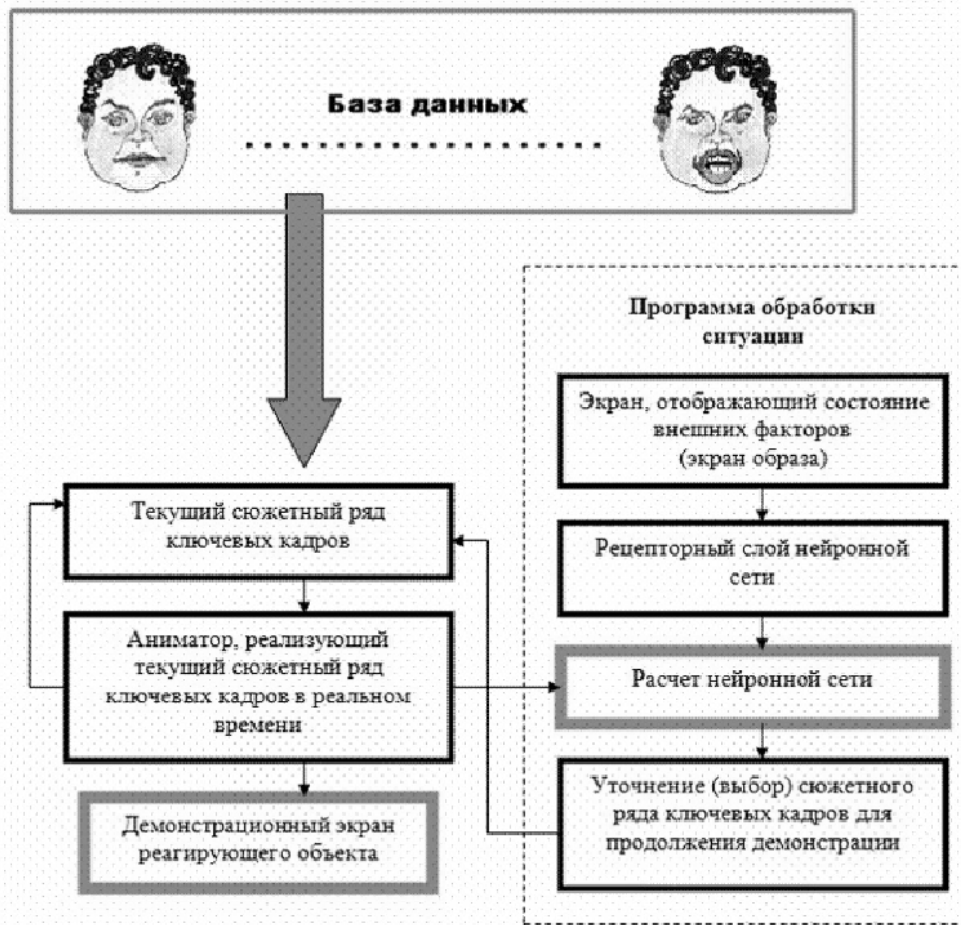


Рис. 15.11. Функциональная схема модели

База данных содержит пополняемое в процессе эксплуатации множество *ключевых кадров*. В рассматриваемом примере это множество различных выражений лица, на основе которых могут быть созданы переходы к различным реакциям на предъявляемые образы.

Поведение *реагирующего объекта* отображается на демонстрационном экране.

Каждая реакция (веселье, огорчение и т.д.) приводит к медленному, в реальном времени, переходу к соответствующему выражению лица. Для этого выделяется *текущий сюжетный ряд ключевых кадров*, определяющий постепенный (однако через значительные промежутки

времени, например, через одну секунду) переход к конечному выражению — к последнему *ключевому кадру* этого ряда.

Этот переход осуществляет аниматор. Он отображает на демонстрационном экране кадры *сюжетного ряда* и промежуточные, рассчитанные им кадры. Промежуточные кадры рассчитываются в соответствии со временем перехода от одного *ключевого кадра* к другому и с необходимой частотой показа кадров, достаточной для комфортного зрительного восприятия.

Для облегчения понимания перейдем к рассмотрению "видимых" объектов — образов: букв, фруктов, зверей и т.д. В общем случае очевидно, что каждую точку в *факторном пространстве*, характеризующем сложную систему, можно интерпретировать как некий образ, изображение которого в определенной конфигурации возбуждает рецепторы нейронной сети. Так что проблема распознавания образов в рамках проблемы компьютерного зрения [4], актуальная, например, в системах игр и развлечений, вовсе не подменяет проблему *интеллектуального отображения в факторном пространстве*, служа лишь наглядности.

Тогда, на экране образа производится асинхронное, по отношению к демонстрации, задание образа — контурного рисунка (с помощью мыши или джойстика) некоторого объекта, как из числа тех, которые могут быть распознаны, так и произвольного.

С каждым объектом, которые система "знает", связан свой *сюжетный ряд ключевых кадров*. Например, показ лимона соответствует выбору *сюжетного ряда*, приводящего к реакции "кисло", показ морковки — к реакции "сладко" и т.д. Необходимо обеспечить именно динамическое, асинхронное, независимое задание образов. При этом переход к новому *сюжетному ряду* ключевых слов осуществляется от только что показанного кадра — промежуточного или *ключевого*.

Выбор нового *сюжетного ряда ключевых кадров* производится блоком обработки ситуации, с помощью нейронной сети.

Ее *рецепторный слой* определяется клетками экрана образа, т.е. выбранной разрешающей способностью. Размер выходного слоя определяется количеством различных реакций. (Допустимое развитие

на этапе высокой актуальности демонстраций пока не рассматривается.)

На основе максимального возбуждения одного нейрона выходного слоя нейросети производится определение реакции: выбор соответствующего этому нейрону *сюжетного ряда ключевых кадров*. Новый *сюжетный ряд* становится текущим.

Для начального этапа разработки — эксперимента и демонстрации — достаточно выбрать однослойную *логическую нейронную сеть*. *Рецепторный слой* соответствует клеткам экрана образа. Величина их засветки (полная, частичная, нулевая) определяет величину возбуждения соответствующего рецептора.

Запоминание образа-эталона, т.е. обучение, производится непосредственным установлением связей между клетками области экрана, включающей эталон (с учетом допусков, позволяющих неточное воспроизведение эталона в рабочем режиме, а лишь приблизительное — нетвердой рукой, с зашумлением и т.д.), и нейроном выходного слоя, "отвечающим" за этот эталон. Веса связей — единичные. Данный принцип изложен в [лекции 2](#).

На [рис. 15.12](#) показаны связи, отображающие запоминание эталона "яблоко". Для упрощения выбран экран в 6x4 клеток. Покрываемые эталоном клетки закрашены.

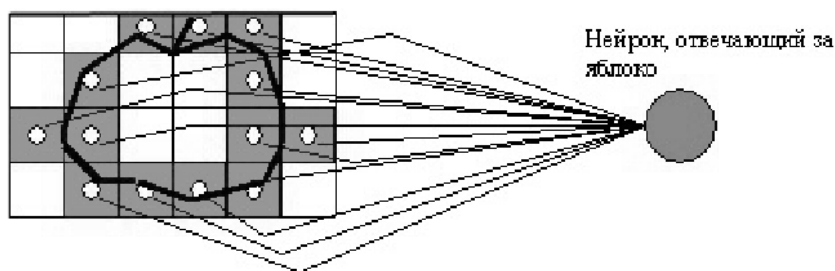


Рис. 15.12. Обучение яблоку

После обучения яблоку, т.е. после фиксации введенных связей, введем эталон "морковка". Отдельно его расположение на экране может быть проиллюстрировано на [рис. 15.13](#).

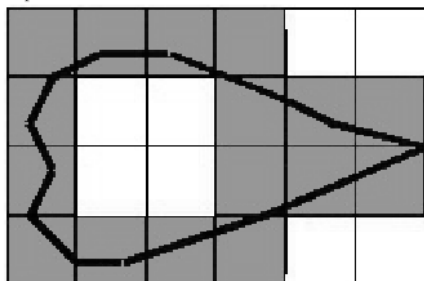


Рис. 15.13. Расположение морковки на экране

Закрепим за морковкой нейрон выходного слоя и свяжем его со всеми закрасенными клетками экрана.

Совместное запоминание двух эталонов показано на [рис. 15.14](#).

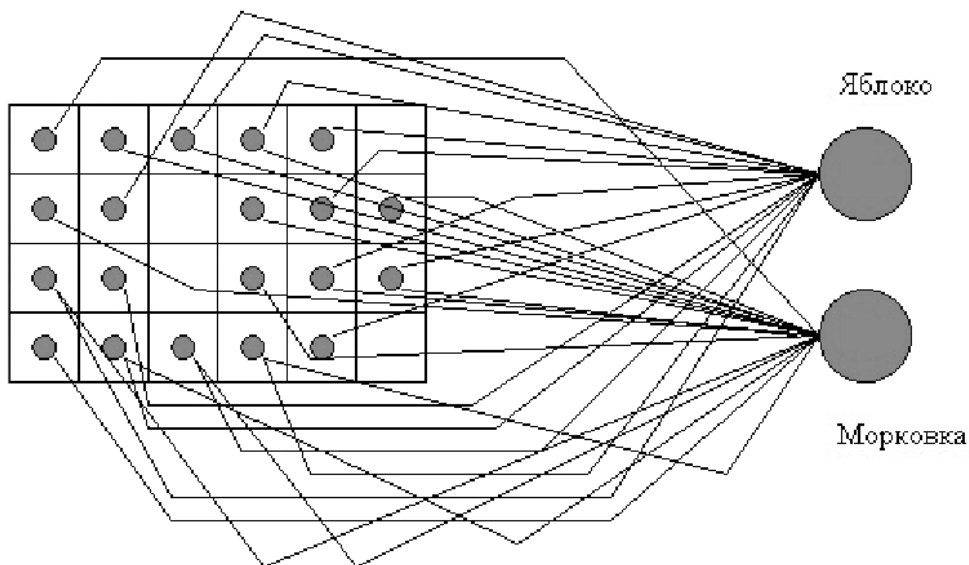


Рис. 15.14. Совместное запоминание эталонов яблока и морковки

Нейросеть представлена матрицей следования на [рис. 15.15](#). Здесь рецепторы — клетки экрана наделены сквозной нумерацией вдоль строк и по строкам.

Здесь j принимает значение номеров рецепторов, участвующих с единичными весами (поэтому веса не указаны в *передаточной функции*) в запоминании эталона, G_j — число таких рецепторов. Делением на G_j производится приведение всех величин возбуждения по эталонам в один диапазон изменения. Это необходимо для того, чтобы "большие" образы, частично засвеченные вследствие возмущения "маленьких" образов, не подавляли эти последние. Так, в данном примере "морковка" засвечивает 16 клеток, а "яблоко" — только 13.

Порог h определяет предел допустимости вывода.

Конечно, в примере экран содержит всего 24 клетки, поэтому его разрешающая способность, а следовательно, вероятность распознавания очень низки. Тем не менее с достаточной уверенностью можно считать, что если на экране произвольно нарисовать яблоко, то, скорее всего, рисунок более засветит те рецепторы, которые связаны с нейроном "яблоко", и в меньшей степени — рецепторы "морковки". Тогда при расчете возбуждения нейронов выходного слоя следует ожидать, что нейрон "яблоко" будет обладать большей величиной возбуждения.

Однако в любом случае величина возбуждения должна преодолеть порог на тот случай, если это и не морковка, и не яблоко, а также при большом "зашумлении".

Алгоритм обработки выделенных строк *матрицы следования однослойной сети* примитивен.

Последовательно просматриваются строки, соответствующие нейронам выходного слоя, и для каждой строки рассчитывается значение *передаточной функции*. Для этого используются значения возбуждения тех рецепторов, в столбцах которых находятся единицы. Затем выделяется максимально возбужденный нейрон.

Работа модели, как системы реального времени, тактируется в соответствии с темпом выдачи кадров, т.е. с частотой, обеспечивающей комфортное восприятие.

Проблема анимации заключается в выдаче промежуточных кадров по *ключевым кадрам* с частотой восприятия. На каждом шаге анимации

анализируются два кадра: только что показанный и *ключевой*, к изображению которого аниматор последовательно стремится.

С целью демонстрации потенциальных возможностей воспользуемся упрощенным алгоритмом *плоской анимации* по контрольным точкам и разрезам.

А именно, при формировании промежуточных кадров будем руководствоваться смещением контрольных точек и разрезов в системе координат, связанной с объектом. Список координат контрольных точек, а также координат точек начала и конца каждого разреза, сопровождает каждый *ключевой кадр*. Между кадрами сохраняется соответствие их контрольных точек и разрезов.

На рис. 15.16 показаны два смежных *ключевых кадра* : кадр K и кадр L . Однако кадр K в общем случае — некоторый начальный (*ключевой* или промежуточный) кадр i (значению $i = 0$ соответствует кадр K), от показа которого необходимо плавно, через показ ряда промежуточных кадров, перейти к показу кадра L . Тогда следующий, $i+1$ -й кадр, выдаваемый в следующем такте работы модели, должен быть получен на основе одного шага изменения кадра i .

Пусть на рисунке K_1-K_5 — исходные контрольные точки (конечно, невидимые), заданные своими координатами (x_{Kj}, y_{Kj}) ; L_1-L_5 - соответствующие им контрольные точки (x_{Lj}, y_{Lj}) в результирующем кадре L , $j = 1, \dots, 5$.

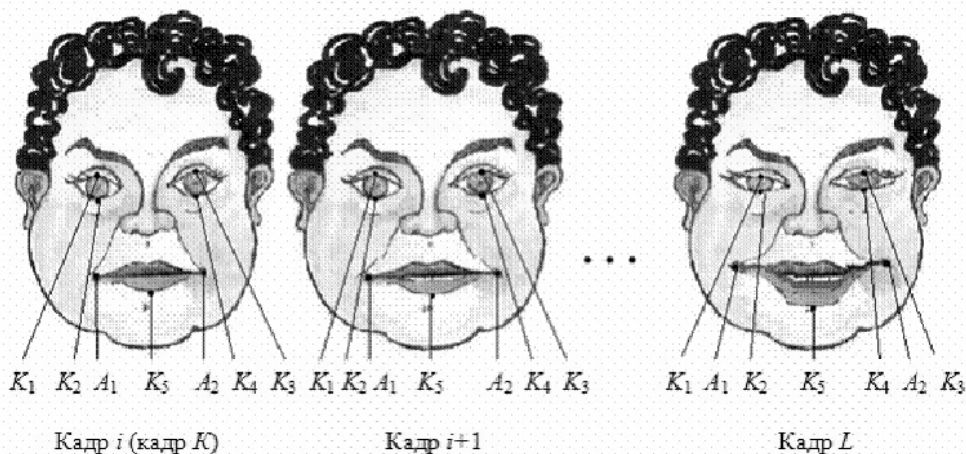


Рис. 15.16. Кадры анимации

Контрольные точки K_j через N тактов должны перейти в контрольные точки L_j , увлекая за собой некоторую окрестность Δ_j . Причем моделируется свойство "вязкости", при котором более удаленные от центра (K_j) точки смещаются меньше. Более того, если область Δ_j пересекается разрезом, то перенос точек по другую сторону разреза не производится (см. открытие рта). В примере задан такой отрезок $[A_1, A_2]$, который должен перейти в отрезок $[B_1, B_2]$.

Алгоритм

1. Производится начальная загрузка в буфер показа БП "нулевого" кадра, т.е. фонового, нейтрального выражения объекта в состоянии ожидания.
2. Реализуется временной режим работы модели, поддерживающий показ кадров — *ключевых* или *промежуточных* — с частотой 20 Гц. Он вхолостую вырабатывает остаток времени, оставшийся от работы программ, до ближайшего значения, кратного 50 мс.
3. Демонстрируется очередной кадр из БП и производится обращение к Программе обработки ситуации, рассмотренной ранее.
4. Выполняется после возврата из Программы обработки ситуации.

Анализируется: был ли показанный кадр *ключевым* или промежуточным? Если показанный кадр был промежуточным, выполняется 7.

5. Если показанный кадр был *ключевым*, анализируется: *иссяк ли сюжетный ряд*, т.е. выработала ли Программа оценки ситуации продолжение *сюжетного ряда*, как реакцию на внешние факторы (рисунок)? Если *сюжетный ряд* *иссяк*, управление передается 1 для загрузки "нулевого" кадра, который будет показываться, пока не выработается новый *сюжетный ряд*. Если *сюжетный ряд* не *иссяк*, выполняется 6.
6. Производится подготовка формирования промежуточных кадров между только что показанным K и следующим L . Вводится счетчик показа промежуточных кадров СПК. Для упрощения демонстрационной модели считаем, что между любыми двумя смежными *ключевыми кадрами* следует показать 19 промежуточных. То есть выполняется операция СПК := 20.
7. СПК уменьшается на единицу для показа очередного промежуточного кадра.
8. Если в результате 7 СПК достиг нуля, показ промежуточных кадров считается законченным, следующим показываемым кадром должен быть кадр L . Выполняется 9. Если СПК не равно нулю, выполняется 10.
9. Кадр L загружается в БП, далее выполняется 2.
10. Если СПК не равно нулю, организуется цикл по j (шаги 11-16) перебора контрольных точек объекта. По окончании цикла организуется переход к 20.
11. Для контрольной точки K_j (x_{Kj} , y_{Kj}) отыскивается соответствующая ей точка L_j (x_{Lj} , y_{Lj}) из СКТ кадра L . Отыскивается область Δ_j радиуса R_j с центром в точке K_j , включающая точку L_j (ее проекцию на кадр K). Радиус R_j вычисляется из соотношения

$$2R_j = r(K_j, L_j) + l,$$

где r — расстояние, l — поправка, подбираемая экспериментально.

12. Организуется цикл перебора точек $P(x, y)$ из окрестности Δ_j контрольной точки K_j . Этот перебор осуществляется "циклом в цикле":

```

for  $x = x_{K_j} - R_j$  to  $x_{K_j} + R_j$ 
for  $y = y_{K_j} - \sqrt{R_j^2 - (x - x_{K_j})^2}$ 
to  $y = y_{K_j} + \sqrt{R_j^2 - (x - x_{K_j})^2}$  do ...

```

По окончании перебора выполняется 17.

13. Проверяется, разделены ли точки P и L разрезами. Данная проверка производится в результате решения следующей задачи.

Пусть на рис. 15.17 точки $P(x, y)$ и $L(x_L, y_L)$ (индекс j опустим) находятся по разные стороны отрезка $[A, B]$, где $A = (x_A, y_A)$, $B = (x_B, y_B)$, так, что отрезки $[P, L]$ и $[A, B]$ пересекаются. (Именно этот случай запрещает перемещение точки P за разрез — в окрестность точки L .)

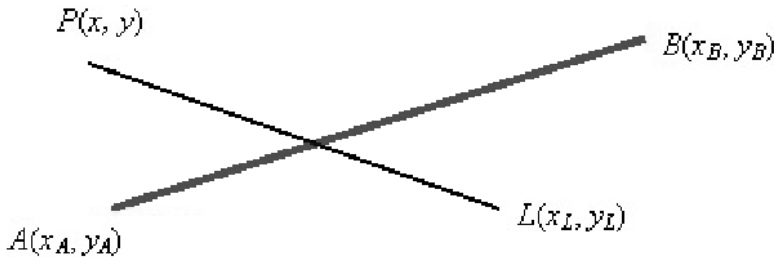


Рис. 15.17. Анализ разреза

Рассмотрим параметрическое описание этих отрезков и запишем уравнения для нахождения точки пересечения (ξ, η) .

$$[P, L] : \begin{cases} \xi = x + k_1(x_L - x) \\ \eta = y + k_1(y_L - y) \end{cases}, 0 \leq k_1 \leq 1,$$

$$[A, B] : \begin{cases} \xi = x_A + k_2(x_B - x_A) \\ \eta = y_A + k_2(y_B - y_A) \end{cases}, 0 \leq k_2 \leq 1.$$

Или:

$$\begin{aligned} x + k_1(x_L - x) &= x_A + k_2(x_B - x_A), \\ y + k_1(y_L - y) &= y_A + k_2(y_B - y_A). \end{aligned}$$

Отсюда

$$\begin{aligned} k_1 &= \frac{(x - x_A)(y_B - y_A) - (y - y_A)(x_B - x_A)}{(x_B - x_A)(y_L - y) - (y_B - y_A)(x_L - x)} \\ k_2 &= \frac{(x - x_A)}{(x_B - x_A)} + k_1 \frac{(x_L - x)}{(x_B - x_A)}. \end{aligned}$$

Если отрезки пересекаются, выполняется условие $0 \leq k_1 \leq 1$, $0 \leq k_2 \leq 1$. В противном случае отрезки не пересекаются, и перемещение точки P в направлении к точке L_j допустимо. Если точки P и L разделены разрезом, производится переход к 12. В противном случае выполняется 14.

14. Рассчитываются координаты переноса точки P при формировании очередного промежуточного кадра K*:

$$\begin{aligned} x_{\text{пер}} &= x + (x_{Lj} - x_{Kj}) \left(1 - \frac{r(P, K_j)}{R_j} \right) \frac{1}{|\text{СПК}|}, \\ y_{\text{пер}} &= y + (y_{Lj} - y_{Kj}) \left(1 - \frac{r(P, K_j)}{R_j} \right) \frac{1}{|\text{СПК}|}. \end{aligned}$$

Здесь учитывается удаление точки P от центра — от точки K_j. Так, на окружности радиуса R_j перенос затихает вовсе. Текущее значение СПК характеризует оставшуюся часть смещения.

15. Проверяется, является ли перенесенная точка контрольной или границей разреза. Если не является, выполняется 12.
16. Если перенесенная точка является контрольной или границей

разреза, фиксируются новые значения координат перенесенной точки (ее образом становится точка K_j^* , если это была контрольная точка) в СКТ промежуточного кадра K^* . Далее выполняется 12.

17. Выполняется по окончании цикла перебора точек из Δ_j окрестности (шаг 12). Формируется цикл обработки разрезов, оказавшихся в этой окрестности. По окончании цикла выполняется 10. Организуется перебор точек $P(x, y)$ этой окрестности, подобно организованному в блоке 12.
18. Проверяется, является ли очередная анализируемая точка "пустой" вследствие деформации (растяжения) объекта. При отрицательном результате анализа выполняется 20.
19. Если точка "пустая", находится ее усредненное значение на основе непустых окружающих ее точек:

$$(P(x, y)) := \frac{1}{S} [(P(x_{n-1}, y_{m-1})) + (P(x_{n-1}, y_m)) + (P(x_{n-1}, y_{m+1})) + (P(x_n, y_{m-1})) + (P(x_n, y_m)) + (P(x_n, y_{m+1})) + (P(x_{n+1}, y_{m-1})) + (P(x_{n+1}, y_m)) + (P(x_{n+1}, y_{m+1}))]$$

где $P(x, y)$ отождествляется с точкой $P(x_n, y_m)$ для отображения окружающих точек, S — число таких "непустых" точек.

20. Организуется цикл обработки разрезов кадра K^* . После обработки всех разрезов выполняется 24.
21. Для каждого разреза формируется некоторая окрестность, опирающаяся на разрез, и организуется цикл перебора точек $P(x, y)$, образованного прямоугольника.
22. Проверяется: точка $P(x, y)$ — "пустая"?
23. Если да, отыскивается точка $Q(x, y)$ в кадре L (с теми же координатами) и налагается на точку P : $P(x, y) := Q(x, y)$.

Далее выполняется 20.

24. После анализа всех разрезов (шаг 20) считается, что кадр K^* полностью сформирован. Производится загрузка сформированного кадра в буфер показа БП и управление передается 2 для продолжения работы модели.

15.13.3. Клип-моделирование интеллектуального отображения

Программная реализация *клип-модели реагирующего объекта* создана в среде системы визуального объектно-ориентированного программирования Delphi [6].

Внешний вид экрана для работы пользователя с моделью представлен на рис. 15.18.

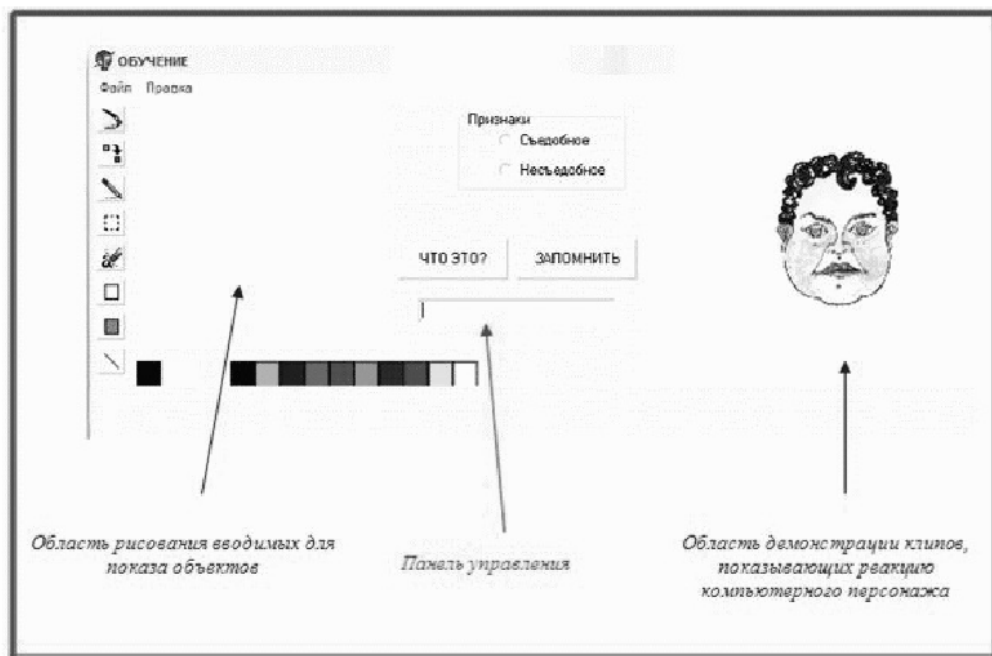


Рис. 15.18. Вид экрана для работы с клип-моделью

Область рисования вводимых для показа компьютерному персонажу объектов реализована в виде простого графического редактора на основе примера, приведенного в [6]. Данный графический редактор

позволяет создавать контурные рисунки в области рисования с помощью набора предлагаемых инструментов (карандаш, ластик) и заливать замкнутые контуры выбранным из палитры цветом с помощью инструмента "кисть".

Блок анализа внешней обстановки модели, имитирующий работу "органов зрения" компьютерного персонажа, запускается нажатием кнопки "ЧТО ЭТО?". Прежде всего осуществляется сканирование области рисования. При обнаружении в этой области какого-либо изображения производится нахождение границ изображения. Для этого вокруг него строится прямоугольная рамка, которая затем разбивается на число клеток, соответствующее размеру *рецепторного слоя* нейронной сети с учетом пропорций изображения. Таким способом в модели реализовано "фокусирование взгляда" компьютерного персонажа на объекте, и поэтому размер изображения в области рисования не имеет значения, что очень важно при распознавании.

Размер *рецепторного слоя* нейронной сети, т.е. разрешающая способность "зрения" компьютерного персонажа, является параметром данной модели. В данном примере он выбран 20x20, т.е. 400 рецепторов.

Область демонстрации клипов, показывающих реакцию компьютерного персонажа, реализована с помощью компонента *Animate* из библиотеки компонент *Delphi5*. Данный компонент обладает свойством *FileName*, которое позволяет программным способом указывать имя файла, содержащего требуемый для проигрывания клип. Кроме того, у компонента *Animate* имеется метод *Play* с параметрами *FromFrame* (начальный кадр), *ToFrame* (конечный кадр) и *Count* (число повторений анимации). Этот метод позволяет программным способом, динамически указывать, с какого кадра начинать и каким заканчивать проигрывание анимации. Таким образом, остается с помощью любой программы, позволяющей разрабатывать клипы, создать базу клипов, демонстрирующих реакции компьютерного персонажа на внешние раздражители.

В модели введены ограничения на качественные характеристики изображаемых объектов. Объекты могут быть только двух видов: съедобными и несъедобными. Соответственно компьютерный

персонаж имеет только три реакции на внешнюю среду: 1 — состояние ожидания, оно же — если предмет незнакомый, 2 — состояние удовольствия, если узнан съедобный предмет, 3 — состояние неудовольствия, если узнан несъедобный предмет.

Для демонстрации достаточно, чтобы база данных содержала всего один клип, созданный с помощью программы Flash4 [17]. Данный клип содержит четыре фрагмента, разделенных *ключевыми кадрами*, и демонстрирует переходы компьютерного персонажа из состояния ожидания в состояние удовольствия, затем снова в состояние ожидания, затем переход в состояние неудовольствия, затем снова в состояние ожидания.

Первоначально компьютерный персонаж "ничего не знает", т.е. база данных, где хранятся матрицы следования, значения *передаточных функций* и качественные признаки эталонов изображаемых объектов, пуста. Данная программная реализация модели позволяет сразу начать "обучение" компьютерного персонажа.

На рис. 15.19 показан результат предъявления необученному компьютерному персонажу изображения цветка.

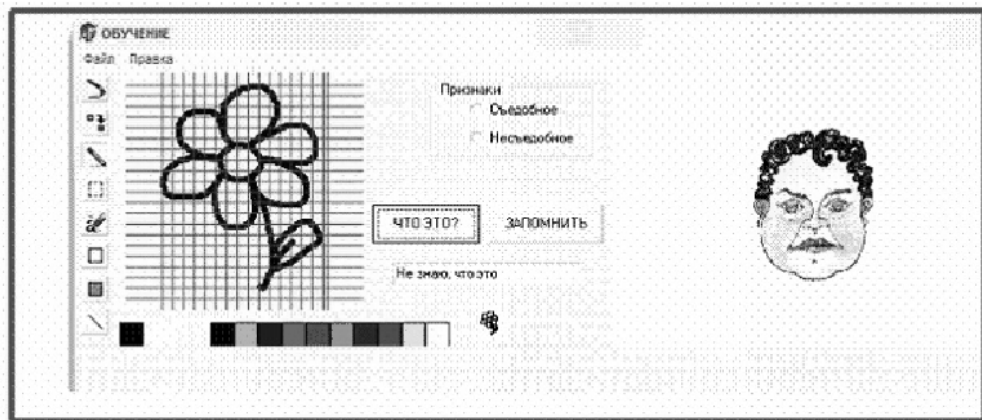


Рис. 15.19. Реакция на предъявлении цветка до обучения

Видно, что реакция отсутствует, а в поле названия написано "Не знаю, что это". Теперь, если в поле названия написать "Цветок", указать признак "несъедобное" и нажать кнопку "ЗАПОМНИТЬ", то начинает работать блок обучения модели. Прежде всего, вдоль контура рисунка

производится закрашивание тех клеточек, которые захватываются контуром, затем каждая закрашенная клетка обрамляется закрашенными клетками справа, слева, сверху и снизу. Таким образом, вдоль контура формируется полоса допуска для учета "дрожания руки" при рисовании. Далее вычисляется значение *передаточной функции* и формируется строка *матрицы следования*. Затем вся информация об объекте запоминается в базе данных. Так происходит обучение компьютерного персонажа.

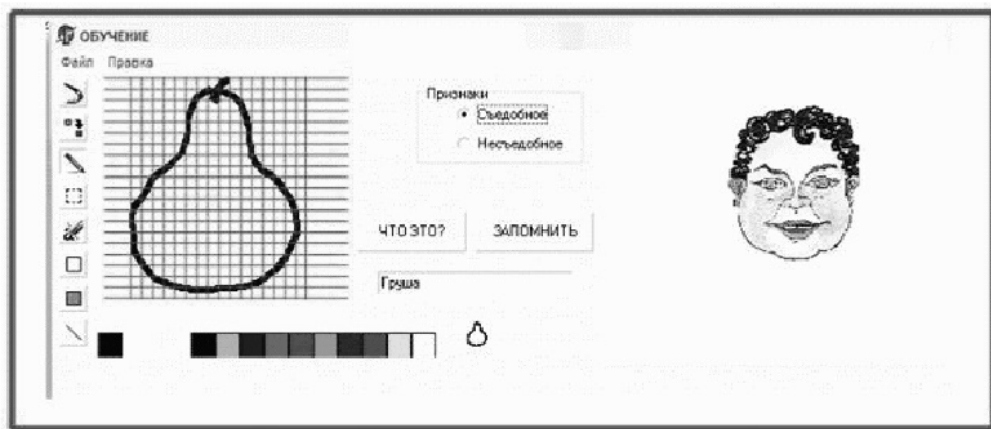


Рис. 15.20. Реакция на знакомое съедобное

Когда компьютерный персонаж уже обучен, т.е. база данных не пуста, то при нажатии на кнопку "ЧТО ЭТО?" после сканирования области рисования и "фокусирования зрения" запускается блок распознавания. В нем сначала осуществляется закрашивание клеточек вдоль контура рисунка, а затем производится перебор эталонов базы данных и попытка "уложить предъявленный" образ в границы эталонов, хранимых в базе данных; при этом формируется список эталонов, в границы которых вписывается предъявленный образ. Если список пуст, то образ не узнан. В противном случае в качестве узнанного выбирается эталон в соответствии с максимальным возбуждением нейрона выходного слоя. В поле названия записывается имя эталона, указывается значение качественного признака (съедобное, несъедобное) и запускается клип реакции компьютерного персонажа. Пример такого результата показан на [рис. 15.20](#).

Изложенный в лекции материал имеет скорее принципиальный характер, затрагивая широкий спектр вопросов экспериментального моделирования. Несомненно, что задача "живого" моделирования, как и проблема создания на основе *реагирующих объектов* интеллектуальной надстройки для оперативного отображения состояния сложных управляющих систем, а также систем предупреждения, является актуальной в большей степени, чем задача рынка бизнеса и развлечений (хотя применение в сфере бизнеса сулит немалый успех).

Вместе с тем все более проявляется возможность не только научно-технического, например, в робототехнике, но и художественного воплощения и использования таких моделей, могущих определить новое направление в области искусства: изобразительного, зрелищного, сценического (театр марионеток), культурно-развлекательного и просветительного (исторические реконструкции), в технике кино, в компьютерных играх, в рекламе, в учебных наглядных пособиях, в новых способах *интеллектуального отображения* реальных процессов (например, прогноза погоды) и т.д.

Тогда возникает и проблема отображения коллективного разума. Это может быть актуально как для систем вида "человек — модель ИИ", так и для систем взаимодействующих моделей ИИ. При компьютерной реализации *реагирующих объектов* можно проделать следующий эксперимент: образовать круг мониторов (снабженных средствами видеовхода каждый) рабочих станций локальной вычислительной сети, чтобы реализованные на этих станциях объекты "видели" других членов "коллектива" [26]. Можно понаблюдать, как они будут реагировать друг на друга — придет ли система в умиротворенное состояние или, все более возбуждаясь, потребует постороннего вмешательства.

Перспективные нейросетевые технологии

Читателю предлагается пофантазировать и наметить пути собственного применения логических нейронных сетей в задачах обеспечения безопасности, защиты информации, медицинской диагностики, в сфере образовательных услуг, в задачах распознавания рукописных текстов, составления расписаний, экстренного торможения, прогнозирования, а также в искусстве и индустрии развлечений.

" – А теперь действовать, действовать и действовать! – сказал Остап, понизив голос до степени полной нелегальности".

И. Ильф, Е. Петров. "Двенадцать стульев"

Мы не из праздного любопытства интересуемся принципами искусственного интеллекта – и нейросети в частности. Нам необходимо уловить тот минимум знаний, который развязывает нам руки для эксперимента, демонстрации, внедрения и бизнеса. Мы считаем обязанностью спросить себя: "Ну, и что полезного, великого и славного, веселого и наваристого из всего этого можно получить? В каком направлении трудиться, какие задачи решать?"

... Прошли годы. Герои-реформаторы, теоретики-идеалисты, не боясь проклятия потомков, с этим непотопляемым "если все – сознательно и дружно", открыли клапана для самого низкого, звериного, всколыхнули всю муть с хлипкого дна рабски незрелой человеческой души ...

... Русский ландшафт украсился величественными воровскими коттеджами на разбазаренных полях, забывших ласковое ощупывание комбайнов ... Угрожающими шествиями с хоругвями потянулись "патриоты" ... Братва ринулась покорять мир ... Торговые точки благословенного побережья Испании запестрели объявлениями: "Мы говорим по-русски!" ... Насилие заполонило телеэкран, облагораживая физическое и нравственное калечение молодой смены в войнах, конфликтах, политических и воровских разборках ...

Легализация теневого бизнеса обрела главенствующую силу. Джентльменизация (стимулируемая обнадеживающим взаимным отстрелом!) люмпен-пролетарского, криминально-коммунистического

контингента бизнес-воинства, этих новых русских космополитического происхождения, бурными темпами социалистического соревнования свершилась за одно-единственное поколение . . .

. . . Раздираемая Страна все более превращалась в Территорию с населением . . .

И вот, когда мы обречено и покорно, с голоду забыв о Демократических Преобразованиях, мечтая о в срок выданном профессорском жаловании, грустно сидели на своих рабочих местах, не тронутых евроремонтом, раздался требовательный телефонный звонок. Густым басом, которым исполняют "Интернационал", свободно и независимо, уверенно и повелительно, братски проникновенно и покровительственно шутиливо дядя Рамзай пригласил нас к себе, на загородную виллу о тридцати пяти комнатах.

Инстинктивно прижимаясь друг к другу, стесняясь заношенных штанов, затравленно зыря на открывающееся великолепии, утверждающее победный бросок "из грязи – в князи", мы неловко повылезали из шикарных лимузинов . . .

. . . Мы робко грудились невдалеке от обильно сервированного стола, а дядя Рамзай вводил нас в стыд и уныние от чувства неполноценности сообщением о головокружительном взлете наших деятельных недавних знакомых . . . А мы-то им прочили!.. Нет, нам действительно нет места в этом мире . . .

. . . Вот меж золотых колон двери зала вдвинулся черный квадрат Малевича, дополненный модно лысым полушарием, – и мы узнали Васю . . . Мы соболезненно и чинно выпили за Петю, безвременно ушедшего в слоисто бисквитном шикарном саркофаге с кисточками и иллюминатором . . . Сновали затасканно-рельефные, еще узнаваемые в буйных красках макияжа, женские фигуры славных боевых подруг – Марины, Регины и др. – с кокаинно блестящими глазами . . . Царственно вплыла св. Аполлинария, таща за собой шлейф подозрительно смуглых отпрысков . . . Преданно суетилась охрана (служба безопасности!), не отрывая мобильных от ушных раковин (Как неузнаваемо выросли мальчишки!), с достоинством потенциальных производителей гордясь зловещей оттопыренностью пиджаков . . . Тут же мелькали раскрученные эстрадные певички в окружении

респектабельных молодых сугенерского вида ...

Не стихал светский разговор – вокруг политики, выборов, депутатов, губернаторов ... Назывались суммы – запредельные, в нашем понимании ...

Наконец все утомилось. Отвалившись от доблестно побежденного и поверженного стола, единственно являвшего символ разрушения на этом празднике жизни, несколько осмелев и охмелев, мы окружили нашего благодетеля. А дядя Рамзай, утопая в глубоком кресле у камина и задрав ноги на пуфик навстречу жару, сыто ковыряя в зубах и поминутно сплевывая в огонь, делился своими Соображениями ...

Мы с умилением внимали словам нового Кремлевского Мечтателя, и чувство достоинства медленно возвращалось в нас ...

Нет, все было не зря! Это мы вложили в светлые уста дяди Рамзая эти слова надежды. Это наше воздействие, наше воспитание породили востребованность и понимание ... Разве в потертых штанах дело? Духовность – вот главное, ценное и непреходящее ... И шипящие паровые фонтанчики плевков дяди Рамзая салютуют самому радужному будущему, в котором нам есть, есть место!..

Может, и мы потребуемся в этом вертепе! Может, вернется жизнь в наши хилые тела с дрожащими коленками и с наивными мечтаниями? Может, нажрут-ся-наворуются, наконец, и, рыгнув, оглянутся окрест, и взалкают пищи духовной, – эдакого чего-то воздушного и эфемерного? Может, пригласят поучить их чад-наследников, этих пухленьких мальчишей-плохишей, уже презрительно гримасничающих в сторону зачуханной интеллигенции? А может, и нас покатают в джипе-"чероки"? ...

Мы мужественно перевели все сказанное дядей Рамзаем с поэзии блатного жаргона и сформулировали десять (всего лишь десять!) ярких пунктов, сулящих блистательное применение нейросетевых технологий.

16.1. Служба безопасности

После долгого бдения и дележа средств в Международном Фонде

Вспомоществования Семьям Жертв Российской Науки Прошлого Века (МФВСЖРНПВ), утомленный и удовлетворенный, вы возвращаетесь среди ночи домой. Тихо, крадучись, вы открываете дверь квартиры и включаете свет в прихожей. Но вы совсем забыли о той встрече, которая ждет вас с Несимпатичным Искусственным Существом (рис. 16.1), включенным в электросеть и выставленным для охраны Ваших близких от непрошенных ночных гостей-грабителей! "Здравствуй, папа! – произносит мощный железный голос. – А почему у тебя на лице губная помада?!"

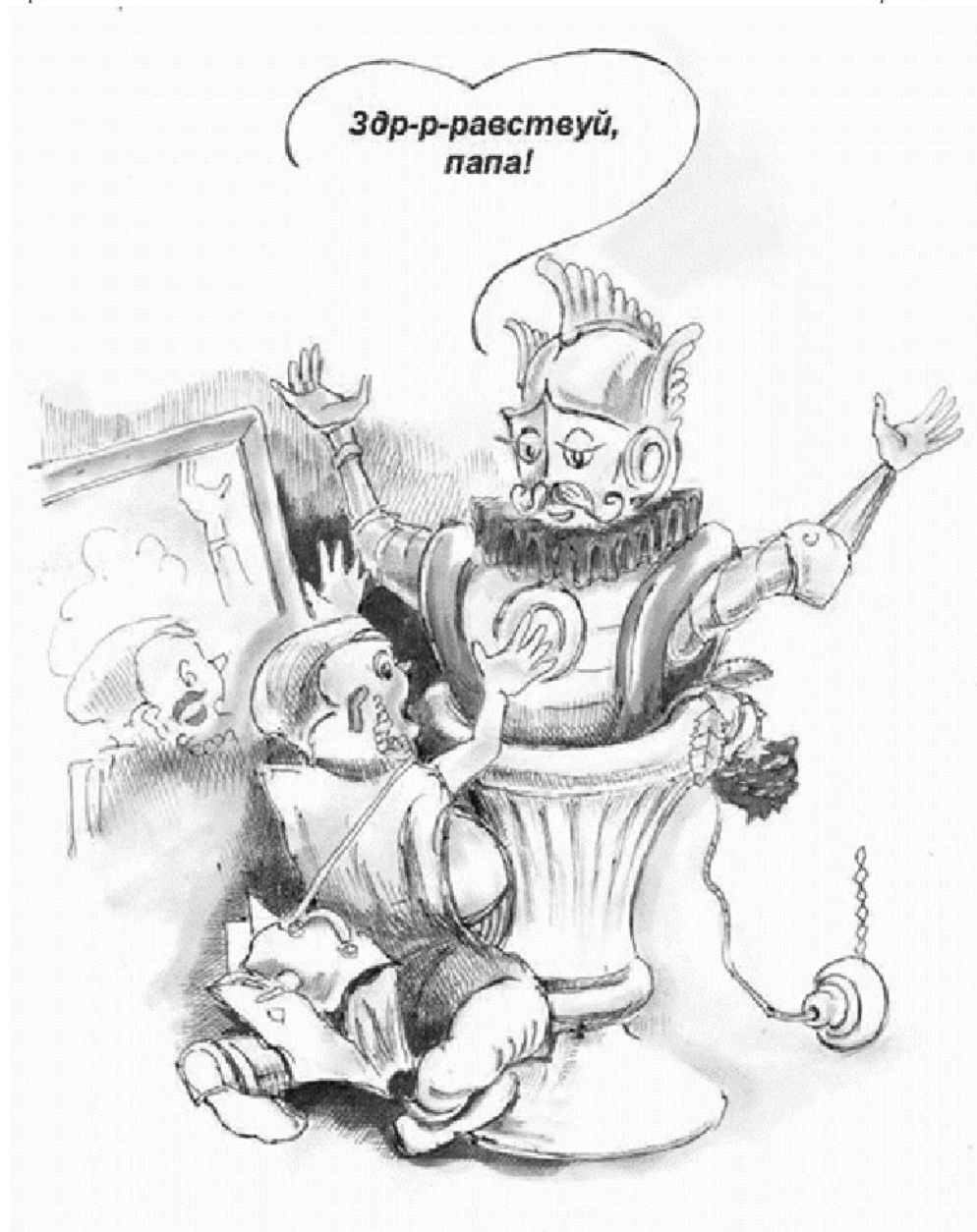


Рис. 16.1. Охрана

Да, охрана покоя семьи, противодействие взломщикам-грабителям, – актуальная задача нашего сложного времени. Мы справедливо предполагаем развитие рынка средств автоматического распознавания

"свой – чужой", находящего все большее число покупателей-приверженцев, по мере роста объявленного благосостояния народных масс.

Как же может быть устроено такое средство охраны-предупреждения?

Отбросив эстетическую наружную часть, несомненно, выполненную со вкусом и выдумкой, заглянем в "нутро".

Основой логической деятельности Существа, конечно же, является *нейросеть* (хотя обработка видеоввода может оказаться гораздо сложнее).

Представим себе многоуровневую *нейросеть*, первый *логический уровень* которой подбирает "словесный портрет" на основе признаков изображения, поступившего на его экран. Не следует удивляться приемам криминалистики, в совершенстве овладевшей подходом к идентификации преступника. В результате анализа, возможно, с двух точек – анфас и в профиль, на этом уровне может быть установлено:

1. волосы присутствуют, черные, шевелюра пышная, на 5 баллов;
2. лоб высокий, залысины есть, поперечные морщины есть;
3. брови густые, с разлетом, сросшиеся на переносице, № 8;
4. нос орлиный, № 4 и т.д.

Конечно, баллы и номера не выставляются. Они означают только, какой нейрон промежуточного выходного слоя возбуждился максимально. Да и это нам знать ни к чему. Просто произойдет какое-то распределение возбуждений между нейронами промежуточного выходного слоя той части нейросети, что обучена распознаванию признаков. Возбуждение этих нейронов, как нейронов *входного слоя* второй части нейросети, второго логического уровня, должно в результате обучения привести к существенному отличию "папы" от "слесаря дяди Юры", приходившего днем проверить нетекущий кран. То есть должно привести к идентификации – кто это (или это "свой", или "чужой"). Можно предусмотреть и побочные признаки: чрезмерно красный нос, подозрительные красные пятна на овале лица вне носа и др. При формировании составного ответа голосом (на третьем логическом уровне?) могут быть учтены и эти дополнительные признаки, как это

мы и предположили вначале.

И мы видим путь, по которому можно гнать и гнать лошадь фантазии, соревнуясь во все более и более фантазмагорических предложениях.

Например, мы можем представить себе медведя с золотым блюдом в дверях шикарного ресторана, приветствующего завсегдатаев словами: "Дорогой и любезный Иван Иванович! Отстегни от щедрот своих, пожертвуй, душа, на хлеб насущный российской науке!"

А увлекшись окончательно, мы можем представить себе парк фантазмагорий, населенный чудовищами, реагирующими на Ваше появление, ваш облик и ваше поведение. "Оживлены" могут быть и фонтаны, и элементы ландшафта, и т.д. и т.д. – так, как это рассмотрено ниже.

16.2. Парк фантазмагорий

Представим себе парк развлечений, в который мы заходим, желательно – группой под руководством гида-экскурсовода. Намотившись по аттракционам и желая наконец протянуть отекающие ноги, мы вступаем (по приглашению коварного гида!) в дивный тенисто-прохладный, ласково журчащий уголок с причудливым ландшафтом. И вдруг!..

... Выползают (рис. 16.2) из-за трепещущих кустов и камней монстры-крокодилы, вздымаются головы ящеров и тянутся к нам. Лохматый неандерталец ладит каменное орудие, привязанное жилой к сучковатой палке. Подымается на свои куры лапы избушка.



Рис. 16.2. Парк фантазмагорий

Гигантской головой оживает скала, покрытая нежными незабудками. Деревья и кусты трясут ветками – радуясь или возмущаясь. Фонтаны воды преследуют нас . . . Наше замешательство, паническое движение

вызывают ответную реакцию: "население" смеется, сердится, радуется, угрожает. Явно проявляются симпатии и антипатии . . .

Рассмотрим возможный принцип действия отдельного "объекта" (рис. 16.3).

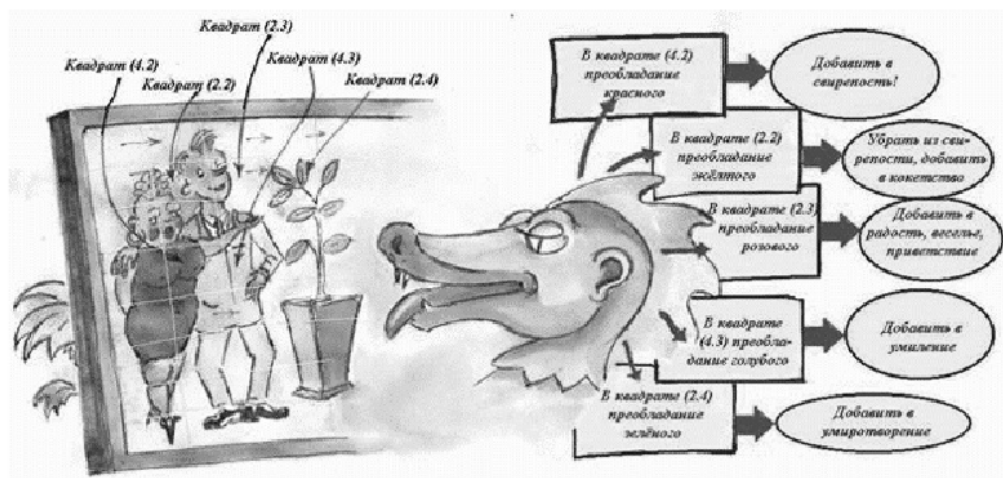


Рис. 16.3. Действия "объекта"

Объект контролирует сектор обзора. Электронно-оптическое устройство "зрения" может быть не обязательно расположено в глазах, а упрятано в более неподвижных "органах", например на груди.

Весь сектор обзора разбит на элементарные сегменты, образованные телесными углами в 1-2 градуса.

До последующих усовершенствований, первоначально должна быть предусмотрена реакция только на преобладающий цвет в элементарном сегменте обзора, на это – "Каждый Охотник Желает Знать, Где Сидят Фазаны". Например, красный цвет должен приводить к запуску комплекса программ обращения в ярость и к повторному обзору строки, чтобы усилить эту ярость при последующем раздражении. (– Женщина в красной кофте! Отойдите от крокодила немедленно!..)

Желтый цвет может вызывать кокетство и эротическое возбуждение. Зеленый – умиротворение. Коричневый – ритмичный шаг и салютование головой. Голубой – благочинное умиление с

возведенными глазами и т.д.

Обзор сегментов совершается по строкам, с частотой обзора, согласующейся со скоростью обработки компьютером. Эта обработка обусловлена прохождением входного сигнала через *нейросеть* и различной степенью возбуждения нейронов выходного слоя, запуском программ движения, связанных с нейронами выходного слоя, – с параметрами, определяемыми величинами возбуждения, инерционностью такой сложной системы.

Итого, из каких элементарных движений складывается поведение объекта? Расширение-сужение глаз, открывание/закрывание рта, растягивание его в улыбку, покачивание головы, поднятие/опускание рук (если они есть), угрожающий шаг ноги (если вес позволяет), вертикальное/горизонтальное помахивание хвостом, привлекающее движение тазобедренным суставом и др.

(– И технологии позволяют . . . Видал в парке Горького мастодонта-звероящера? А гориллу рядом? Если такую макаку, да не к магнитофону, а – к компьютеру!.. А передачу "Куклы" глядишь регулярно?..)

Как ранее говорилось (лекция 2), в одном такте могут запускаться несколько программ, инициирующих движения "от текущего" состояния. Это должно имитировать суммарную сложную реакцию на ситуацию в секторе обзора с его изменением, обеспечивать ее непредсказуемость, неожиданность и восторг публики. А если представить себе, что таким свойством реакции наделены деревья, скалы, воды и ветры, то такой фантазмагорический фейерверк получится!..

(– С этого ж такие бабки можно поиметь!..)

Что ж, ваше слово, товарищ спонсор!

16.3. Компьютерный человек КОМПИ

Мы видим, какое значение компьютерным играм придает самый богатый человек планеты Гейтс, потрясая игровой приставкой нового поколения перед миллиардами телезрителей. Мы вволю наигрались в холодную войну и можем себе тоже позволить подумать о радости и

удовольствию, а также – о бизнесе!..

С этим человечком мы уже встречались в лекции 2. Сформулируем идею окончательно.

... Все чаще посещает нас мечта жизни: пора объединить все разнообразие направлений творческого применения компьютера как средства, инструмента и объекта искусства. Это может быть материализовано виртуальной структурой под названием *Computer Art Studio* и отражать следующие направления: трехмерная и *плоская анимация*, графика и живопись; туристические объекты и исторические реконструкции; нейрокомпьютерные игры и анимация; литературный и музыкальный дизайн; фантастика и др.

Как видите, вполне справедливо, что благодаря логической цепочке "интеллект → моделирование → искусство", нейросетевые технологии должны занять подобающее им место в рядах Великого Искусства ...

Представим себе отрока, поздним утром вставшего в солнечный день весенних каникул и, слегка протерев заспанные глаза, уже включающего компьютер.

В "окне" появляется милая, упитанная мордашка (рис. 16.4). Тут же, в углу экрана, "мышкой" или "джойстиком" наш "хорошист" с английским уклоном не очень твердо, но вполне узнаваемо рисует ананас. Мордашка, в которой мы узнаем КОМПИ – популярного (пока неизвестного) героя компьютерного монитора, приходит в неопишуемый восторг и радость. Все мышцы его лица, ведающие мимикой и жестами соответствующего морально-психологического состояния, приходят в движение.

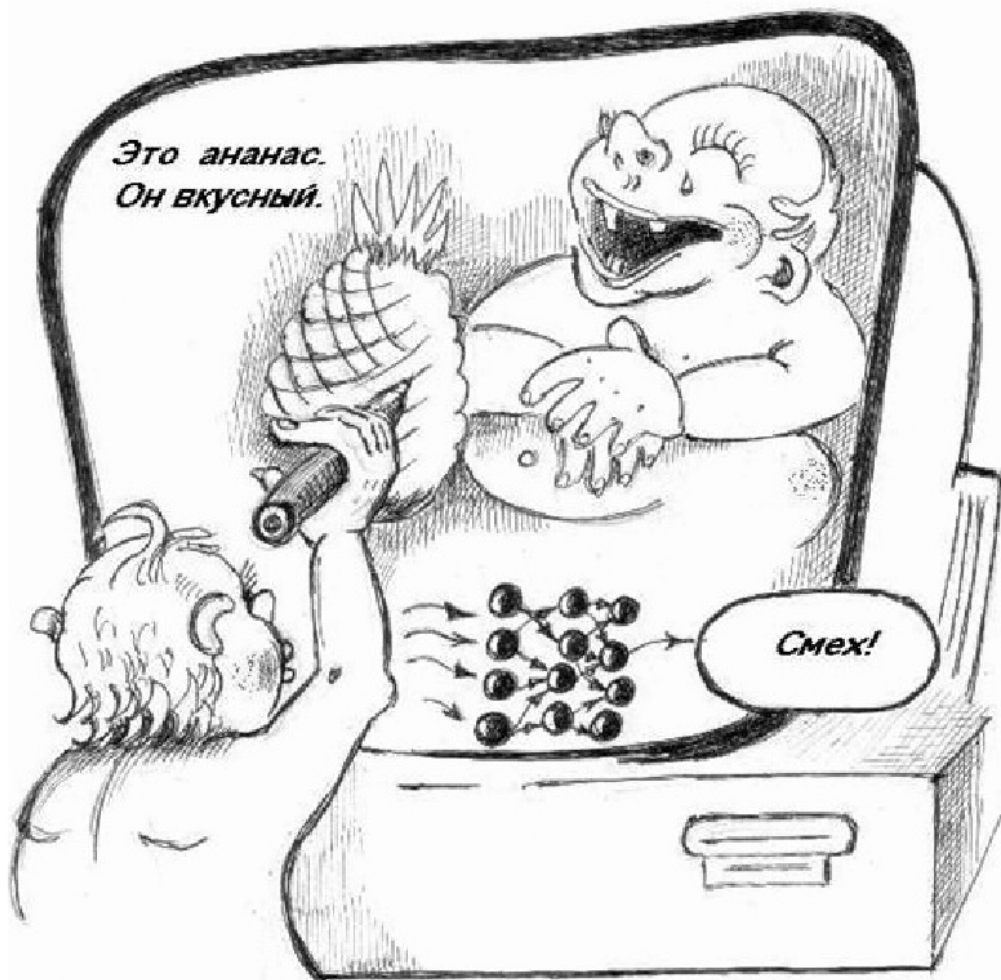


Рис. 16.4. КОМПИ

Тогда наш юный джентльмен стирает рисунок. КОМПИ постепенно успокаивается, приходя в состояние недоуменного ожидания. И вот, после некоторых размышлений, художник изображает . . . велосипед.

"Что это такое?" – вопрошает КОМПИ.

"Это – велосипед, на нем катаются", – следует ответ.

Дальнейшая пауза свидетельствует о том, что происходит обучение нейросети. Ситуация, которая введена на нейроны входного слоя, образующие экран для рисования, связывается (возможно, методом

опорного пути) с нейроном выходного слоя (или – коры). Этот нейрон будет теперь соответствовать эталону – велосипеду. Этот же нейрон, в свою очередь, должен определить возбуждение нейрона, ведающего действием "то, на чем катаются". Запускаются программы, имитирующие действия КОМПИ, и он неуклюже, конечно же, неправильно и смешно – как его учили, например, обращаться с осликом, – подминает под себя копию введенного изображения – велосипед.

Возникает потребность расширения (в условиях, конечно, производящей компании) возможности игры, введя специальные программы имитации красивого велосипеда (распознанного по корявому эталону) и всех действий КОМПИ по езде на нем.

Средствами трехмерной графики, например 3D Studio MAX, производя деформации объема в направлении сокращающихся мышц, можно имитировать действия и мимику. Состав же программных процедур, включаемых в каждом такте управления для такой деформации, и их параметры определяются возбуждением нейронов выходного слоя сети при распознавании заданной ситуации – рисунка на экране. Мы еще не установили точно, может ли это MAX. Однако ясно, что эти действия достигаются средствами двумерной и трехмерной компьютерной графики, на основе управления деформацией объектов. Мы не исключаем, что здесь потребуется разработка новых систем компьютерной графики, основанных на "мышечной" деформации, о чем поговорим далее.

Например, создав объект в плоской или трехмерной памяти, мы можем, дважды "топнув" мышкой, задать точку начала и точку конца "мышцы", зафиксировав в списке эту "мышцу" как новый невидимый объект. Затем мы можем сокращать и растягивать "мышцу", что сопровождается расчетом адресов такой пересылки информации, при которой мы увидим, например, сокращение или растяжение рта в улыбке и другие движения.

Таким образом, с помощью "мышц" мы можем осуществлять программируемую анимацию. Создав ряд таких программ различных действий и осуществив их запуск в зависимости от решений нейросети, мы можем создать комплекс интересных игр и даже сценических

постановок.

Таким образом, развитие игры, выпуск все более новых модификаций, расширяющих возможности, может в конце концов привести к весьма сложному и многообразному взаимодействию юного существа, влюбленного в компьютер, и компьютерного человечка, живущего в нем.

Можно представить себе и более широкое полотно . . .

Представим себе построенные в ряд компьютеры локальной вычислительной сети. Мы сидим за отдельно стоящим компьютером и "мышкой" на его экране рисуем различные предметы: сладкие, горькие, радующие и неприятные. И начинают оживать на мониторах зверюшки, целый зоосад. Они радуются морковке, катаются на велосипеде, огорчаются при виде неприятного, учатся новому и т.д.

И это не только весело. Это – серьезный и важный принцип управления, который может быть использован во многих производственных и зрелищных приложениях.

Дерзайте, предприниматели-программисты! Мы раздаем идеи бесплатно!..

16.4. Диагностика

Представляется, что диагностика – самое прямое назначение рассматриваемых *логических нейросетей*. Действительно, именно в этой области непосредственно проявляется принцип логического мышления на основе причинно-следственных связей вида "если . . . , то . . . ".

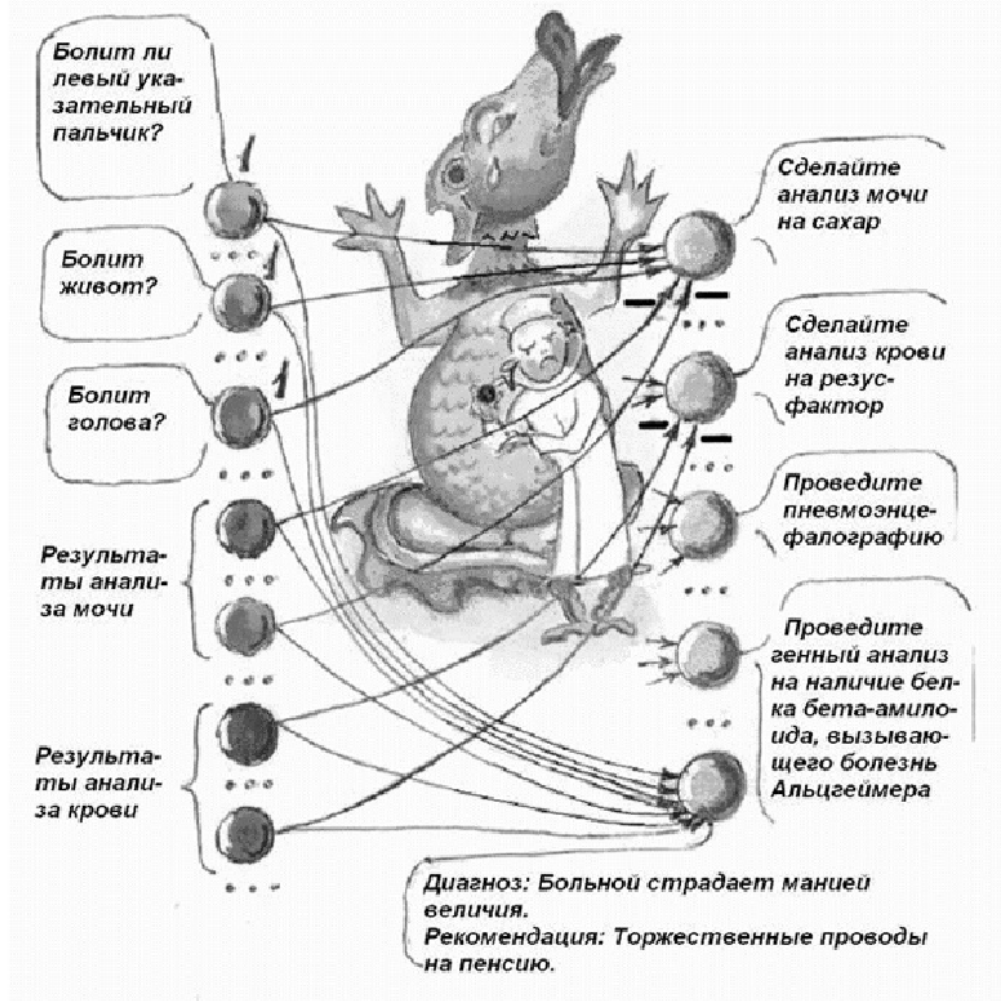


Рис. 16.5. Диагностика

Это относится как к техническим, так и к медицинским системам. Более того, практически всегда отсутствующая полнота информации, зачастую ее противоречивость, нечеткость проявлений, "зашумленность", приводят к актуальности принципа ассоциативного мышления, основанного на опыте коллектива экспертов (в медицине – консилиума).

Рассмотрим фрагмент проекта системы диагностики, которая призвана решить пример, предложенный в разделе 5.3 (рис. 16.5).

Рецепторный слой отображает жалобы больного, симптомы и результаты анализов. Нейроны выходного слоя связаны с высказываниями (текстами), определяющими диагноз или рекомендации дальнейших действий по его уточнению.

Итак, рассмотрим жалобу { болит голова, болит живот, болит левый указательный пальчик }.

По-видимому, этой информации недостаточно для вынесения приговора. Должны быть рекомендованы дальнейшие действия медицинского персонала по дифференциации диагноза. Такие действия связаны с максимальным возбуждением нейронов выходного слоя из области дальнейших рекомендаций. В частности, здесь же формируются запросы о ковырании в носу на сон грядущий, о чистоте рук и т.д.

При выполнении этих рекомендаций, т.е. при возбуждении рецепторов, их отображающих (например рецепторов, соответствующих результатам анализа мочи), использованное возбуждение нейронов выходного слоя снижается с помощью связей с отрицательными весами.

Таким образом, учитываются все рекомендации по стратегии поиска диагноза. Окончательный диагноз определяется тогда, когда максимально возбудится нейрон выходного слоя из соответствующей области.

Данный пример свидетельствует о той грандиозной и только коллективной, технически оснащенной глубине познаний человечества, воплощенной в столь несложной *обученной нейронной сети*.

16.5. Тестирование в сфере образовательных услуг

Работникам высшего образования хорошо известны насущные проблемы в условиях их деятельности:

- общественно низкая значимость труда, выражающаяся в презрительной кличке "бюджетник";
- отсутствие непосредственного влияния уровня образования на надежды, связанные с будущим жизненным успехом студента;
- профанация идеи платного образования, поставившего

преподавателя в зависимость от милости студента-"платника" (рис. 16.6);

- предельное повышение нагрузки и т.д. и т.п.

В этих условиях "головная боль" не покидает несчастного преподавателя весь день такой его благородной, творческой и гуманистической деятельности.

Понимая преходящий характер исторических трудностей, мы спешим на помощь. Мы хотим хоть как-то снизить психологическую нагрузку доцента-профессора, дать ему простой и ясный механизм для формализованного, единственно верного принятия разнообразных решений. Помочь хотя бы при проведении тестирования, в том числе – в период экзаменационной сессии.

(Мы не собираемся строить систему принятия решений. Мы освещаем и без того светлые идеи дяди Рамзая.)

Ода платному образованию

Рис. 16.6. В защиту платного образования

Ограничим *факторное пространство* для создания базы знаний следующими событиями:

$A = \{\text{множество студентов, согласно списка группы}\};$

$B = \{\text{студент- "бюджетник"},$
 $\text{студент- "целевик"},$
 $\text{студент- "платник"}\};$

$C = \{\text{множество тестов, упорядоченных по невозрастанию сложности}\};$

$D = \{\text{множество возможных баллов, получаемых студентами на прошедшей ранее аттестации}\}.$

Множество принимаемых решений по тестированию студента:

- Какой по сложности тест предложить студенту (в заданной точке *факторного пространства*) первым?
- Что делать, если он: а) ответил правильно, б) ответил неправильно? Указывает ли это на то, какую стратегию опроса продолжать далее?

Здесь три возможности:

- закончить тестирование;
- продолжить тестирование в сторону повышения сложности тестов;
- продолжить тестирование в сторону понижения трудности тестов.
- Какая оценка должна быть зафиксирована в зависимости от обрабатываемой точки *факторного пространства* и от результата тестирования?

Представляется, что адекватная *логическая нейронная сеть* является однослойной. Однако длительный процесс тестирования диктует необходимость обратных связей (подобно системе диагностики). Действительно, если вы нечаянно задали студенту-*"платнику"* трудный тест, вам на это должны мягко, но непреклонно указать. Система может с помощью обратной связи снизить допустимую сложность теста на входе, потребовать смены теста. Аналогично, в этом же случае она должна ограничить ваше рьяное стремление докопаться до истины и оценить знания по заслугам.

Несомненно, стратегия тестирования успевающего студента должна отличаться от стратегии тестирования *"среднячка"*. И т.д. и т.д.

Как видим, подобная СПР не только позволяет скрыть за формальной оболочкой наши вынужденные несправедливые действия, но и значительно поднять настроение философским отношением к действительности.

16.6. Графический диспетчер движения поездов

Ранее мы рассмотрели замечательную железнодорожную игру, основанную на современных воспитательных ценностях катастроф и насилий. Ну а серьезно?

Традиционно, расписание движения поездов через перегон, контролируемый одним диспетчером, составляется графически (рис. 16.7), в виде "картинки" с подразумеваемой системой координат. Горизонтально – время, вертикально – расстояние от одного крайнего пункта до другого, с указанием промежуточных. Тогда график движения каждого поезда – наклонная линия, с переменным углом наклона, с учетом остановок. Общий угол наклона линии соответствует направлению движения поезда через перегон. Несколько поездов, следующих через перегон, порождают сложную "картинку" их обслуживания с учетом приоритета (литера), возможного схождения путей в один, длины поезда (с учетом длины путей станций), необходимой скорости следования, запрета остановок.

Диспетчер должен уметь "развести" все поезда так, чтобы за счет ожиданий одних поездов на пунктах перегона провести другие поезда, выдержав приоритет: скорый, пассажирский, сквозной, сборный и т.д., – по возможности избежав столкновений и остановок вне отдельных пунктов.

На рисунке показан график движения поездов на однопутном перегоне. Здесь кривая 1 соответствует скорому поезду, кривые 2 и 3 – двум сквозным поездам. Эти три поезда следуют в одном направлении, из пункта А перегона. Кривая 4 соответствует сборному поезду, следующему в противоположном направлении из пункта В.

Составление расписания – оптимизационная задача (оптимизируются известные *техничко-экономические показатели*) значительной трудоемкости. Поэтому до сих пор диспетчер фактически работает

"вручную", хотя разработаны некоторые эвристические алгоритмы. Можно представить себе мастерство, интуицию и опыт диспетчера, демонстрирующего настоящее искусство.

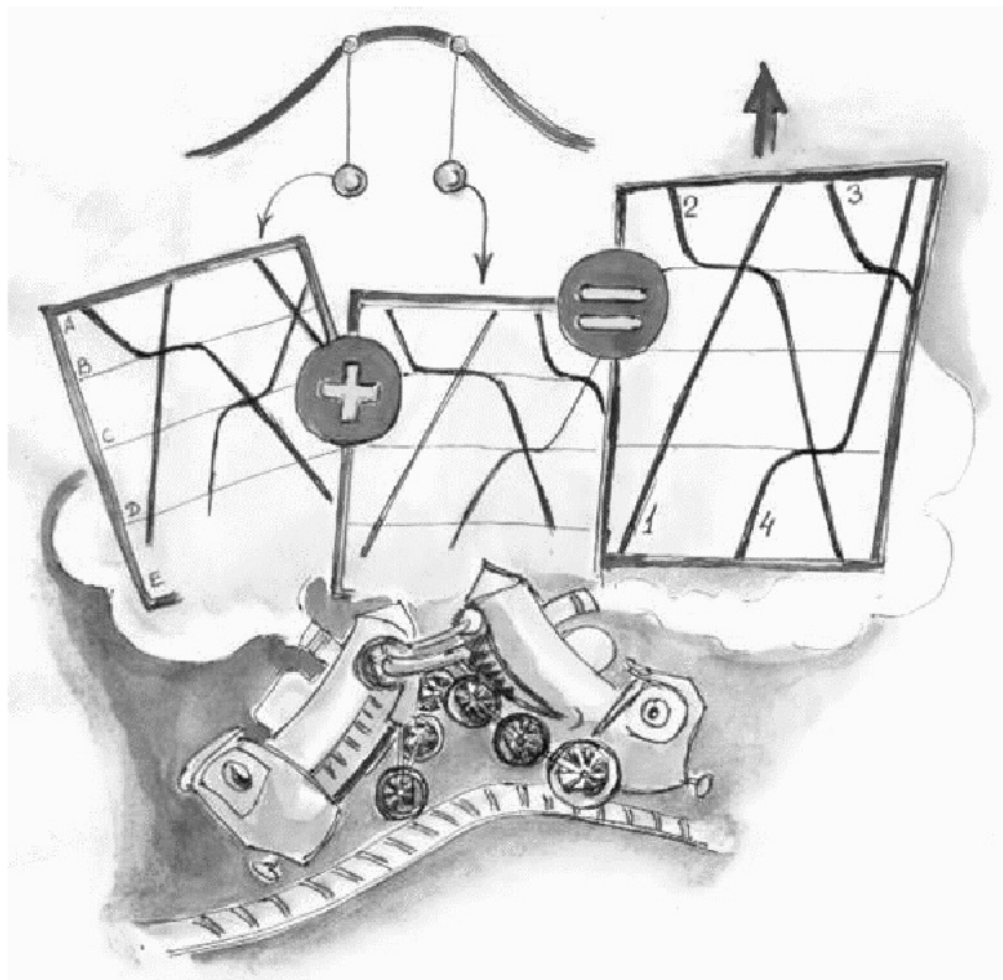


Рис. 16.7. Синтез расписания движения поездов

Мы имеем дело с *трудно формализуемой задачей* и с необходимостью расширения интеллектуальных возможностей диспетчера. Конструируя *нейросеть*, мы должны выделить основные события, характеризующие исходную ситуацию. Относительно этих событий у нас должен быть ясный ответ вида "да – нет", "1 – 0".

Как бы мы могли автоматизировать решение этой задачи (к сожалению

– для каждого конкретного перегоа), на основе богатого опыта диспетчера-эксперта?

Во-первых, мы должны определить понятие "ситуация", выделив и систематизировав все ее показатели и характеристики: максимальное количество поездов, следующих через перегон в разных направлениях, и допустимые интервалы следования, типы поездов, различные сочетания этих типов, установленные скорости движения и допустимые отклонения, разрешенные остановки, длину состава (при известной длине резервных путей на пунктах перегона). Мы должны научиться задавать входные ситуации на *входном слое* нейросети.

Во-вторых, мы должны выделить эталонные ситуации и, подобно эксперту, сопоставить им картинки, похожие на представленную на рисунке. То есть за каждой картинкой мы должны закрепить нейрон выходного слоя сети и проложить трассу – опорный *путь возбуждения* от каждого эталона к соответствующей картинке.

Нейросеть позволит нам уйти от статичности экспертной системы (фактически, таблицы) к динамике, позволяющей по любой случайной ситуации получать ответ на вопрос: "какая картинка в большей степени следует этой ситуации?"

Однако такой простой ответ не может исключить аварии. Т.е. на этом анализ ситуации закончить нельзя. Мы не использовали пока всю информацию, которую нам сообщила *нейросеть*.

Какую же информацию нам следует использовать?

Во-первых, сравнение величины возбуждения нейрона выходного слоя (хотя и максимальной) по предъявленной ситуации с величиной возбуждения этого нейрона по эталону может свидетельствовать о *степени доверия* полученному ответу. Если эта разница лежит в пределах экспериментально установленной точности, то считаем, что ответ найден. Если же мы считаем, что точность недостаточна, мы можем построить (автоматически, конечно) приоритетный ряд, выделив нейроны выходного слоя с близкими (ближайшими) величинами возбуждения. Эти величины возбуждения нейронов образуют веса, с которыми должны быть учтены соответствующие этим нейронам картинки.

Во-вторых, должна быть решена проблема формирования "средней" картинку по нескольким картинкам с их весами, где основные элементы картинок должны сместиться в некоторое среднее положение. Только тщательное экспериментальное исследование может ответить на вопрос о возможности и конкретной практической применимости такого решения проблемы.

Например, на обозреваемом отрезке времени поезд появляется на контролируемом перегоне. Вопросы:

1. Сколько поездов на обслуживании

- $n = 1?$
- $n = 2?$
-
- $n = n_{\max}?$

2. Момент появления поезда

- $t = 1?$
- $t = 2?$
-
- $t = t_{\max}?$

3. Направление движения

- от пункта А?
- от пункта В?

4. Тип поезда

- скорый?
- товарный? и т.д.

Так – по всем поездам на отрезке времени планирования. Таким образом, на основе обстановки, которая ожидается на период планирования, необходимо составить сложную "картинку" – график движения.

Однако мы говорим об *ассоциативном мышлении*, реализуемом нейросетью, о возможности нейросети интерполировать принимаемое решение. Что означает такой ответ нейросети: "На 60 из 100 это – "картинка" А, но на 30 из 100 это – "картинка" В, а на 8 из 100 это – "картинка" С"?

Такой ответ наводит на мысль о возможности автоматического выполнения операций над "картинками". Но возможность операции (в данном случае – получение "средней картинки") должна быть строго обоснована непрерывностью основных параметров, отсутствием скачков. Например, кажущаяся непринципиальной разница в один вагон может привести к тому, что на станции Пырловка поезд не впишется в длину запасного пути.

"Средняя картинка" может быть получена на основе, например, зависимости

$$\text{Средняя картинка} = \frac{0,6 \times \text{Картинка}A + 0,3 \times \text{Картинка}B + 0,08 \times \text{Картинка}C}{0,6 + 0,3 + 0,08}$$

где "+" – условная операция наложения "картинок", "×" – применяется с весом данной "картинки" и определяет долю ее участия в формировании "картинки"-результата.

Графически это может означать направление смещения, растяжения и сжатия, в общем – деформации основных общих элементов "картинок" А, В, С для получения результата.

Кстати, на основе такой операции преобразования картинок можно получать среднестатистические образы населения стран и регионов, этническое прогнозирование и другие интересные вещи.

В любом случае работа нейросетевого диспетчера должна сопровождаться работой анимационной модели, красочно имитирующей движение поездов по сформированному графику и эффектно, в деталях воспроизводящей возможную катастрофу. Это может служить самостоятельным предметом развлечений диспетчера (гражданина) на его трудовом посту. В серьезном же применении такая модель может служить средством обучения нейросети.

А потому всегда должно быть предусмотрено окончательное принятие решения человеком!

16.7. Печать рукописи

Даже в пору, когда чистописание было одним из главных предметов первоклассника, воспитывающих красоту и порядок (несомненно, стимулирующий внутренний порядок) владения пером № 86 с нажимом, разбор рукописного текста был проблемой для развитого воображения. Мы содрогаясь, видя, с каким разнообразием извращений приемов владения современными изобразительными средствами студенты конспектируют мудрость своих наставников.

По прошествии некоторого времени мы с трудом понимаем собственные каракули, не говоря уж о чужих. А если Время поработало капитально, и ваши старания сравнимы с прочтением Кумранских рукописей? Ясно, что только *ассоциативное мышление* способно восстановить рукописный текст.

Поэтому автоматизация воспроизведения написанного является актуальной задачей.

... И окончательно впадая в детство, мы видим в основе распознавания по признакам все ту же методику чистописания.

Еще в лекции 2 мы обратили внимание на "признаковое" определение букв: а – кружочек с палочкой справа, заканчивающейся хвостиком вправо, б – кружочек с хвостиком сверху, вверх и вправо, ц – палочка с хвостиком вправо, к нему примыкает палочка, а к ней, справа внизу, странный специфический крючок, позволяющий сказать "что-то странное, но обязательно существующее", и т.д.

Представим себе "окно просмотра", сканирующее текст (рис. 16.8).



Рис. 16.8. Распознавание буквы по признакам

Оно "наложено" на *входной слой* нейросети. Связи нейронов *входного слоя* – рецепторов – соединяют их (скорее всего, на следующем же слое) с нейронами промежуточного *выходного слоя* (или на коре), закрепленными за признаками. Это соединение выполнено так, чтобы "захватить" случайно (в зависимости от почерка) отклоненную от эталона в букваре букву и выявить в ней отдельные признаки. Комбинация выявленных признаков на следующем (следующих) окончательно *выходном слое* определяет распознаваемую букву.

Возможно, что при наложении очередной буквы, ожидаемой в сложной вязи письма, нам понадобится процедура разглядывания: увеличение/уменьшение, наклон влево/вправо, – до тех пор, пока при некоторой фиксации образа не начнется их действительное распознавание. Поэтому, в дополнение к попытке "захвата", изображение располагается в "окне просмотра" так, что его "центр тяжести" совмещается с центром окна. Далее, изображение растягивается или сжимается, наклоняется влево или вправо так, чтобы максимально совпасть по размеру и углу наклона с размером и углом наклона эталонов, подававшихся в процессе

обучения. Впрочем, это растяжение/сжатие и наклон могут осуществляться динамически, на основе проб, в процессе распознавания в поисках лучшего угадывания буквы.

Для выявления всех признаков и их относительного положения "окно просмотра" необходимо разбить на сектора, определяющие понятия "слева", "справа", "вверху", "внизу", "внизу справа" и т.д.

Определенную трудность представляют пропуски и пробелы. Например, как определить разрыв окружности буквы с от полной окружности буквы о? Ведь может возбудиться "полный кружок", только с меньшим значением величины возбуждения.

Мы предлагаем наряду с позитивным изображением элементов букв анализировать их негативное изображение. Тогда пустота в соответствии с эталоном будет "кричать" о себе, возбуждать рецепторы, обретет активность. Наличие пустоты мы сможем так же "поймать" посредством возбуждения нейронов, как и видимые линии.

Легко увидеть на основе наших действий, что распознавания букв недостаточно. Не поняв ничего, мы поднимаемся на уровень распознавания слов, включаем в работу контекст. Только таким комплексным методом мы восстанавливаем смысл написанного, и это сулит неограниченные возможности развития и совершенствования нейросетевой технологии прочтения.

... Выслушав данное (да и другие) соображение, мы не стали разочаровывать дядю Рамзая тем, что это давно известно, делается, и существует обширная литература. Мы привели его для иллюстрации естественной взаимной вытекаемости идей, базирующихся на одном основании, стараясь менторски наставить молодое поколение – рекомендовать и заинтересовать.

16.8. Экстренное торможение локомотива

В разделе 2.1 рассматривался простейший пример применения *логической нейросети* для распознавания букв. Было указано на обобщение принципа *перцептрона* Розенблата. Действительно, распознавание выполняется методом "голосования" по суммарной

величине возбуждения рецепторов, на которое "ложится" эталон.

Что, если такой принцип положить в основу распознавания препятствия на железнодорожном пути? Ведь на прямолинейном участке (а именно здесь скорость максимальна!) локомотив "видит" перед собой одну и ту же картинку: сходящиеся в бесконечности рельсы. Всякое нарушение этого однообразия требует если не экстренного торможения, то хотя бы снижения скорости.

Если на пути возникает преграда, то картинка напоминает букву А. При этом задача облегчается тем, что картинка строго привязана в пространстве, буква А отцентрирована и отмасштабирована, т.е. фокусировка уже произведена.

Представим себе некоторый экран (рис. 16.9), образованный рецепторами нейронной сети, подобно рассмотренной выше задаче распознавания текста.

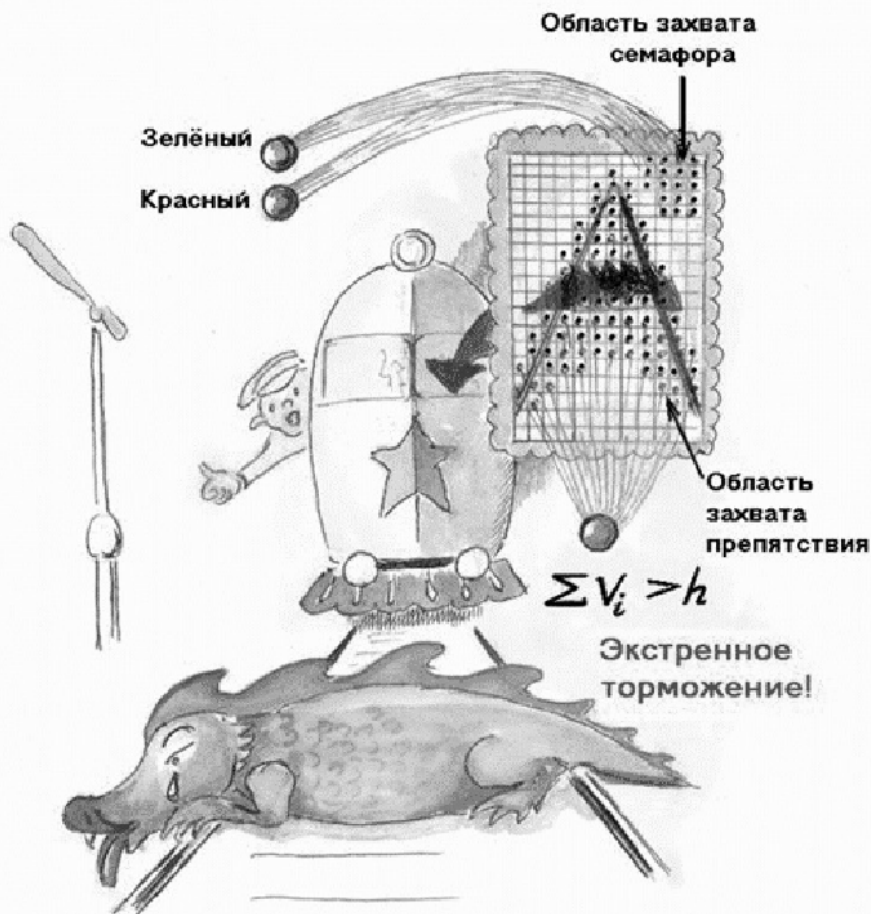


Рис. 16.9. Система экстренного торможения

Рецепторы, возбуждение которых необходимо выделить, связаны с нейроном выходного слоя. Передаточная функция основана на простом голосовании: величины возбуждения складываются, и если образующаяся сумма превышает порог, формируется сигнал тревоги, по которому включается тормозная система.

Принцип крайне прост, однако упирается в хорошее "зрение". Требуется обеспечение высокой контрастности изображения как рельсов, так и

препятствия. Что, если препятствие настолько светлое, что значение возбуждения *выходного нейрона*, наоборот, снижается? Следует поставить "ловушки" как для превышения суммарного сигнала, так и для низкого значения этого сигнала, т.е. ввести два порога. Тогда передаточная функция имеет вид

$$V := \sum_i V_i; \quad V_{\text{тревоги}} = V, \text{ если } (h_1 > V) \vee (V > h_2).$$

Исследуемая картинка предполагает увеличение, приближение вида вдаль – для упреждения реакции. При этом детализация удаленных объектов должна быть выше объектов близких. Это означает, что рецепторы на экране, закрепленные за удаленным изображением (верхняя часть экрана), должны располагаться гуще.

Для большей определенности, по-видимому, следует анализировать как позитивное, так и негативное изображение, формируя результирующий сигнал.

Видимая картинка разбивается на области "захвата", в которых предполагается нахождение интересующего объекта. Тогда та же картинка может использоваться для анализа сигналов семафора, для детального распознавания типа препятствия, для обслуживания поворота и т.д. Области захвата должны быть достаточно "тесными", в идеале включающими, например, только рельсы с незначительным отклонением.

Если путь совершает поворот, то в действие может быть введен другой эталон.

В целом необходимо отметить большой объем экспериментальных исследований для практического применения рассмотренной идеи.

16.9. Защита информации

Мы знаем, какой важностью обладает сегодня информация. "Взломать" финансовую, коммерческую или транспортную систему является делом чести уважающего себя хакера. А информация о движении грузов на железнодорожном транспорте! Не говоря о информации, составляющей

государственную тайну . . .

Защита информации от несанкционированного доступа – исключительно важная задача, актуальность которой все возрастает с развитием сетевых компьютерных и телекоммуникационных технологий. Высокие затраты на ее решение окупаются с лихвой, сопровождаясь тайной воплощения средств.

Однако ширятся и средства "взломщика". Сегодня ему нет нужды "взламывать". Он просто врывается (рис. 16.10) в ваш офис, и под дулом автомата Калашникова, купленного у прапорщика Н-ской воинской части, вы становитесь покорным исполнителем его желаний.

Однако, как бы вы ни старались мобилизовать свои артистические способности, моральное состояние выдает вас чуткому серверу, на расстоянии выполняющему заявки с вашего компьютера в режиме "клиент-сервер".

Наряду со сложной системой паролей, ограничения функций и доступа с вашего терминала, успешно преодоленных (с вашей помощью!) насильником, предпринимателем и террористом, остается что-то непреодолимое, воплощенное в средствах идентификации пользователя и аутентификации. Это – ваш образ, выражающийся в "почерке", в выявленных чертах вашего характера, в общей предпочитаемой стратегии вашей работы. То есть – в ваших индивидуальных чертах, протестированных системой защиты. Система защиты хранит ваш "образ" и периодически сверяет характер вашей работы с этим образом.

"Образы"-эталоны находятся персонально для каждого пользователя на основе тестирования. Какие особенности работы пользователя могут составить его "почерк"? Это могут быть:

- скорость ввода символов на клавиатуре;
- длительность пауз между порциями символов;
- длина порций символов;
- частота использования клавиши delete ;
- частота выполнения специфических операций, связанных с деятельностью пользователя (обращение к механизмам таблиц, формул, рисунков, вставок и пр.);

- частота выдачи запросов к базе данных;
- средняя длительность сеанса связи и т.д.

Одного "почерка", конечно, недостаточно для "образа" пользователя. По соглашению со службой безопасности устанавливаются специальные сигналы и тайные действия, которые необходимо производить пользователю с рекомендуемой частотой, но в случайные моменты времени.

Видно, что основная информация, характеризующая пользователя, включает элементы неопределенности в границах допустимого. Она, скорее, свидетельствует о достоверности, с какой определяются отдельные параметры "образа" пользователя. В этом случае система принятия решений об идентичности пользователя должна основываться на модели *ассоциативного мышления*, т.е. реализовывать искусственный интеллект.

Сервер (или специальный процессор контроля) с высоким приоритетом должен решать основные задачи контроля доступа к БД, связанные с системой паролей и др. На этом уровне обрабатывается определенная, точно заданная информация, не допускающая приблизительной оценки. Однако на уровне фоновой задачи он постоянно должен анализировать ту статистику на основе сложившейся ситуации, которая формируется динамически, при работе многих пользователей.

Пусть $Y = \{y_1, \dots, y_m\}$ – множество параметров "образа" пользователя. Эти параметры описываются величинами различных типов и структуры: *Boolean*, *integer*, *real*, *array* и т.д.

Перейдем к использованию понятия "событие" следующим образом.

Величины типа *Boolean* определяют высказывание о наступлении или отсутствии события.

Переменные типа *real* требуют разбиения всего диапазона их изменения на отрезки. Тогда принадлежность величины этой переменной некоторому отрезку является событием.

Если тип переменной – *integer*, то каждое возможное значение этой

переменной является событием. Однако при большом диапазоне изменения, как и в предыдущем случае, этот диапазон может быть разбит на отрезки.

Если переменная задана массивом (например, множеством допущенных к работе пользователей), то ее равенство элементу множества является событием и т.д.

Корректность *системы принятия решений* требует, чтобы такое преобразование приводило к построению *исчерпывающих множеств событий*.

Так формируется *факторное пространство событий* для множества параметров Y , определяющих "образ" пользователя. Точки этого пространства определяют ситуации.

Теперь необходимо провести тестирование множества пользователей $P = \{P_1, \dots, P_p\}$, а также сообщить им персонально дополнительные контрольные действия. Так, каждому пользователю ставится в соответствие та "правильная", эталонная ситуация, соответствующая его "почерку", секретной сигнализации и ограничениям доступа.

Напоминаем, что практически, при реализации на компьютере или в локальной вычислительной сети, для возможности оперативной обработки, нейронная сеть задается процедурой счета передаточной функции и структурой связей нейронов, отображаемой *матрицей следования*. Поэтому, когда говорят о построении таких сетей, подразумевают их предполагаемое абстрактное изображение, например на мониторе компьютера.

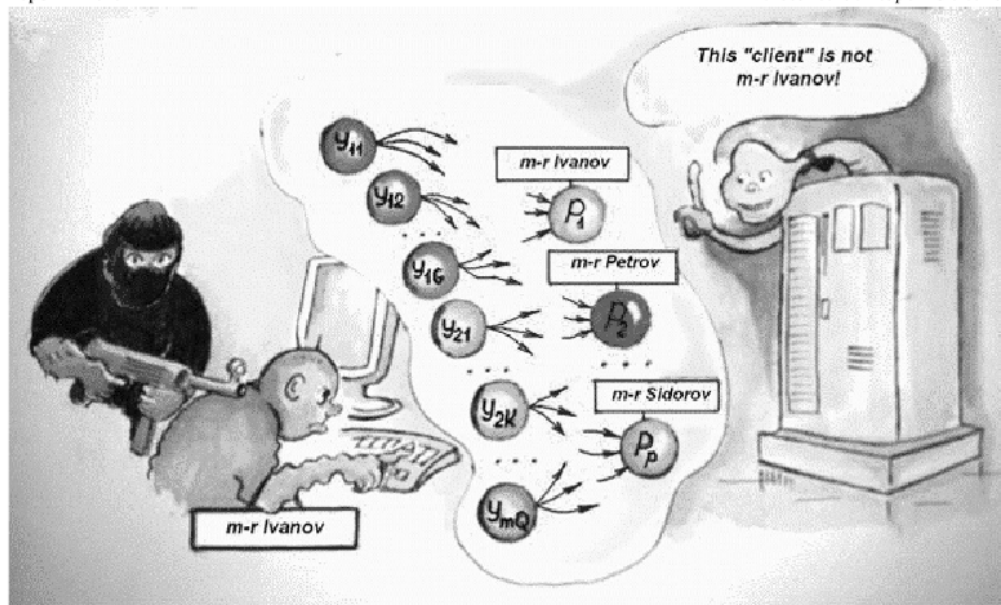


Рис. 16.10. Идентификация и аутентификация

Тогда нейронная сеть строится следующим образом.

Создаются нейроны-рецепторы, которые соответствуют каждому возможному событию, связанному с параметром y_1 . Добавляются нейроны-рецепторы, которые соответствуют каждому событию, связанному с параметром y_2 , и т.д. Так создается *рецепторный слой* нейронной сети.

Для построения единственного (выходного) слоя, необходимо исследовать влияние каждого фактора на процесс принятия решений и выяснить, какие решения должны приниматься по результатам работы нейронной сети. Большая часть решений выносится за рамки нейронной сети и возлагается на дополнительный анализ.

Например, что делать, если в результате проверки выяснилось, что за терминалом № 5 работает не м-р Иванов, а м-р Петров?

По-видимому, целесообразно ограничить нейронную сеть лишь решением о том, на чей "почерк" более всего похож "почерк", реализуемый на данном терминале.

Тогда количество нейронов выходного слоя равно количеству пользователей, а каждый нейрон закрепляется за одним из них.

С каждым нейроном пользователя связываются те нейроны-рецепторы, которые участвуют в эталонной ситуации, характеризующей правильную работу этого пользователя. Таким образом, нейронная сеть оказывается совершенной.

Теперь предположим, что сервер или специально выделенный компьютер периодически, по статистически накопленной информации, производит проверку каждого терминала, за которыми работают "клиенты". При этом известно, кто из множества допущенных лиц должен работать за проверяемым терминалом.

На основе статистических данных формируется исходная информация на рецепторном слое. Ее смысл – достоверность (вероятность) каждого предусмотренного события.

Например, пусть нейроны-рецепторы y_{11}, \dots, y_{1S} закреплены за диапазонами времени между моментами использования клавиши delete. А именно, y_{11} закреплен за диапазоном 0–5 [мин.], y_{12} закреплен за диапазоном 5–10 [мин.], y_{13} – за диапазоном 10–15 [мин.] и т.д. Пусть на основе текущего анализа частоты обращения к клавише delete найдена вероятность реального попадания пользователя в соответствующие диапазоны: $P_{11} = 0,2$, $P_{12} = 0,7$, $P_{13} = 0,1$. Тогда величина $f_{y_{11}}$ возбуждения нейрона y_{11} полагается равной $P_{11} = 0,2$, аналогично $f_{y_{12}} = P_{12} = 0,7$, $f_{y_{13}} = P_{13} = 0,1$.

После формирования значений величин возбуждения нейронов рецепторного слоя с помощью заданной передаточной функции и в соответствии со связями рассчитываются значения возбуждения нейронов выходного слоя. Если более других возбудился нейрон, соответствующий правильно работающему пользователю, сервер (в режиме фоновой задачи) приступает к проверке следующего терминала. Если более всех возбудился нейрон, соответствующий другому пользователю из числа допущенных, или возбуждение всех нейронов не преодолело порог (что свидетельствует о несанкционированном

доступе), формируется сигнал тревоги.

Сигнал тревоги может подаваться не только тогда, когда вместо вас работает кто-то другой, но и тогда, когда ваше моральное состояние порождает серьезное отклонение от запечатленного образа. При этом, как говорилось выше, целесообразно использовать и тайные сигналы, наподобие точек в условном месте, сигналы, подобные завитушкам ответственного каллиграфа и т.д. Возможно периодическое случайное анкетирование на основе единственного вопроса, требующее ответа "да – нет" и способствующее утверждению сложившегося образа.

Возникает вопрос: а при чем здесь *нейросеть*? Ведь все эти проверки можно делать старым, "традиционным" способом. И здесь мы еще яснее осознаем, что *нейросеть* – это средство параллельного выполнения сложных логических предикатов! Параллельного – значит, быстрого! А современный сервер – это *многопроцессорная вычислительная система*, которая успешно может выполнять роль *нейрокомпьютера*. Кроме того, осуществляя *ассоциативное мышление*, по принципу "на что это похоже", мы допускаем (конечно, в каких-то пределах) неполные и неточные данные, заставляющие нас по крайней мере встревожиться и послать администратора или даже охрану с проверкой.

16.10. Сивилла-прорицательница

В век всеобщей информатизации мы, нетерпеливо пропуская конкретику и детали, жадно ловим правду "между строк", ассоциативно формируя предвидение и прогноз: что будет с нами завтра, каковы тенденции и что предпринять?

Формализация обработки огромных потоков информации, в основном в СМИ (а это – оперативная информация), все более привлекает специалистов по прогнозу. Здесь огромные возможности для шарлатанов, и кажется удивительным, как незамеченными остаются "прогнозы" авторитетных, не сходящих с телеэкрана "предсказателей", данные ими несколько лет назад и не заставляющие их покраснеть сегодня.

Поставить прогноз на научную основу – задача современных и честных ревнителей перспективы.

Основой прогнозирования является состав и частота появления слов и выражений, характеризующих политический этап или целый исторический период. Далее следует анализ исторического опыта, определяющий, к чему это приводило в прошлом и к чему может привести впрддь ...

... История помнит замечательные, навязчиво мелькающие, "новые" слова и выражения. Например – "конвергенция", выражающее сомнение и крах режима, "интернациональный долг" – саморазрушающая агрессия, "консенсус" – псевдодемократический, криминальный обвал, "экстрадация" – мировой процесс очистительной ловли тараканов, "суверенитет" – самостоятельный выход из окружения. А что дальше? Дальше – слова "наркомания", "терроризм", "сепаратизм", "курс доллара", "приватизация", "прожиточный минимум", "коррупция" и т.д.

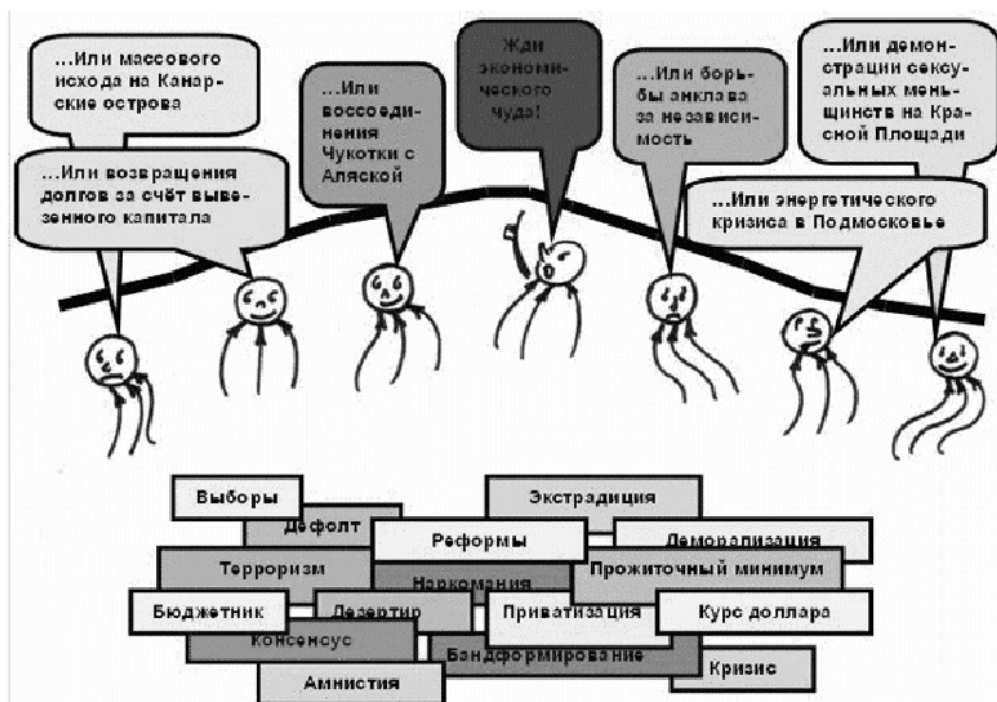


Рис. 16.11. Предвидение

Комбинации и частота следования слов (рис. 16.11) образовывали критическую массу для неотвратимой цепной реакции ...

Итак, как в фантазмагорическом парке: появление каждого "судьбоносного" слова добавляет определенное настроение нейросети. Результирующее настроение определяется распределением возбуждений нейронов выходного слоя – его максимумом, значением среднего, предпочтительным рядом. Частота их появления меняется во времени, обязательно определяя знаменательные события: развал Союза, путч, расстрел Белого Дома, вехи экономического краха, социальные и региональные беспорядки...

– Но ведь подобная методика применялась при определении святости слов Святого Писания!

– Вот посмотрите только заголовки единственно выписываемого мною во все времена "АиФ" № 8 за февраль 2001 года, – послунил палец дядя Рамзай.

Оптимистический ряд	Пессимистический ряд всех оттенков
Мантимер Шаймиев: Не будем строить то, что однажды развалилось*) (стр. 3)	Апокалипсис в одной отдельно взятой стране (стр. 2)
На XXI век нефти хватит (стр. 6)	Новая версия скандального ареста госсекретаря союза России-Белоруссии (стр. 4)
Пятьдесят регионов уже приняли свои земельные законы (стр. 6)	Батальоны просят рубля, Не плачь, мальчонка!, Ордера на мундирах, Эликсир под "кирзу" (стр. 7)
Даже у "исконно русского" КВНа обнаружился зарубежный аналог*) (стр. 10)	Ежегодно в России пропадают 40 тыс. граждан (стр. 12)
Реклама*) (стр. 5, 8, 9, 11 и далее)	Наши видеопираты всех достали (стр. 12)
Репрессии песни "Ландыши" длились 23 года *) (стр. 13)	Каков современный поп, таков и приход (стр. 13)
Хорошей колбасы должно быть много, главное, чтобы она была не	Станет ли Россия братским

из мяса бешенных коров *) (стр. 16)	"могильником"? (стр. 18)
"Не блатной" кроссворд (стр. 21)	Хлебный рынок столицы далек от цивилизации, На улицах – непроходняк, Что делать с бомжами? и т.д. и т.п.

*) Может, мы ошиблись колонкой?

(– А ты что ж, дядя, хотел бы вернуться к дутому оптимизму?

– Нет, браток, – и внушительный палец метнулся ввысь, – я хочу лишь сказать, что демократия не объявляется, демократия воспитывается, воспитывается долго и жестоко! Ты можешь представить себе немца, англичанина, француза, которому по полгода не платят зарплату? А? Вот то-то!)

... Наша задача – не только формально, скажем, по частоте использования тех или иных слов и выражений, но и на основе ассоциативного восприятия воспроизвести динамику социально-экономического, политического и морально-психологического движения общества.

(– Как, в натуре, замелькают слова "наведение должного порядка", "приоритет образования", "показательный процесс с конфискацией", "восстановление государственного сектора экономики", "антикриминальный террор" и другая лабуда, так сразу и поймешь: "Ну, – [конец], пора рвать когти, пока не дошло до катаклизмы . . . ")

Например, желая вскрыть динамику, таким же способом, деля на оптимистический и пессимистический ряды, обрабатываем "АиФ" № 45 за ноябрь 2006 года.

Оптимистический ряд	Пессимистический ряд
Грузия. "Дорогим друзьям" – дорогой газ (стр. 2)	Кто заражает армию дедовщиной (стр. 3)

"Традом" по миру. Южная Осетия готова к свободе, а Грузия – к войне (стр. 4)	Игра в бутылочку. Кто спасет Россию от спаивания (стр. 6)
Полезный совет. Помогите спасти мужа (стр. 5)	Я не понимаю ... почему министр финансов самый бедный? (стр. 7)
М. Жванецкий: Какие дети от секса, а какие – от любви? (стр. 10)	Наболевшее. Наш бизнес порочат грязные деньги (стр. 8)
Кальций без гипноза. У кого не болят кости (стр. 11)	"Титлер умер в ... 1964 году" – утверждает автор скандального бестселлера (стр. 16)
В. Шумаков: Пересадить мозг можно, но... (стр. 12)	Лес рубят и будут рубить (стр. 21)
А. Нечаев: "Храните доллары в рублях" (стр. 13)	Почем гаражи для народа (стр. 22)
День без взяток (стр. 15)	В столице могут появиться обманутые собственники (стр. 25)
Рак предстательной железы – не приговор (стр. 17)	Головоломка от Создателя, или секреты женского организма (стр. 27)
Как дать сдачи учителю? (стр. 19)	Не торопитесь отдавать долги (стр. 37)
Богатство в курсе (стр. 20)	Дыба и кнут все перетрут (стр. 39)
Трансгены – вон из столицы! (стр. 23)	Дареному вождю в зубы (стр. 39)
"Зебру" – пешеходам (стр. 26)	Горькая пилюля поддельщикам (стр. 45)
Залейте в бак... навоза (стр. 40)	Скрытая угроза всему человечеству (стр. 47)
Секрет долгожителей (стр. 49)	Хлеб – удар по почкам? (стр. 48)
Наследство – дар ясновидения. Тайна третьего глаза (стр. 50)	Потный мужик с сигарой ей не нужен. А. Пугачева. Стилист? Юморист? Вот и пойте (стр. 56)
Еж + уж = ? Но не метр колючей проволоки (стр. 53)	Гус Мутко не товарищ? (стр. 58)

Миома... испаряется (стр. 61)	Мобильная импотенция? (стр. 64)
-------------------------------	---------------------------------

Как видим, обработка этого номера не вселяет больше оптимизма, чем пять лет назад. Так что опасения дяди Рамзая пока беспочвенны.

. . . И мы испытываем светлое "чувство глубокого удовлетворения" теми Великими Преобразованиями, которые, несомненно, объявлены Высшим Разумом, но пока недоступны нашему пониманию. И только Третий Глаз, черпая энергию Непознанного, продолжает сообщать нам радостные и тревожные, а порой – вещи видения-сны, связывая нас с Бытием . . .

Компьютерная технология рисунка и ее применение в научных исследованиях и образовании

Исследуется компьютерная технология выполнения и редактирования рисунка для научных и образовательных иллюстраций, как важной сюжетной основы систем мультимедиа, плоской и объемной анимации, интеллектуального отображения управляющих систем и цифрового кино. Основное внимание уделяется юмористическому рисунку. Приводятся рисунки автора

Введение

Раки любят, чтобы их варили живыми.

Л.Н. Толстой. Дневники. Ноябрь, 1897.

Приемы естественных наук, основывающие свои выводы на фактах, самые неудачные приемы. Фактов нет. Есть наше восприятие их. И потому научен только тот прием, который говорит о восприятии, о впечатлениях.

Там же. Май, 1903.

Научная и, в особенности, преподавательская деятельность убеждают в важности средств адекватного восприятия материала, в важности создания впечатлений и ассоциаций, способствующих пониманию и запоминанию. Это – задача современного представления иллюстративного материала, основанного на применении информационных технологий изобразительных средств, средств мультимедиа и даже кино. Современный учебный курс немислим, и это зафиксировано в требованиях, без средств мультимедийного воплощения. Лекции читаются и демонстрируются с помощью мультимедийных дисков. Подготовка средств дистанционного обучения через Интернет требует комплексного применения всех возможных средств видео-воздействия. Условия применения, несомненно, предъявляют требования и ставят задачи по развитию изобразительных средств на основе информационных технологий.

Любая идея применения таких средств начинается с рисунка. С помощью рисунка вырабатывается образ, рисунок подсказывает сюжет, технику его воплощения, да и весь сценарий доходчивого изложения материала. Таким образом, рисунок, сам являясь продуктом методологического подхода, развивает этот подход, выдвигая дальнейшие задачи. Здесь, конечно, многое зависит от ученого – преподавателя, от того, насколько он смог развить в себе образное, *ассоциативное мышление*, каким примерам он стремится следовать.

Творческая жизнь ученого, ученого результативного, оставляющего след и память, не страдает однообразием и узостью интересов. Чаще такие ученые – увлекающиеся люди, обладающие не просто "хобби", а имеющие вторую, а то и третью профессию. Можно перебрать много великих ученых, одновременно, - писателей, музыкантов, художников. Что сказать о великом А.П. Бородине? Бесконечно восхищает Де-Сент-Экзюпери, воплотивший две "противоположные" профессии: утонченный писатель-философ – за штурвалом самолета! Ведь и в той, и в другой профессии он стал первопроходцем. (Кстати: можно ли представить себе "Маленького принца" без "непрофессиональных" рисунков автора?)

Но оттолкнемся от уровня школьного сочинения (а то еще вынесет на художника Адольфа Шикльгрубера), и подчеркнем следующую мысль: отдельные направления деятельности ученого не служат лишь благотворному переключению его внимания для снятия нагрузки и отдохновения, подобно пикианью Шерлока Холмса на скрипочке. Для этого достаточно просто лечь в постель, - спать. Нет: все его увлечения, включая основное, все направления его деятельности питают друг друга, а в значительной степени - одно превалирующее направление. В конечном счете, они служат единой цели. Поддерживают ее с разных сторон. Разнообразным подходом они способствуют решению тех основных задач, которым, по общественному признанию, посвятил себя ученый. Тогда его усилия в познании нового возрастают многократно. Так что не напрасно Шерлок Холмс пикирует на скрипочке в минуты напряженного раздумья, и его автор прекрасно это понимал.

Бытует представление о приверженцах точных наук: дескать, сухие, заформализованные люди – психические уроды. Слагают анекдоты о их рассеянности, непрактичной зауми. Очень часто это относится к тем

"ученым", что играют роль таковых.

Однако "царица" точных наук, математика, справедливо признана естественной наукой. Да, она самая естественная из всех естественных наук, ибо отображает все реалии жизни – как прошлой, так и будущей, и без явственного представления жизненных аналогий она действительно мертва! И продуктивно работать в области, где основой является математика, может лишь человек с богатым воображением.

Все математические объекты имеют реальные аналогии.

...И роятся в голове исследователя странные геометрические фигуры, зацепляясь или отталкиваясь, шумя, бурля и подсакивая... Вот уж фантастические образы теснят и высовываются среди них... Растет напряжение мысли... Широко раскрывается глаз куда-то в сторону... Уже нет слов, только видение подменило вашу цель, вашу задачу... Вы напрягаетесь, вы подталкиваете его, вы торопите его сложиться так, как нужно, чтобы получить ответ...

Фу, черт, - гоните вы наваждение, понимая грань возможностей нормального психического здоровья.

Но надо, надо... И основная жизнеутверждающая догадка осеняет вас: "Без здорового чувства юмора здесь работать опасно. Если все эти движущиеся модели, эти связанные веревками бруски, эти монстры и крокодилы с зачатками мышления, эти летающие болванки будут еще и смешными, возможно, "крыша поедет" не скоро".

Да здравствует спасительное чувство юмора – "вечно живое и всепобеждающее учение"!..

Подчеркнем еще раз, что рисунок, в особенности - юмористический, несомненно, передает динамику, но он статичен. Однако статический рисунок лежит в начале технологической цепочки, определяя свое дальнейшее развитие на основе применения средств анимации и мультимедиа, пробуждая фантазии по созданию кинофильмов – от научно-популярных до игровых. Пройдя такое развитие, рисунок может вернуться на свой исходный рубеж, и на новом уровне стать основой мультимедийных средств иллюстративной и развлекательной поддержки учебного процесса.

Представленные ниже рисунки, использованные уже не только автором в учебном процессе, в книгах и статьях, сгруппированы по темам с необходимыми объяснениями.

Технология компьютерного рисунка

Компьютерные технологии все увереннее проникают в сферу искусства. Мультимедиа породило шедевры, поражающее впечатление и создающие ощущение колоссальных возможностей в будущем. Компьютерные макеты и декорации, спецэффекты и анимация – все это, по-видимому, лишь начало бурного вторжения Компьютера в Искусство. (Утверждается термин Computer Art.) А что обещает скорое нашествие средств и методов искусственного интеллекта?!

И уж тем более, приверженцы точных наук, для которых компьютер – основное орудие и объект труда, обладающие хотя бы зачатками пространственного воображения, используют компьютерные технологии для самовыражения и образной поддержки своих исследований.

Какова же техника современного рисунка на основе применения компьютерных технологий?

1. В основе рисунка лежит акварель. Акварель – не в смысле присущего ей тонкого, элегантного, воздушного, светлого воплощения. Речь идет, скорее об "оформительском" варианте, грубо ограничивающем использование палитры, но ни в коем случае не исключаяющем ее.

Можно выразиться строго: начальное приближение рисунка выполняется с помощью карандаша и школьных акварельных красок.

Однако уже на этом уровне "художник" вправе выразить свою "индивидуальность". Например, предпочтение автора, направленное на юмористический стиль – стиль карикатуры, уделяется четкому контурному выделению объектов рисунка и даже их штриховке – преимущественно с помощью черной шариковой ручки. Таким образом, необходимое оттенение может

выполняться как цветом, так и штрихом.

2. Затем рисунок сканируется для дальнейшего редактирования и развития. Автор убедился в достаточности такой системы, сопровождающей любую современную версию операционной системы Windows, как Paintbrush. Его привлекательность – в доступности, простоте и в отсутствии необходимости длительного предварительного изучения. Тем более, что речь идет не о создании художественных объектов, а только о их редактировании.

Основной недостаток Paintbrush заключается в том, что он использует растровые файловые форматы JPEG и BMP. Это главным образом приводит к тому, что линии, отличные от вертикальных и горизонтальных, имеют ярко различимую ступенчатую структуру.

3. Говоря о редактировании, следует вернуться на уровень рисования. Еще на этом уровне, для сокращения трудоемкости, повышения точности воспроизведения и уменьшения степени зависимости от возможностей средств редактирования, может быть предусмотрена компоновка рисунка на основе его фрагментов. При этом могут быть использованы свойства симметрии, многократной повторяемости элементов, вариантности, масштабируемости.

Таким образом, процесс редактирования зачастую сводится к процессу полной компоновки рисунка. Если к этому добавить еще и необходимость композиционных поисков и исправлений, то редактирование превращается в процесс значительно более трудоемкий и творчески насыщенный, чем процесс предварительной черновой заготовки рисунка.

Рассмотрим некоторые "Библейские" исследования (и это уже не юмор).

Например, Храм Иерусалимский (Ирода) "собран" на основе следующих элементов:

- золотой капитель коринфского ордера (найденный на Храмовой

Горе, возможно, - от храма Юпитера, воздвигнутого позже) для золотой колонны;

- этот же капитель, "растянутый" для внешней мраморной колонны;
- собственно колонна, породившая золотой и мраморный варианты;
- повторяющийся фрагмент обрамления входа золотой лозой;
- фрагмент гроздей винограда над входом;
- фрагмент верхней плиты с орнаментом;
- херувим и фрагмент центральной пальмы для верхней плиты;
- шпиц в единственном экземпляре;
- рисунок половины Храма без приведенных выше элементов, ограниченный осью симметрии, на фоне галереи, а также с учетом жертвенника спереди.



Храм

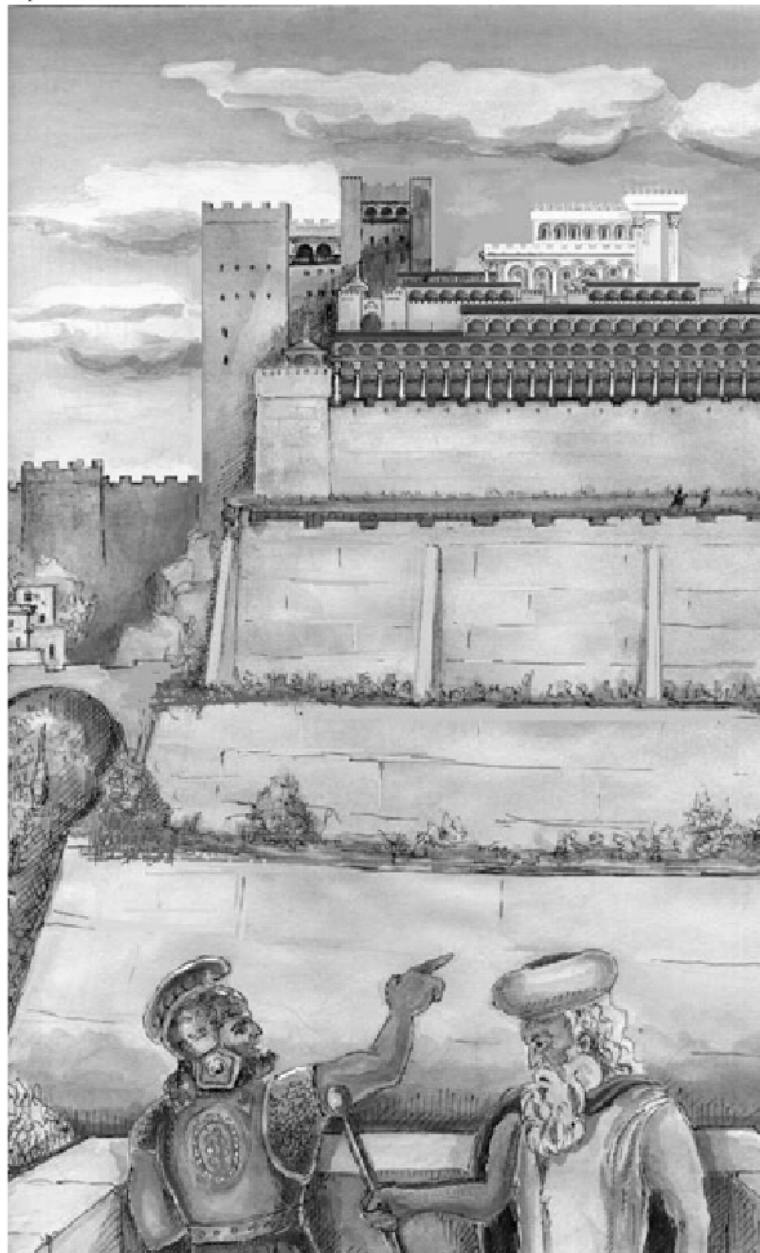
В результате "сборки" сначала создается половина изображения. Затем с помощью операции симметричного отображения создается почти окончательный рисунок. Его остается лишь скорректировать с учетом

несимметричных элементов. В данном случае – с учетом подъема к жертвеннику лишь с южной стороны.

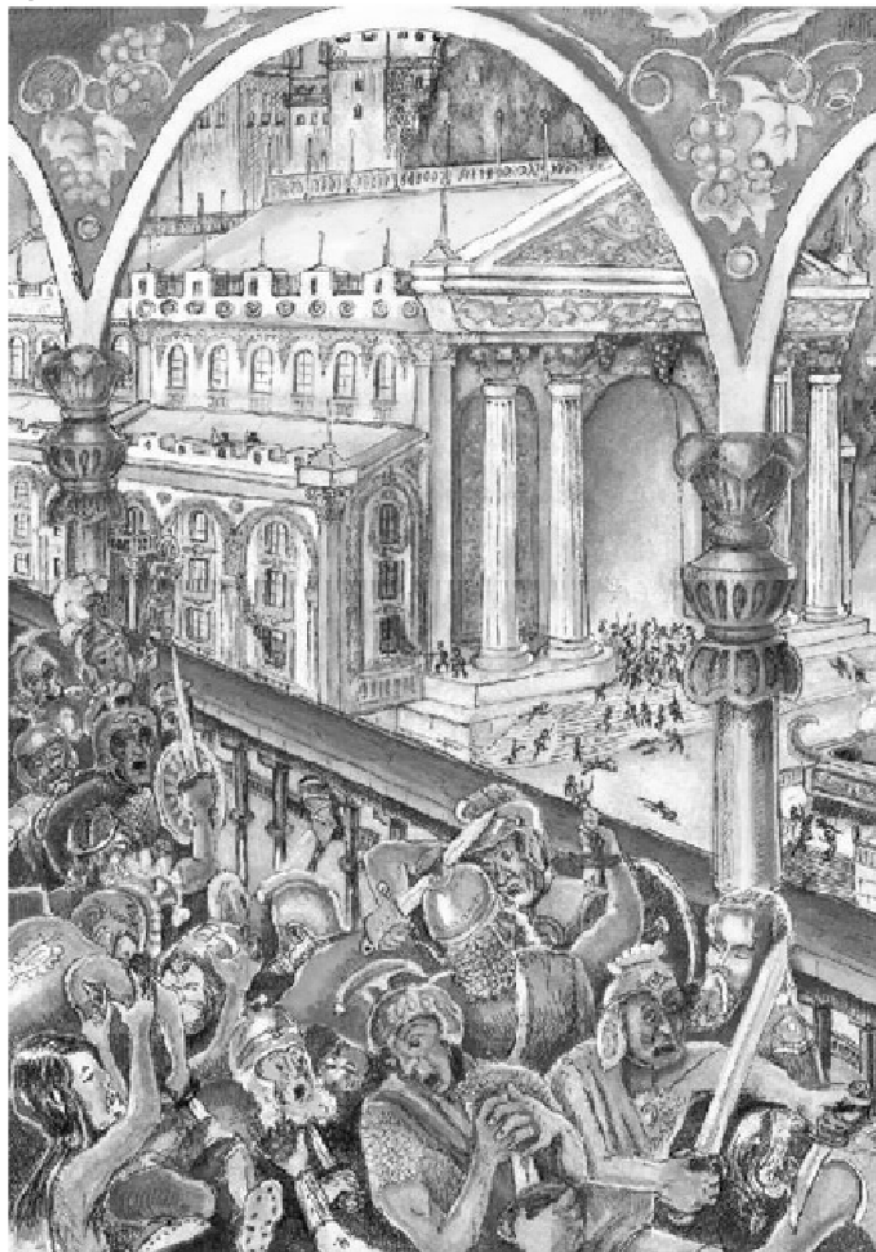
В разработке следующего рисунка, где изображен вид Храмовой Горы, роль редактирования минимальна. Оно коснулось лишь самого Храмового комплекса, где требовалось миниатюрное исполнение деталей. Чтобы исключить неточности в горизонтальной ориентации, несовпадения размеров и окраски одних и тех же деталей рисунка, например, колоннад и окон, целесообразно использовать возможность выделения фрагментов и их переноса – размножения. Такой перенос исключает разнорядность в изображении, вносит элемент регулярности и соблюдения основных пропорций и направлений. Внимательно взглядевшись в рисунок, можно выделить основные фрагменты, его образующие. Однако основная часть рисунка полностью сохраняет изначальный замысел.

Говоря об истории этого рисунка, следует отметить, что первоначально Храмовое здание было другим. Уточнение представления об этом здании с малой трудоемкостью привело к замене соответствующего фрагмента.

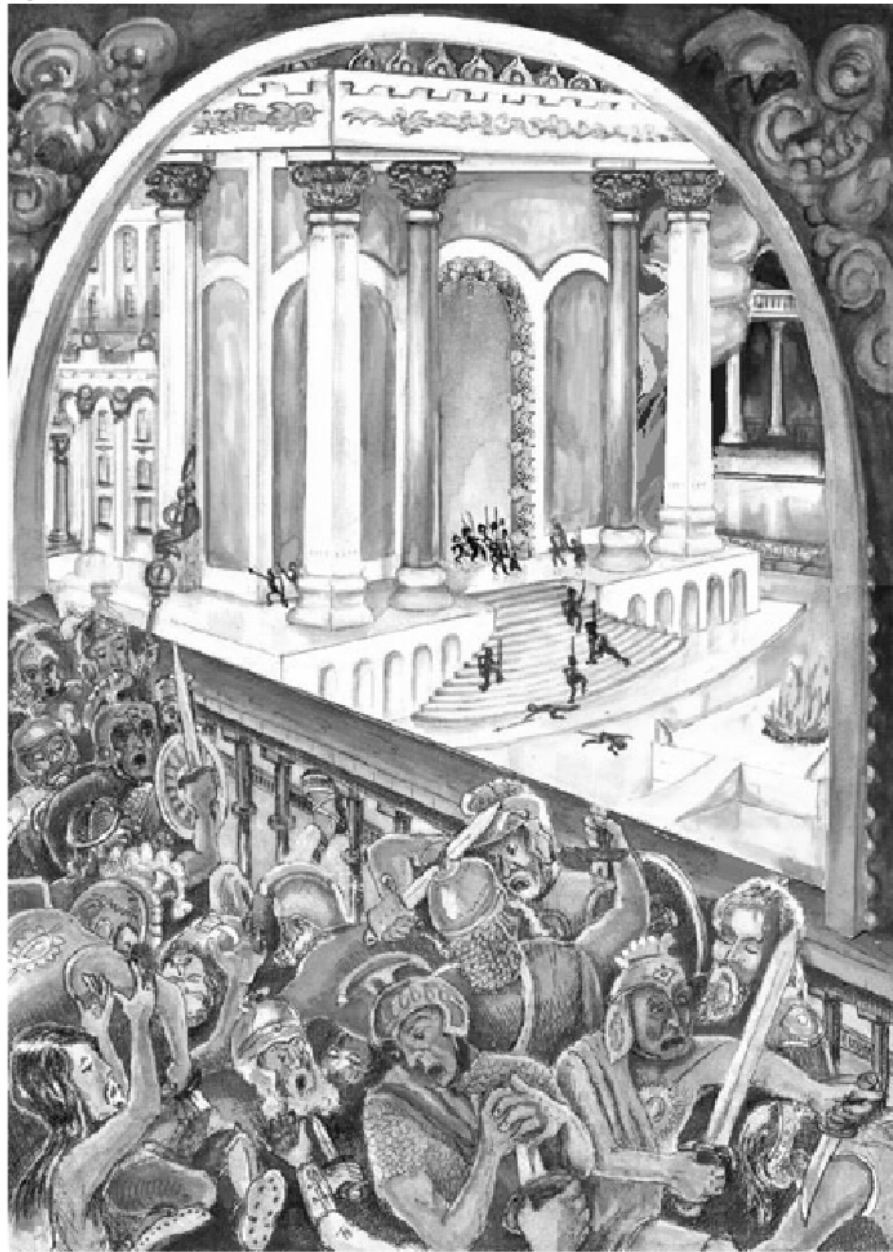
В более значительной степени подобную замену иллюстрируют рисунки, посвященные захвату Храма римлянами. На первом из них фигурирует раннее представление о здании Храма, на втором – окончательное. Рисунок изменен с помощью компьютерной технологии все в том же Paintbrush.



Вид на Храмовую Гору



Битва за Храм (вариант)



Битва за Храм

Рисунок "Святая Святых" выполнен на основе следующей "калькуляции":

- тело херувима на пьедестале – одно;
- крыло херувима – одно;
- Ковчег Завета на постаменте – половина;
- общая декорация (задник, ступени, пол) – половина.

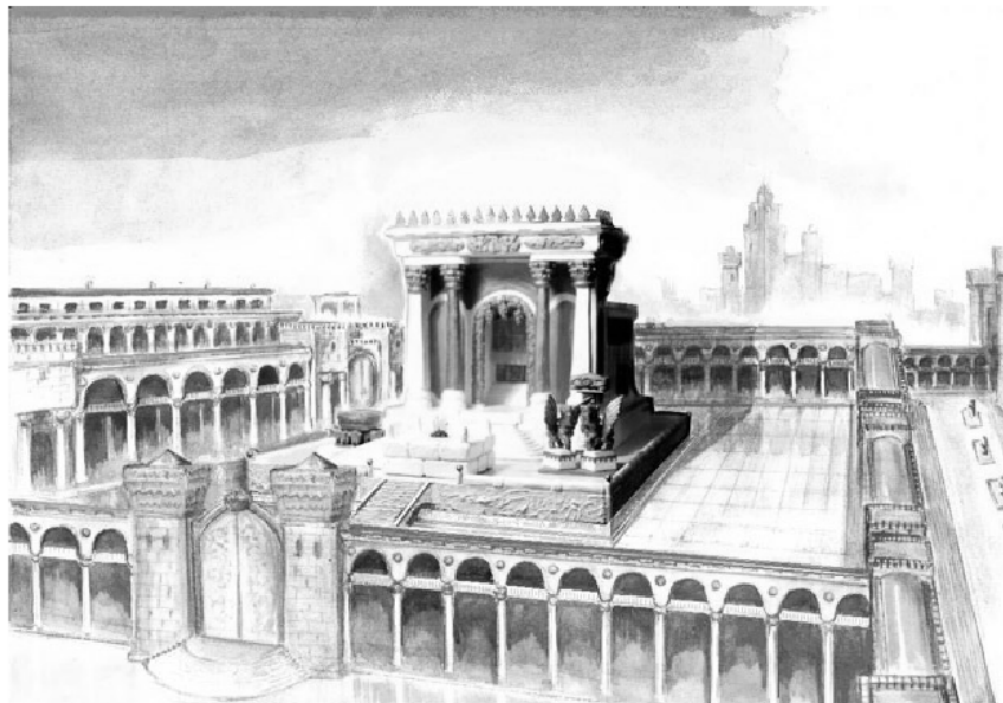
Еще раз следует подчеркнуть, что при редактировании всех рисунков активно используется возможность переноса, отображения, наклона и масштабирования выделенных фрагментов для создания композиционной взвешенности и законченности. При этом образующиеся в результате сдвига пустые пространства заполняются также подходящими полями рисунка.



Святая Святых

Следует остановиться и на возможности комбинированной компоновки рисунка на основе фрагментов различного происхождения. Например, в композиции Святого Места (внутреннего Храмового Двора) использована фотография выполненного автором макета Храмового здания, врезанная в рисунок. Качество исполнения не следует считать удачным (слишком разнятся цвета, хотя в здании преобладает золото),

но технология вполне иллюстрируется. Это позволяет задуматься о создании соответствующих декораций.



Святое Место

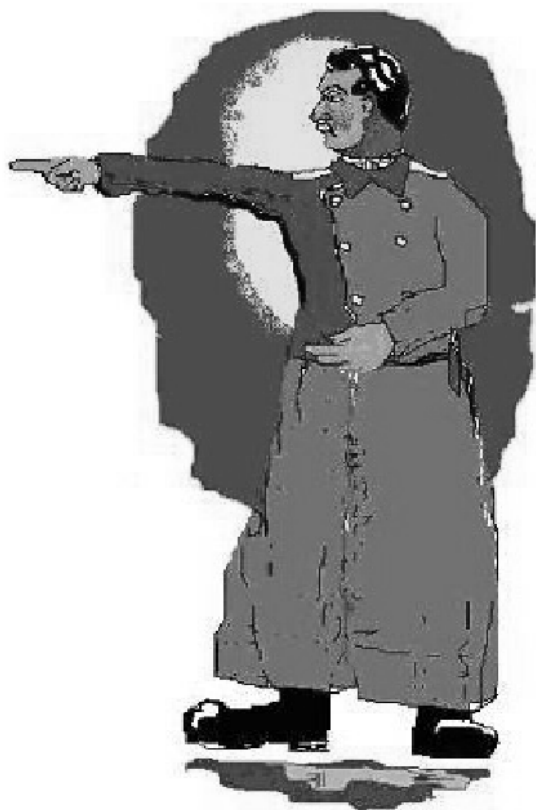
И, наконец, последний рисунок серии демонстрирует полное отсутствие необходимости компьютерного редактирования.



Сражение

Если речь идет о рисунке юмористическом, диктующем стиль исполнения, то возможностей Paintbrush иногда бывает вполне достаточно, как показано на следующем рисунке. Он выполнен

исключительно с помощью "мыши", по живому окну монитора компьютера.



Вождь

Юмористический рисунок в основе методологии литературных исследований

Посильно способствуя литературным исследованиям [32, 33, 34] накануне 111-летия М.А. Булгакова, автор представил несколько шаржированных рисунков, подвергшихся компьютерному редактированию. Некоторые "заказные" рисунки (для налогового инспектора: бесплатно!) представлены здесь.

Внимательное рассмотрение рисунка "Голгофа" иллюстрирует долгие поиски образов Иешуа и М. Булгакова: где-то видны элементы

несовпадения фактуры волос и др.

Рисунок "Арцибальд" иллюстрирует смелое предположение исследователя о том, что образ Арцибальда Арцибальдовича навеян М. Булгакову такими "рамочными" образами, как И. Сталин и пират Сильвер Р.Л. Стивенсона ("Остров Сокровищ").

Развитие литературоведческих поисков привело к раскрытию фактов эксплуатации образов М. Булгакова. Так, не менее смелое, чем выше, предположение было высказано (и подкреплено рисунком автора) относительно известного столпа социалистического *реализма*. Дескать, он в героях "эпохального" романа воплотил аналог всех центральных героев романа "Мастер и Маргарита".

Последний "заказной", но пока не опубликованный рисунок этой серии иллюстрирует тот факт, что Н.К. Крупская читала умирающему В.И. Ленину известный рассказ Д. Лондона, а И.В. Сталин, якобы, играл свою, известную нам роль.



Голгофа



Композиция



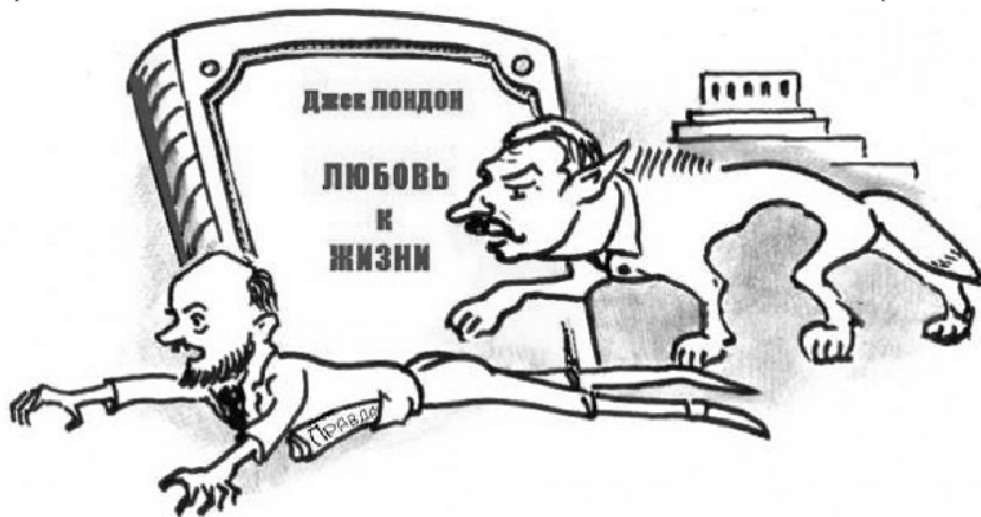
Маргарита



Арцибальд



Чего же ты хочешь?



Любовь к жизни

Из истории вычислительной техники

В данном разделе представлены рисунки, использованные (без указания авторства) в 2006 году в международном студенческом конкурсе на лучший сайт по истории вычислительной техники [36]. Конкурс проводился (и, по-видимому, будет проводиться ежегодно) в рамках структуры известной международной организации IEEE (*Institute of Electrical and Electronics Engineers*). Куратор команды МАТИ выбрал тему из истории счет – тех самых колесиков в раме, на которых мы в детстве пытались кататься с горки.

Сайт ссылка: www.schoty.ru - <http://www.schoty.ru> занял первое место (в мире!) и завоевал приз, по общему признанию, благодаря странице юмора, сделавшей решение неординарным.

Такое "признание" подвигло автора на самостоятельную публикацию рисунков с упомянутого сайта, а также некоторых рисунков, не вошедших в комплект.

Рисунки не только демонстрируют компьютерную технологию редактирования. Они, скорее демонстрируют те высокие требования, которые сегодня предъявляются к Интернет-изданиям по обеспечению

полноты и доходчивости образного восприятия, даже не без развлекательных элементов. Об этом можно судить по тому, что данный сайт предлагается как образец для следующего аналогичного конкурса.

Несомненно и то, что каждый рисунок обозначает и может быть развит в некоторый сюжет, требующий подключения, как минимум, средств анимации.

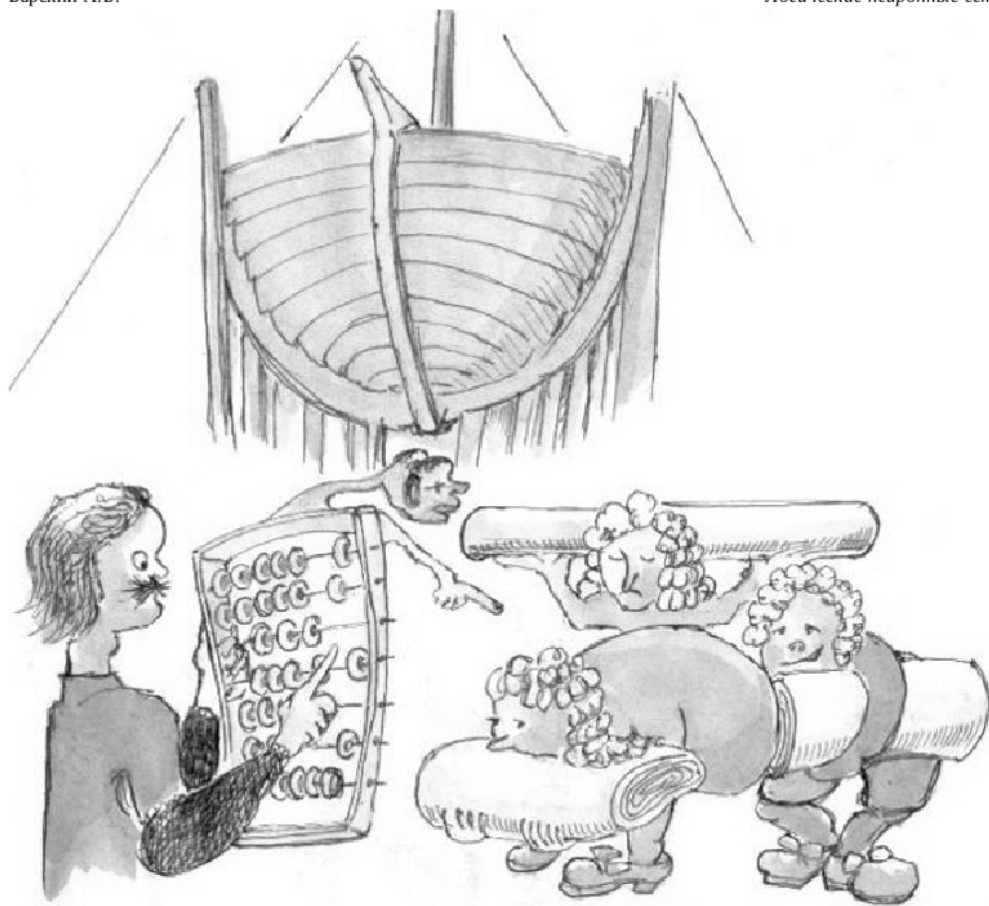
Рисунок "Герберт Аврилакский", по содержанию более подходящий к данному разделу, дополняет булгаковедческие исследования [35], объясняя появление Воланда в Москве.



Аудит



Боллид



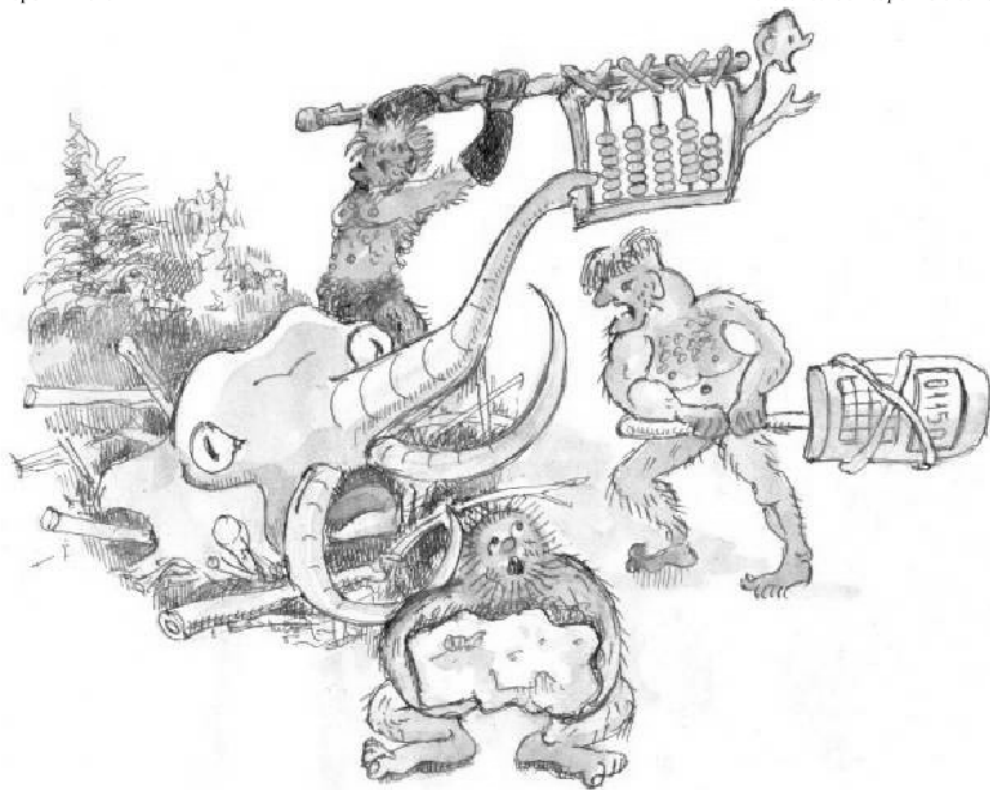
На верфи



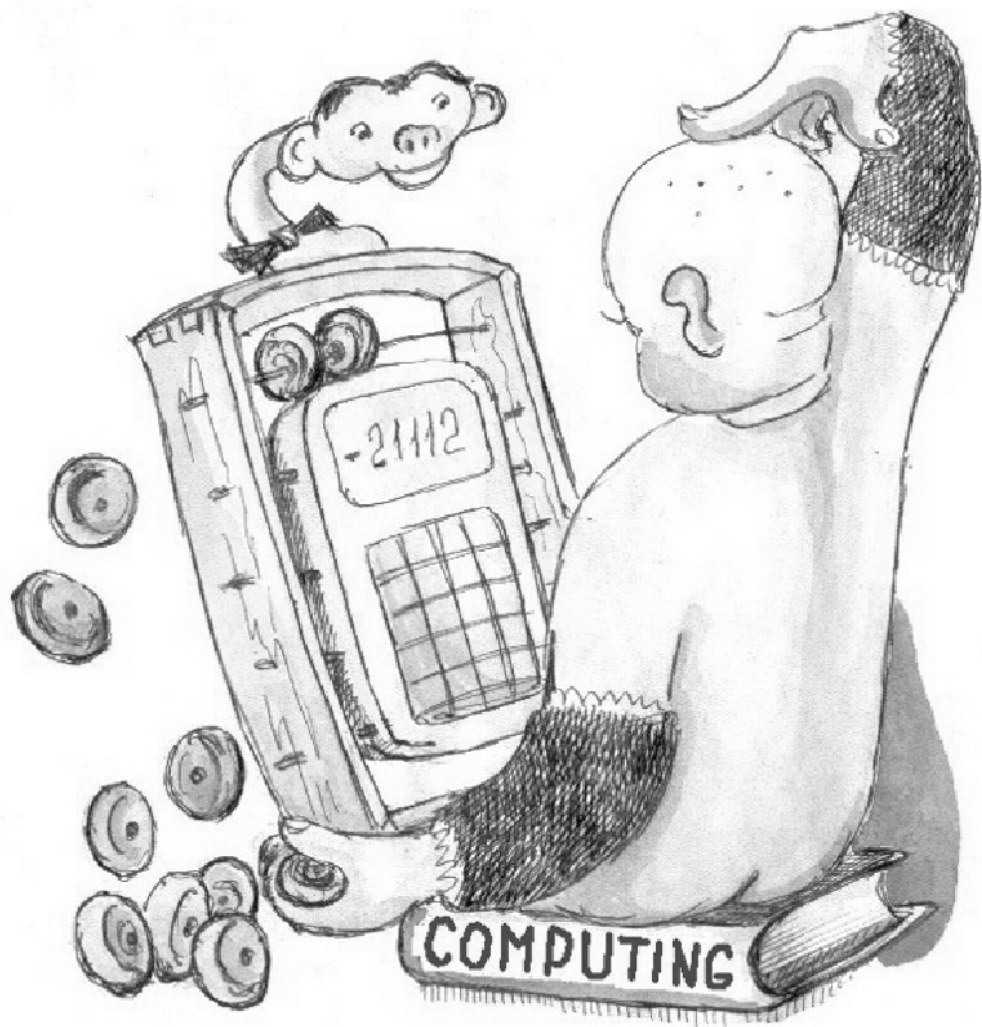
Дорожка



Каток



Охота



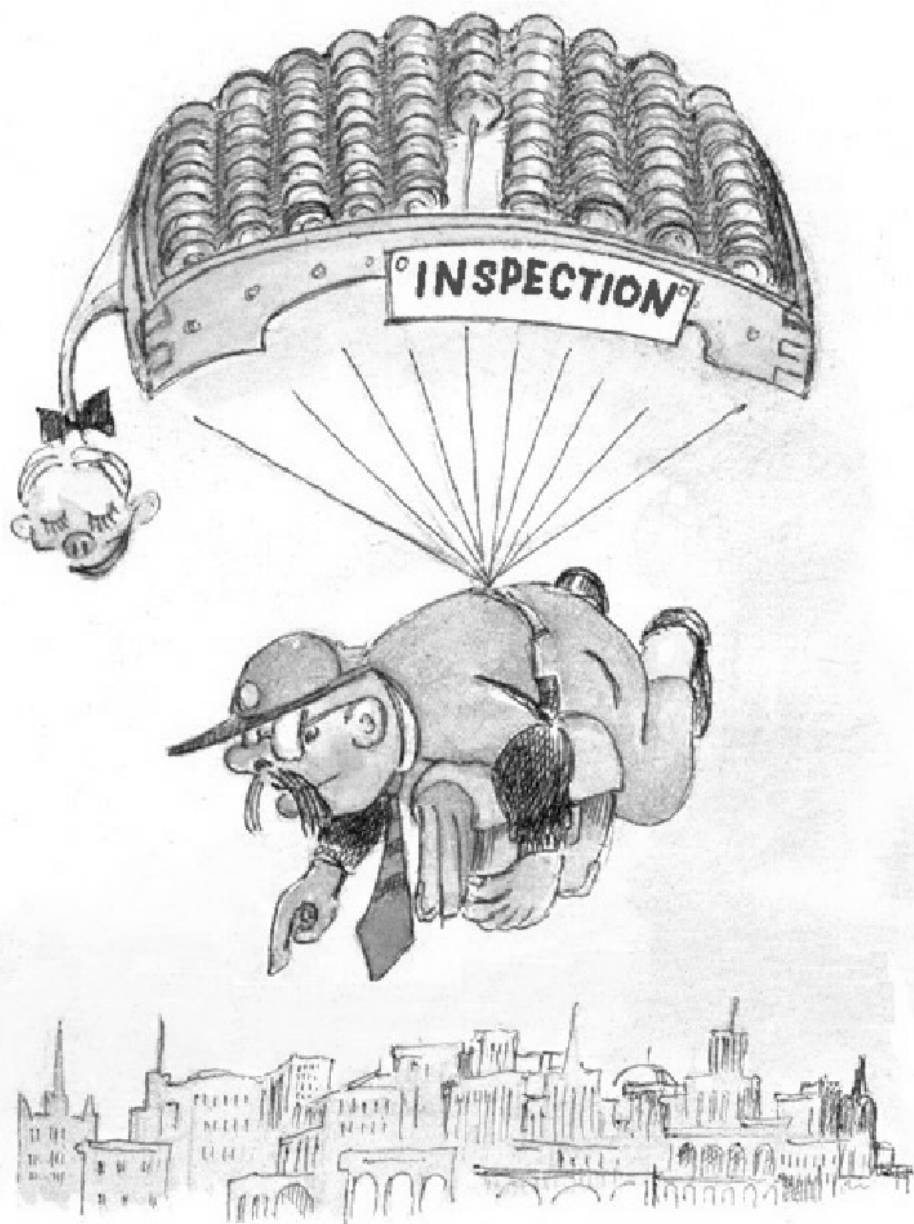
Калькулятор



Шторм



Гуманитарная миссия



Десант



Музыканты



Врата рая



Герберт Аврилакский (папа Сильвестр II)

Нейронные сети – аппарат искусственного интеллекта

В данном разделе представлены рисунки – продукт непосредственной

научной и преподавательской деятельности автора [11, 37, 38]. Несмотря на юмористический характер представленных рисунков, они исключительно серьезны. Как говорилось ранее, их назначение – увести читателя с пути мистического мышления, объяснить с помощью шутки принцип работы того, к чему мы боимся прикоснуться, сделать это доступным воплощению.

Многие рисунки, являясь иллюстративным материалом, содержат в явном виде схемы и объяснение. Однако все рисунки будят фантазию созидания, добиваются впечатления о простоте воплощения тех принципов, на основе которых живое существо считается живым. И не следует задумываться глубже! Это опасно...

Выше говорилось, что мультимедийные курсы рассматриваемой дисциплины существуют. Однако они не устраивают автора недостаточным уровнем анимации и трехмерной графики. Необходимо даже продумать сценарий наглядного освещения материала. Широкое поле деятельности здесь только открывается, и пусть приведенные рисунки это подтвердят. Таким образом, их следует рассматривать в качестве развиваемых сюжетов.

Рисунок "Бабушка" сопровождает рассказ о том, как с помощью нейронной сети – аналогичной мозгу, можно создать памятку пожилому человеку (систему принятия решений), определяющую его действия во все времена дня и года.

Ряд рисунков этой серии, так или иначе, служат проекту создания ПАРКА ФАНТАСМАГОРИЙ на основе уже научно и технологически доступного нейросетевого "живого" моделирования.

Представьте себе парк, подобный Дисней-ленду, в который вы заходите – самостоятельно купив билет или под руководством гида. И вдруг!

... Выползают из-за трепещущих кустов и камней монстры - крокодилы, вздымаются головы ящеров и тянутся к вам. Лохматый неандерталец ладит каменное орудие, привязанное жилой к сучковатой палке. Подымается на свои куры лапы избушка. Гигантской головой оживает скала, покрытая нежными незабудками. Деревья и кусты трясут ветками - радуясь или возмущаясь. Фонтаны воды преследуют вас... Ваше замешательство, паническое движение вызывают ответную реакцию:

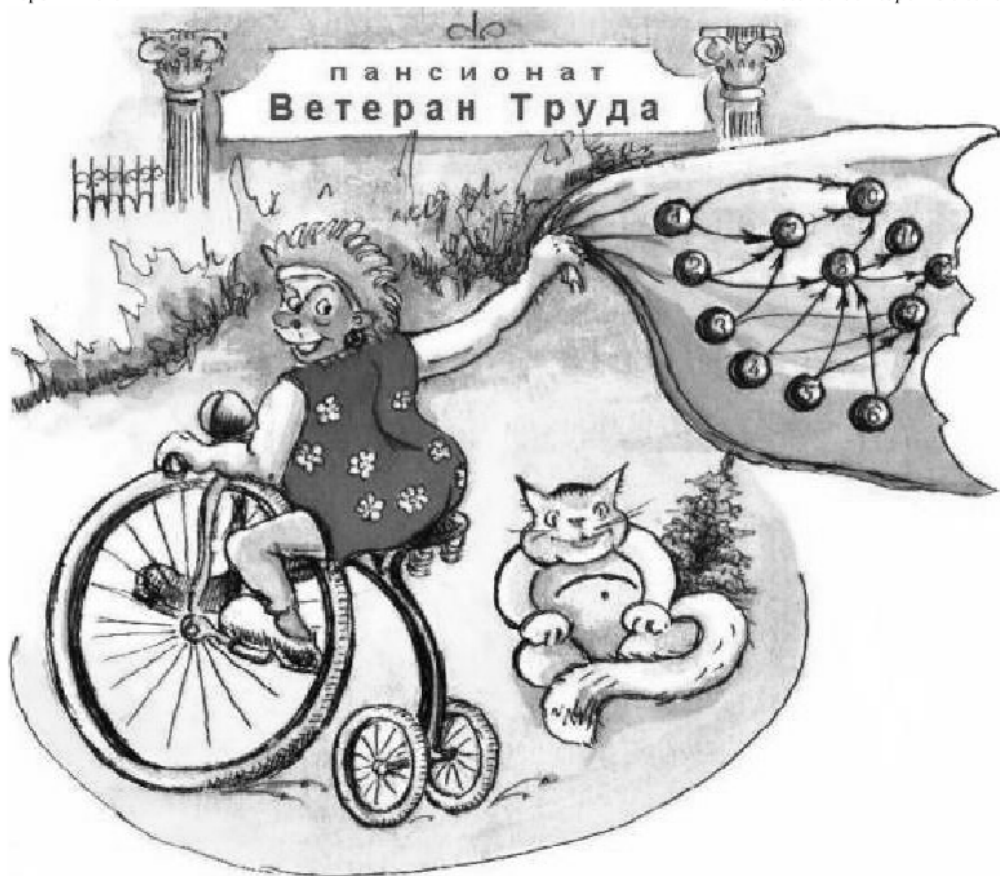
"население" парка смеется, сердится, радуется, угрожает. Явно проявляются симпатии и антипатии...

До последующих усовершенствований, - первоначально, должна быть предусмотрена реакция каждого объекта только на преобладающий цвет в элементарном сегменте обзора. Например, красный цвет должен приводить к запуску комплекса программ обращения в ярость и к повторному обзору строки, чтобы усилить эту ярость при последующем раздражении.

(- Женщина в красной кофте! Отойдите от крокодила немедленно!..)

Желтый цвет может вызывать кокетство и эротическое возбуждение. Зеленый - умиротворение. Коричневый - ритмичный шаг и салютование головой. Голубой - благочинное умиление с возведенными очами и т.д.

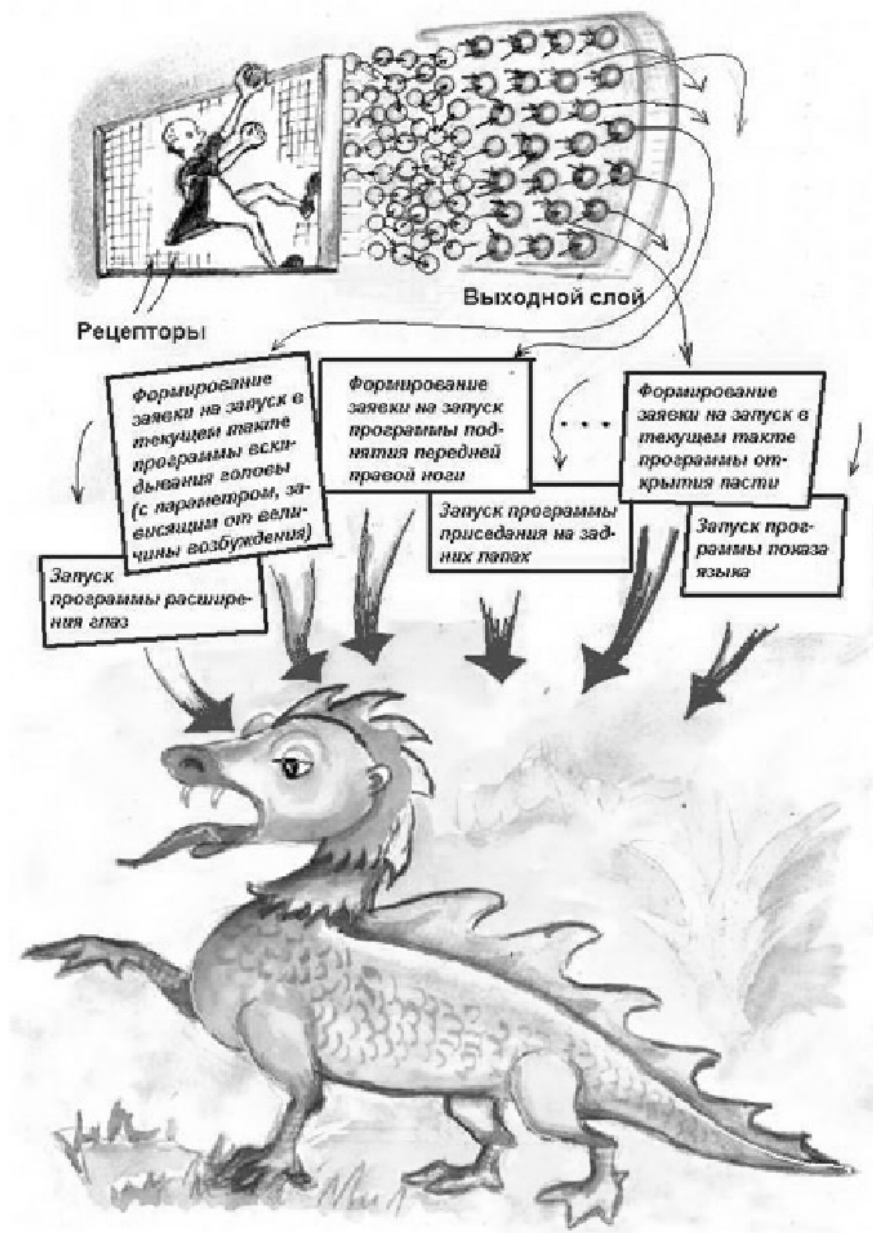
Согласитесь, что это бизнес, достойный перспективных разработок.



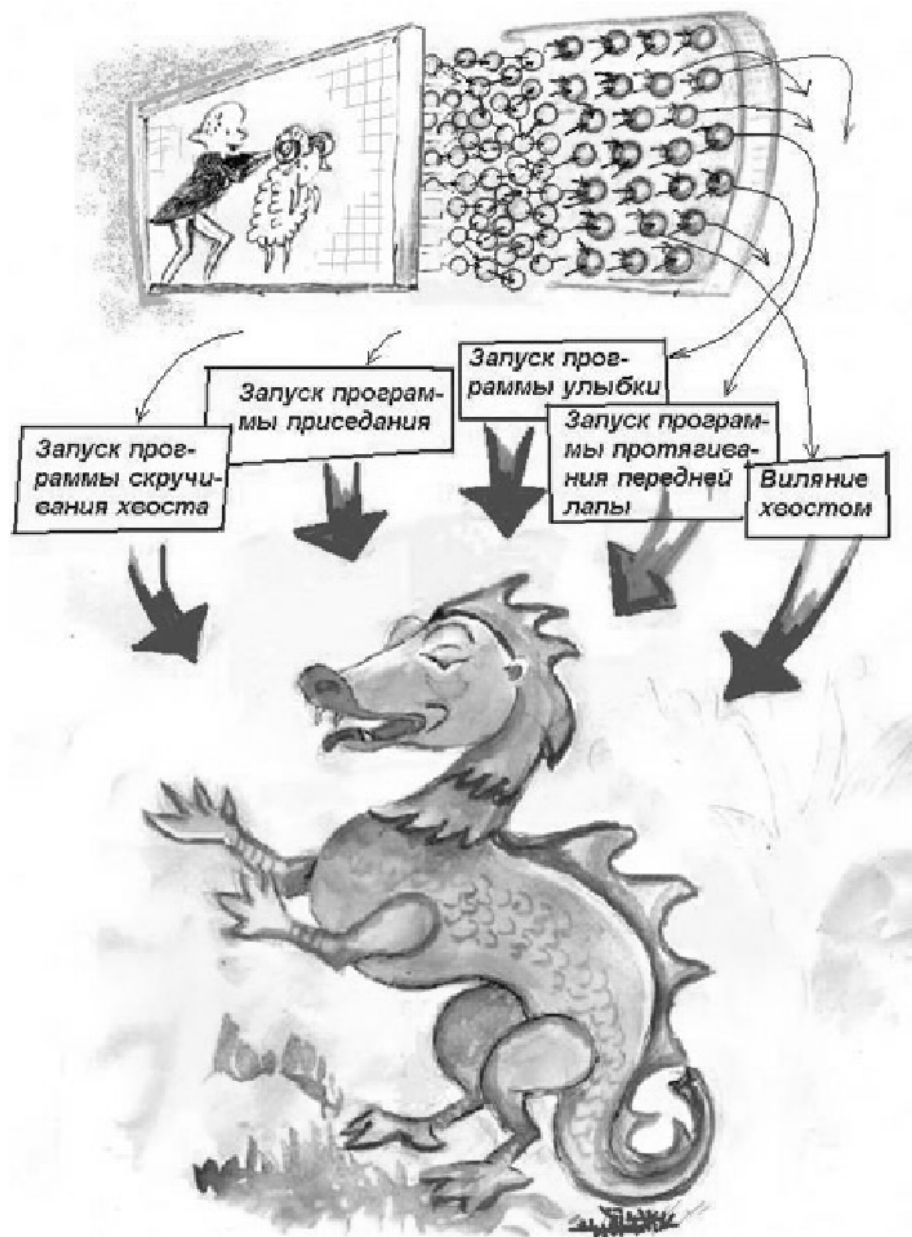
Бабушка



Парк фантазмагорий



Реакция на угрозу



Реакция на ласку



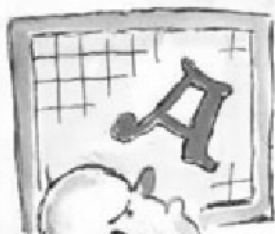
Способ разглядывания 1



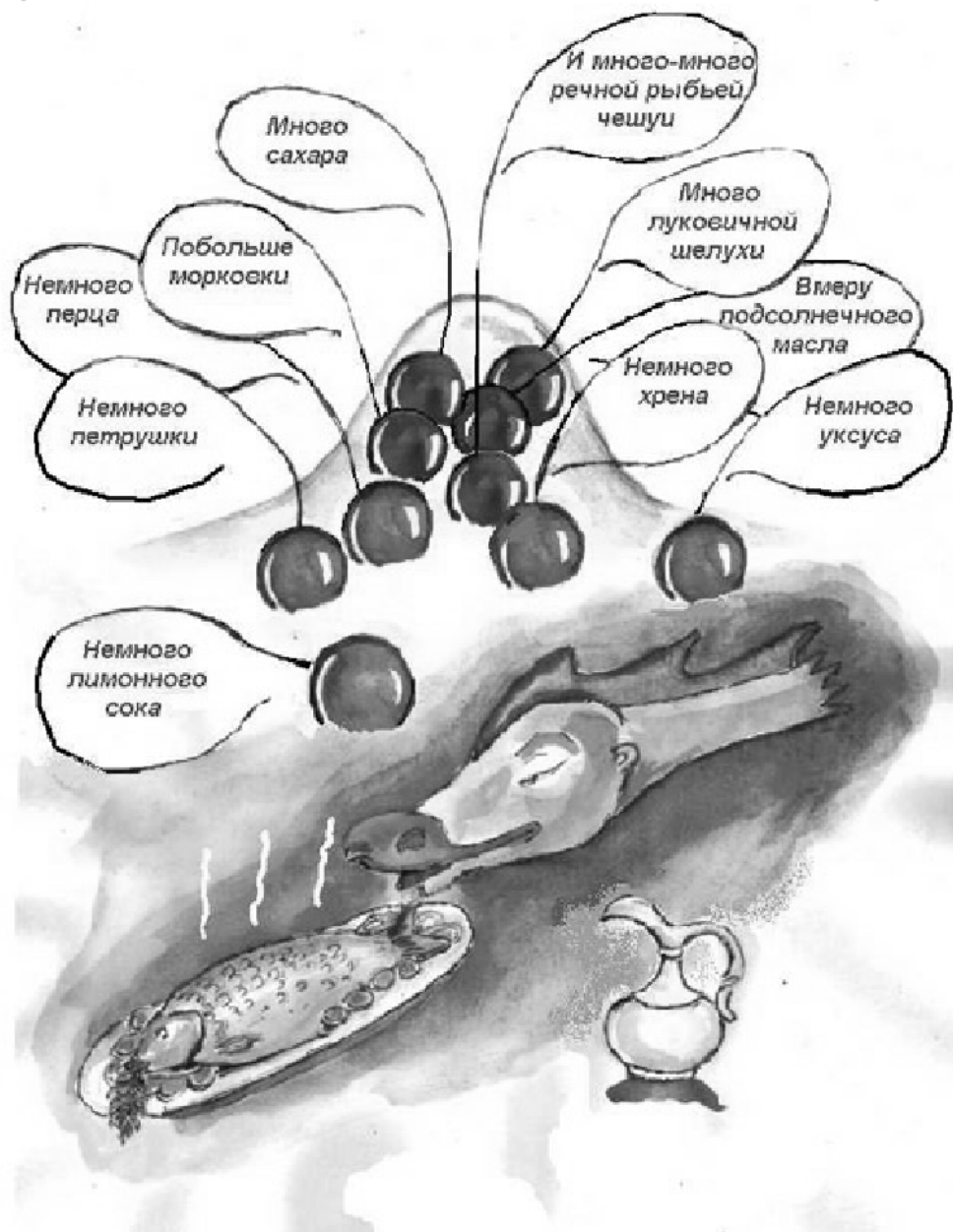
Способ разглядывания 2



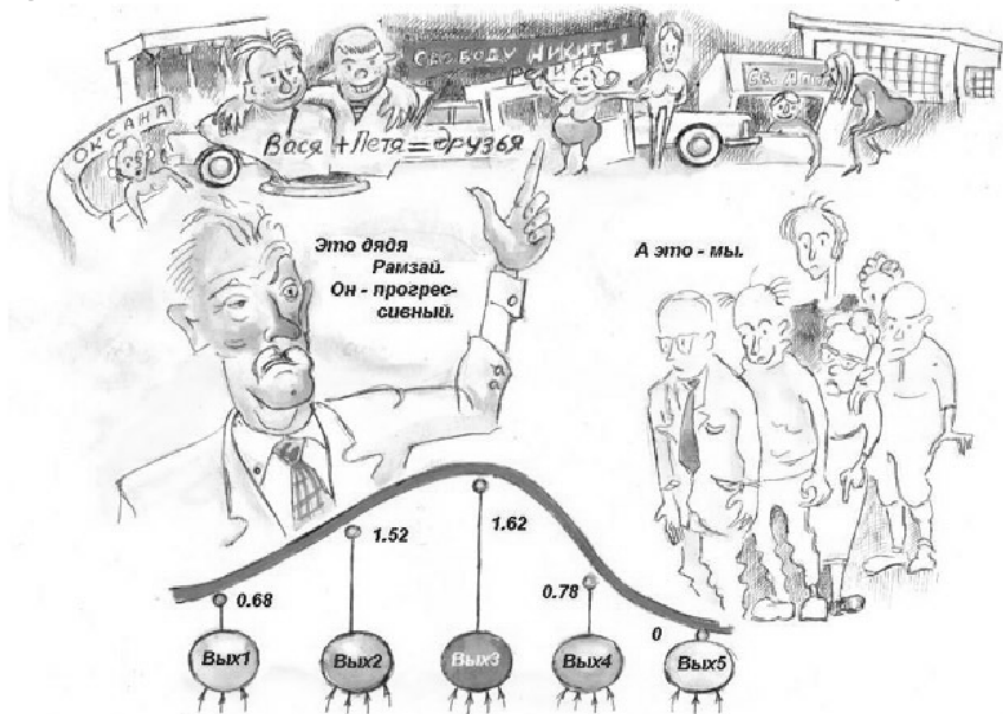
Способ разглядывания 3



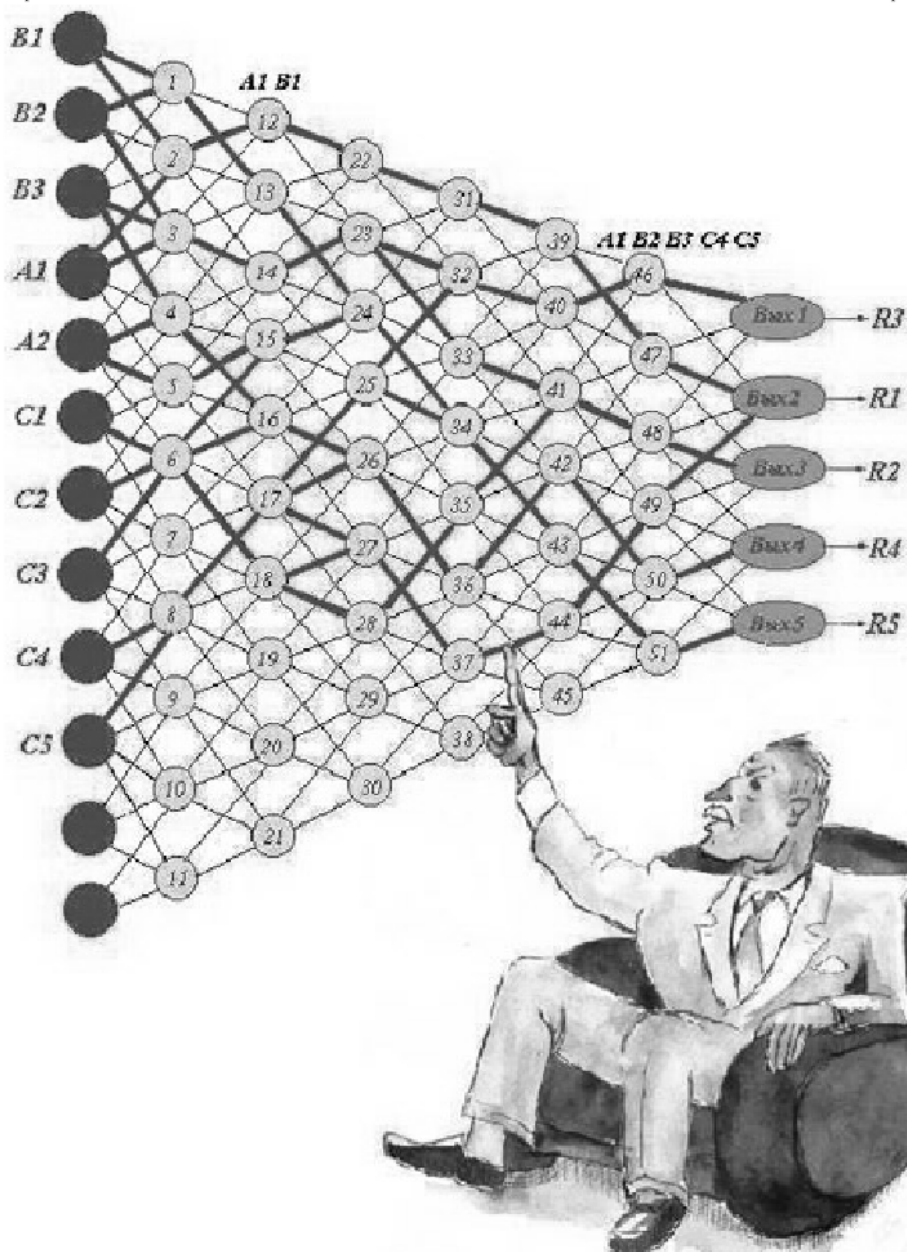
Узнавание



Нейродегустатор



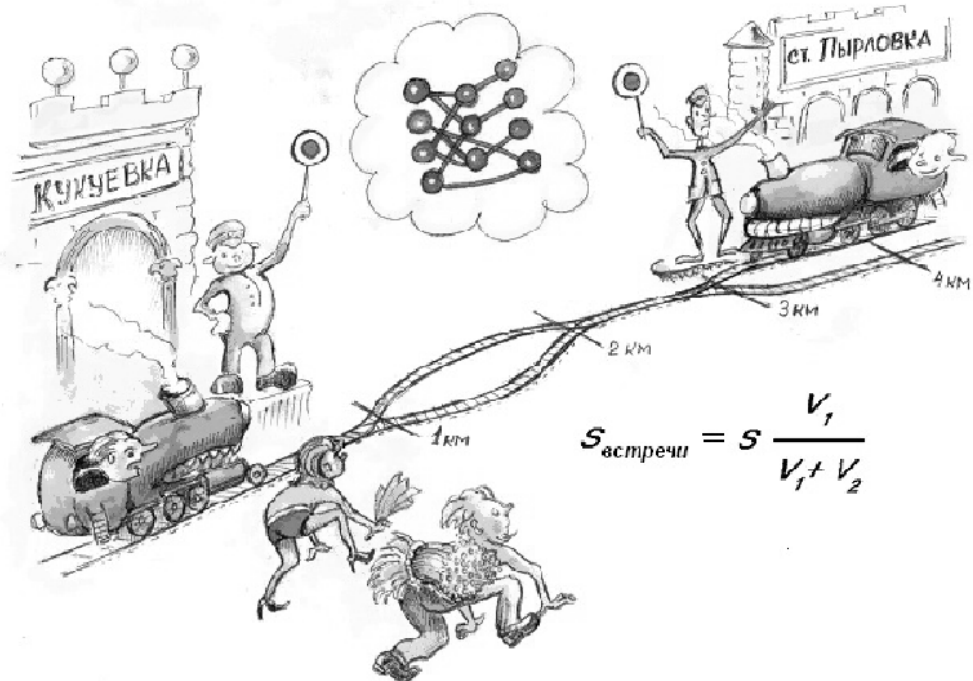
Предмет исследований



Запомнила?



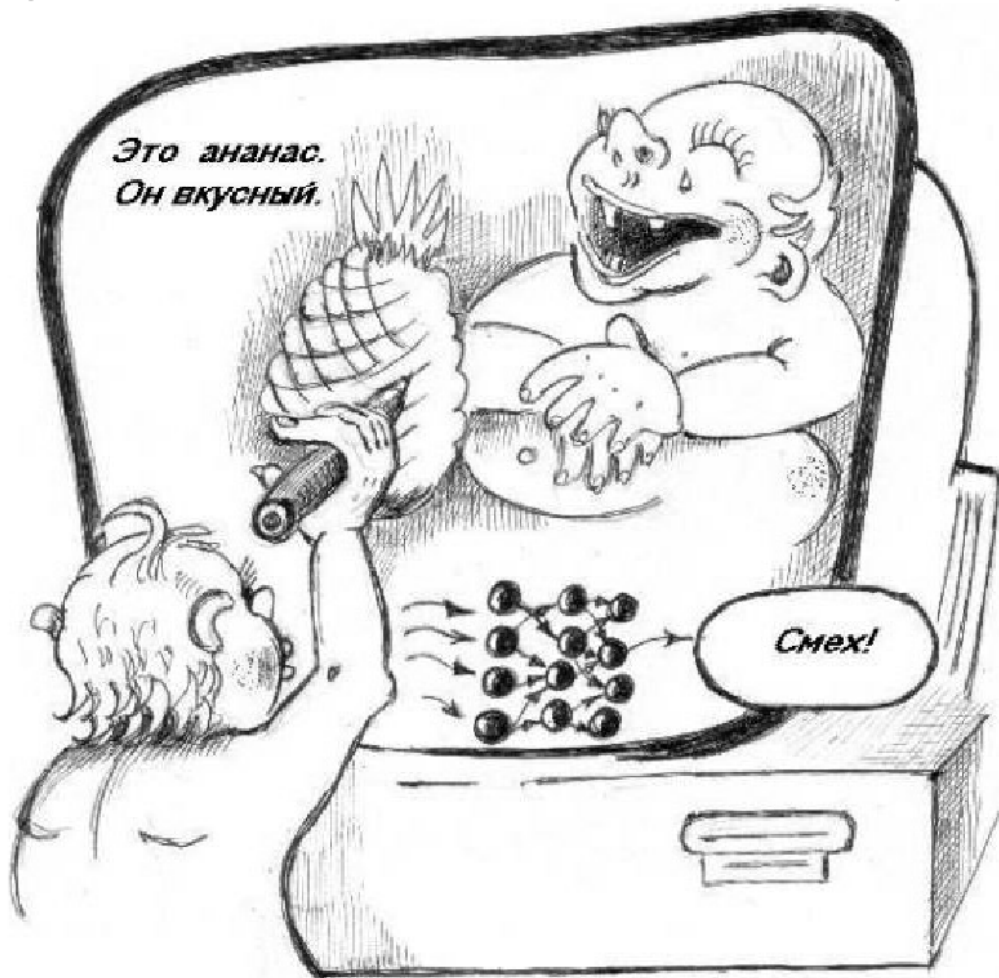
Намек на обучение



Железнодорожная рулетка



Служба безопасности

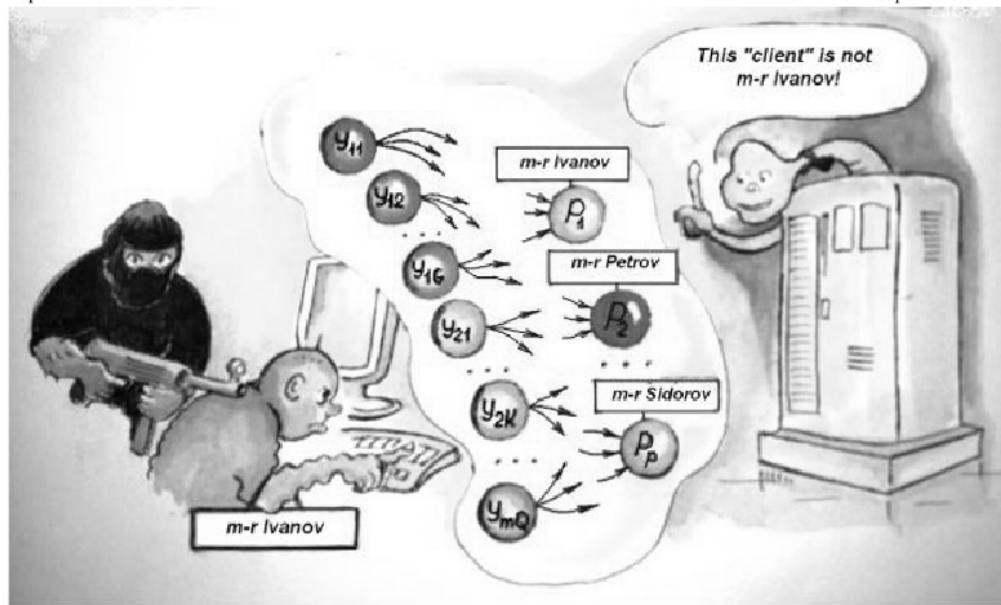


Компьютерный человечек КОМПИ

Ода платному образованию



Профанация



Защита информации

Интеллектуальное отображение

Затронув вопрос о перспективных разработках в предыдущем разделе, следует остановиться на проблеме *интеллектуального отображения*, основанного на построении реагирующих объектов [34, 35].

Управление и контроль состояния современных систем, таких как транспортные, энергетические, социальные, столь сложны, трудоемки и мало обозримы, что требуют начальной грубой, приблизительной, но оперативной оценки. Целью такой оценки является вывод о правильности в целом работы системы или об ее отказе, начальная информация по диагностике, предупреждение о возможной опасной ситуации и т.д. Взгляд оператора на поведение реагирующего объекта – на монитор компьютера или на объект натурального моделирования – может быть достаточным для дальнейшего спокойствия или принятия срочных действий.

Реагирование объекта осуществляется на основе логической нейронной сети, иллюстрации к которой приведены в разделе 4. Простейшее решение при этом может заключаться в анимации, управляемой

значением возбуждения рецепторов. Для такой анимации создается база данных – система ключевых кадров, отображающих основные результирующие реакции объекта. Ключевой кадр, который должен быть показан, выбирается динамически с помощью нейронной сети в реальном времени по состоянию контролируемой системы. Система анимации формирует промежуточные кадры для обеспечения плавного восприятия при переходе объекта от текущего состояния в требуемое.

На рисунке приводится возможный набор ключевых кадров для экспериментальной демонстрационной модели (разработка которой пока не состоялась из-за отсутствия средств). В соответствии с изложенной выше технологией редактирования рисунка, все кадры получены из одного, базового, с помощью *Paintbrush*.



Ключевые кадры

Заключение

Вопросы, затрагиваемые в статье, носят постановочный характер. Приведенные рисунки, совместно с их функциональным назначением – призыв к сотрудничеству в развитии направления исключительно полезного применения изобразительных средств во всех областях деятельности. Это призыв к развитию сюжетного ряда, к совершенствованию средств выражения, их обращению в область анимации, мультимедиа, трехмерной графики, цифрового кино, *интеллектуального отображения* сложных управляющих систем и создания перспективных развлекательных средств.

Список литературы

1. Hopfield J., Tank D, *Neural computation of decision in optimization problems*, Biol. Cabernet., 1985, V.52
2. Kohonen T, *Self-Organizing Maps*, Springer-Verlag, 2nd ed., 1997
3. Deboeck G., Kohonen T, *Visual Exploration in Finance with Self-Organizing Maps*, Springer-Verlag, 1998
4. Rosenblatt F, *Perceptual generalization over transformation groups*, Self-Organizing Systems, Pros. Interdisciplinary Conf., 5-6 May, 1959, PP. 63-100, Русский перевод: Ф. Розенблат. Обобщение восприятий по группам преобразований. Кибернетический сборник, 1962
5. Амосов Н.М., Байдык Т.Н., Гольцов А.Д. и др, *Нейрокомпьютеры и интеллектуальные роботы*, Киев: Наукова думка, 1991
6. Архангельский А.Я, *Программирование в Delphi 5*, М.: БИНОМ, 2000
7. Барский А.Б, *Нейронные сети: распознавание, обучение, принятие решений*, М.: Финансы и статистика, 2004
8. Барский А.Б, *Логические нейронные сети для систем управления и принятия решений*, Нейрокомпьютеры: разработка и применение, № 6, 2005
9. Барский А.Б, *Параллельные процессы в вычислительных системах: планирование и организация*, М.: Радио и связь, 1990
10. Барский А.Б., Шилов В.В, *SPMD-архитектура и параллельный логический вывод*, Приложение к журналу "Информационные технологии", № 2, 2001
11. Барский А.Б, *Логические нейронные сети: методика построения и некоторые применения*, Приложение к журналу "Информационные технологии", № 8, 2006
12. Барский А.Б., Милютин Л.Б., Тимофеев А.Е, *Основы построения реагирующих объектов для систем интеллектуального отображения*, Компьютеры в учебном процессе, № 5, 2005
13. Бондарев А.Б, *Прогнозирование биржевых сделок предприятий*, Практическое пособие. М.: Экономика и финансы, 1999
14. Галушкин А.И, *Теория нейронных сетей. Серия "Нейрокомпьютеры и их применение"*, книга 1, М.: ИПРЖР, 2000
15. Галушкин А.И, *Нейрокомпьютеры. Серия "Нейрокомпьютеры и их применение"*, книга 3, М.: ИПРЖР, 2000
16. Головкин Б.А., *Параллельные вычислительные системы*, М.: Наука, 1980
17. Головку В.А, *Нейронные сети: обучение, организация и применение*, Серия "Нейрокомпьютеры и их применение", книга 4. М.: ИПРЖР, 2000
18. Франклин Д., Патон Б, *Flash4. Анимация в Интернете*, Пер. с англ. СПб.: Символ-Плюс, 2000
19. Игнатущенко В.В, *Организация структур управляющих многопроцессорных вычислительных систем*, М.: Энергоатомиздат, 1984
20. Комашинский В.И., Смирнов Д.А, *Введение в нейроинформационные технологии*, СПб.: Тема, 1999
21. Круглов В.В., Борисов В.В, *Искусственные нейронные сети. Теория и практика*, М.: Горячая линия – ТЕЛЕКОМ, 2001
22. Кемени Дж., Снелл Дж., Томпсон Дж, *Введение в конечную математику*, М.: Изд-во иностранной литературы, 1963

23. Новиков П.С, *Элементы математической логики*, М.: Государственное издательство физико-математической литературы, 1959
24. Поспелов Д.А, *Ситуационное управление. Теория и практика*, М.: Наука, 1986
25. Поспелов Д.А, *Десять "горячих точек" в исследованиях по искусственному интеллекту*, Интеллектуальные системы (МГУ). Т. 1, вып. 1-4. 1996
26. Поспелов Д.А, *Моделирование рассуждений. Опыт анализа мыслительных актов*, М.: Радио и связь. 1989
27. Форсайт Д., Понс Ж, *Компьютерное зрение. Современный подход*, М.: Вильямс, 2004
28. Фролов Ю.В, *Интеллектуальные системы и управленческие решения*, М.: МГПУ, 2000
29. Фролов Ю.В., Пастухов Е.С, *Мониторинг изменений в банковском сообществе России с применением самоорганизующихся карт Кохонена*, Банковские технологии, № 11, 2000
30. Шеннон Р, *Имитационное моделирование систем. Искусство и наука*, М.: Мир, 1978
31. Ярушкина Н.Г, *Основы теории нечётких и гибридных систем*, М.: Финансы и статистика, 2004
32. В.В. Шилов, *Михаил Булгаков и романы приключений: три модели адаптации*, Экономика XXI века, 2002, №11
33. В.В. Шилов, *Ах, слова хороши! 111 лет со дня рождения М.А. Булгакова*, Маркетинг успеха, 2002, №6
34. В.В. Шилов, *Всеволод Кочетов читает Михаила Булгакова*, Экономика XXI века, 2006, №1
35. В.В. Шилов, *Михаил Булгаков: Почему Герберт Аврилакский?*, Маркетинг успеха, 2002, №9
36. *Русские счёты*, URL: <http://www.schoty.ru>.
37. А.Б. Барский, *Логические нейронные сети: методика построения и некоторые применения*, Приложение к журналу "Информационные технологии", 2006, №8
38. А.Б. Барский, *Логические нейронные сети и "живое" моделирование*, Компьютеры в учебном процессе, 2005, №11
39. А.Б. Барский, Л.Б. Милютин, А.Е. Тимофеев, *Реагирующие объекты для систем интеллектуального отображения*, Информационные технологии, 2007, №2

Содержание

Титульная страница	2
Выходные данные	3
Лекция 1. Математическая логика событий	4
Лекция 2. Основы нейросетевых технологий	34
Лекция 3. Построение современной нейросетевой технологии	72
Лекция 4. Трассировка нейронной сети	110
Лекция 5. Стратегии обучения и самообучения	145
Лекция 6. Нейронные сети с обратными связями	160
Лекция 7. Нейросетевые самообучающиеся системы управления	171
Лекция 8. Логическое программирование нейронной сети	186
Лекция 9. Структурное обоснование логической нейронной сети	203
Лекция 10. Корректировка параметров, дистрибутивные преобразования, однослойные и совершенные логические нейронные сети	230
Лекция 11. Трассировка логической структуры нейросети	260
Лекция 12. Методика построения системы принятия решений на основе логической нейронной сети	285
Лекция 13. Нейросетевые технологии в экономике и бизнесе	296
Лекция 14. Нейросетевые модели пошаговой оптимизации, маршрутизации и тактических игр	316
Лекция 15. Основы "живого" моделирования	336
Лекция 16. Перспективные нейросетевые технологии	393

Лекция 17. Компьютерная технология рисунка и ее применение в научных исследованиях и образовании	434
Список литературы	490