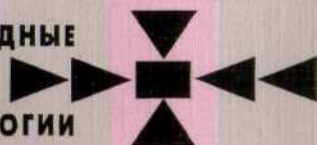


ПРИКЛАДНЫЕ  
ИНФОРМАЦИОННЫЕ  
ТЕХНОЛОГИИ



*А. Б. Барский*

**НЕЙРОННЫЕ СЕТИ:  
распознавание,  
управление,  
принятие решений**



Консультационный центр MATLAB компании SoftLine

Назад Вперед Остановить Обновить Двойной Поиск Избранное Журнал Понты Печать

Адрес: <http://www.matlab.ru>

консультационный ЦЕНТР MATLAB компании Softline


Семинары Обучение Лицензирование Разработка Подписка Форум Регистрация

**MATLAB & Toolboxes** M-разработки

Учебные курсы

[Matlab](#)  
[Toolboxes](#)  
[Simulink](#)  
[Blocksets](#)  
[Femlab](#)  
[3rd-Party Products](#)  
[Полное](#)

# НЕЙРО-СЕТЕВЫЕ ТЕХНОЛОГИИ



*Кризис сметает слабых. Сильные же не только выживают, но и увеличивают доходность бизнеса, быстрее других осваивая новые средства и возможности. Дальновидные и предприимчивые сегодня получили реальный шанс потеснить вчерашних лидеров, используя новейшие технологии в бизнесе.*

**Для тех, кто хочет преуспеть в бизнесе и стать лидером на рынке!**

**Учебный курс**  
**"Применение нейронно-сетевых алгоритмов на базе системы MATLAB в экономическом и финансовом прогнозировании".**

В программе курса - теория нейронных сетей, прогнозирование финансовых потоков, оценка индексов курсов акций, оценка кредитного риска, нейронные сети для блестящей игры на торговом рынке.

Информация о ближайших курсах - <http://www.matlab.ru/lection/>

**Консультационный Центр MATLAB**  
[www.matlab.ru](http://www.matlab.ru)  
 (095)232-00-23  
[info@matlab.ru](mailto:info@matlab.ru)

**softline**<sup>®</sup>  
scientific software

Интернет

*А.Б.Барский*



# НЕЙРОННЫЕ СЕТИ: распознавание, управление, принятие решений



Москва  
"Финансы и статистика"  
2004

УДК 004.89.032.26

ББК 32.813

Б26

Серия  
«Прикладные информационные технологии»  
Основана в 1997 г.

Главный редактор серии  
доктор технических наук, профессор  
*С. В. Черемных*

*РЕЦЕНЗЕНТ:*

*Н. Г. Ярушкина*,  
доктор технических наук, профессор, заведующая кафедрой  
«Информационные системы»  
Ульяновского государственного технического университета

Барский А. Б.

Б26      Нейронные сети: распознавание, управление, принятие решений. — М.: Финансы и статистика, 2004. — 176 с: ил. — (Прикладные информационные технологии).

ISBN 5-279-02757-X

Рассматривается применение нейросетевых технологий при построении информационных и управляющих систем в науке, экономике, финансах и искусстве. Исследуются вопросы разработки нейросетей «под задачу», представления исходной и обработки выходной информации. Предлагаются простые методы обучения в статическом и динамическом режимах. Обсуждаются особенности систем принятия решений, самообучающихся управляющих систем, систем логического вывода, банковского мониторинга, безопасности, защиты информации, политического и социального прогноза, бизнеса развлечений и туризма.

Для студентов технических и экономических вузов, аспирантов, инженеров и исследователей в области современных информационных технологий.

Б —  $\frac{1402070000 - 120}{010(01) - 2004}$  341 — 2004

УДК 064.89.032.26  
ББК 32.813

ISBN 5-279-02757-X

© Барский А. Б., 2004

# К ЧИТАТЕЛЮ

Книга А.Б. Барского посвящена важнейшему направлению искусственного интеллекта — нейронным сетям. Построение нейронной сети — это попытка создания искусственной нервной системы, в состав которой входит и человеческий мозг. Сами нейроны — это десятки миллиардов элементарных объектов, которые за нас с вами «думают», и исследователю в этой области необходимо понять, каким образом происходит этот процесс «думания», как человеческий мозг обучается обрабатывать информацию, и делать соответствующие выводы. Смысл существования нейронной сети, как подчеркивает автор, заключается в обучении и адаптации.

Автор акцентирует наше внимание на принципах ассоциативного мышления, так свойственного человеку в его деятельности. Именно ассоциативное мышление лежит в основе распознавания, движения, управления, принятия решений. Это в значительно большей степени приближает нас к проблемам искусственного интеллекта, в то же время логически подводя к моделям управления в экономике, бизнесе, финансах, производстве и т.д.

Назовем лишь некоторые из широко известных областей, где эффективно работают так называемые нейросетевые технологии, которые не имеют «солидного», как говорят математики, обоснования. Это страховая деятельность банков; прогнозирование банкротств, денежных потоков, налоговых поступлений; оценка кредитного риска, эффективности биржевой деятельности; предсказание результатов займов и многое другое.

Ключевым инструментом здесь являются искусственные нейронные сети, реализуемые на основе специализированных пакетов программ. Разве это не парадокс — всеобъемлющей теории нет, а пакет программ уже есть! Например, пакет прикладных программ Excel Neural Package, использующий в качестве функций активизации гиперболический тангенс, а в качестве алгоритма обучения — алгоритм Rprop.

Кроме того, нейронные сети применяются для грамотного сравнения финансово-экономических программных систем. Обычно это делается на базе экспертных оценок, но такой подход необъективен. Нужна хотя бы и неполная, но формализация. Она возможна, например, на основе Самоорганизующихся Карт Ко-

хонена (СОК) с использованием программного пакета Viscovery SOMine. Возможна потому, что здесь в полной мере могут быть задействованы такие уникальные свойства искусственных нейронных сетей, как способность к обучению, выявлению скрытых закономерностей, накоплению опыта.

Оригинальный подход автора обеспечивает доходчивость изложения проблем искусственного интеллекта и перспективу его практического воплощения. Этот подход основан на принципах построения электронных схем вычислительных устройств, отображающих такие основы математической логики, как булевы функции. Конечно, при этом имелось в виду по большей части программное обеспечение ЭВМ, нежели ее аппаратура.

Перспектива построения систем искусственного интеллекта во многом основана именно на математической логике. Обобщение понятия булевых функций с большей степенью адекватности приводит к воссозданию принципов мозга — к построению схем ассоциативного мышления. Для этого необходимо перейти от оперирования над булевыми переменными к операциям над действительными, отражающими оценку или достоверность исходных данных. А это тотчас же потребует замены "конъюнкторов" и "дизъюнкторов" на универсальный логический элемент — нейрон, выполняющий простую пороговую передаточную функцию. •

Автор является специалистом в области организации и планирования параллельных вычислительных процессов. Это и обусловило его подход к изучению нейронных сетей, а также к их воспроизведению на нейрокомпьютере как параллельных вычислительных устройств и систем, имитирующих ассоциативное мышление. Это же обусловило схемотехнический подход к построению и обучению нейронных сетей.

Книга написана с юмором, живым, образным языком. Она будет интересна широкому кругу читателей и как учебное пособие, и как научно-популярное издание, и как практическое руководство.

*С.В. Черемных,*  
доктор технических наук, профессор,  
заведующий кафедрой математики  
и информационных технологий  
' Московской академии предпринимательства  
при Правительстве г. Москвы

## ПРЕДИСЛОВИЕ

По нейронным сетям накоплен огромный материал, способный привести в растерянность неискушенного читателя, пытающегося понять, что такое нейросети. С этой проблемой столкнулся автор данной книги при чтении курса лекций по нейроинформатике студентам технического вуза, обладающим традиционным объемом математических знаний, и несформированным формально-логическим мышлением. Помогла схемотехническая направленность их знаний в области конструирования электронных схем и программирования. Устранение излишней «математизированности» проблемы и смещение ее на уровень прагматический, оперирующий лишь с простыми логическими элементами, — одна из основных целей настоящей книги.

Как же добиться доступности изложения? Один путь уже указан: использование простых принципов схемотехники. Другой путь — в применении методов распараллеливания обработки информации, присущих такой универсальной нейронной сети, какой является мозг. Однако параллельные вычислительные процессы «обладают» теорией, несложными методами расчета и организации, известными специалистам. Представляется логичным приложить простейших методов и концепции распараллеливания к такой же параллельной системе, как нейросеть.

Подобный подход оказался плодотворным и, в частности, привел к простым и уже знакомым аудитории алгоритмам обучения нейросети, позволяющим полностью устранить взаимное влияние эталонов, по которым производится обучение.

И наконец, основной способ достижения доступности изложения состоит в строгом следовании идеям искусственного интеллекта, воспроизводящим работу мозга. Именно в этом случае нейросети обеспечивают простое и естественное решение тех проблем, о которых заявлено в названии книги.

Мозг умеет все, и, что очень важно, логика его работы весьма проста. Она использует связи «если - то», «посылка - следствие». Более того, это основные отношения, складывающиеся на этапе обучения в незримые таблицы. В свою очередь, основной вопрос: «На что более всего похож предъявляемый образ и что из этого следует?», в то же время является основной функцией обучения

нейросети, воспроизводящей работу мозга и ассоциативное мышление. И мы уже видим, как с помощью такого принципа решаются представленные здесь задачи.

Защищая принципы искусственного интеллекта и нейросетевые технологии, спросим читателя, много ли он считает, т.е. оперирует с числами, в своей обыденной жизни? Как находит угол поворота-рулевого колеса, чтобы удержать автомобиль на дороге? Как выбирает значения массы параметров, чтобы попасть мячом в баскетбольную корзину? Как вообще он передвигается по дороге, не спотыкаясь о бугорки и обходя лужицы?

Интерполируя по тем самым незримым таблицам, реализованным и развиваемым в нейронной сети нашего мозга, мы можем безбедно прожить жизнь, не ставя перед собой тяжелых творческих задач. И только взаимодействие таких таблиц, содержащихся в них отношений (в совокупности с образной памятью), позволяет строить логические цепочки, называемые умозаключениями.

Книга содержит девять разделов. В разд. 1 обсуждаются проблемы построения и применения нейронных сетей. В разд. 2 строится обученная нейросеть на основе «схемотехнического» подхода. В разд. 3 приводятся пример и формальный алгоритм обучения нейронной сети методом трассировки. В разд. 4 исследуется проблема динамического, постепенного обучения нейросети в процессе ее эксплуатации. В разд. 5 рассматривается построение нейронных сетей с обратными связями. Разд. 6 посвящен построению самообучающихся систем управления. В разд. 7 представлена нейросетевая реализация АЛГОЛ-программы. В разд. 8 аппарат «карт Кохонена», реализуется на универсальной нейросети, позволяющей значительно увеличить число рассматриваемых факторов. Разд. 9 посвящен перспективе применения нейросетевых технологий в сфере обеспечения безопасности, в системах защиты информации, при управлении следованием поездов и др.

Книга рассчитана на любознательных, предприимчивых компьютерщиков и программистов, желающих найти область приложения своим интересам для личных успехов в *Computer Art Studio* в науке, экономике и бизнесе, а также в индустрии развлечений и зрелищ.



## ВВЕДЕНИЕ

В моменты принятия сложных решений человек старается «заглянуть в себя» и постичь, каким образом он справляется с трудными и порой не решаемыми формальной логикой задачами. Естественное беспокойство и жажда познания обуревают его наряду со смутным сознанием того, что математический, алгоритмический подход к построению сложных кибернетических систем искусственно абсолютизирован. Все должно быть к месту, все должно быть взвешено, И обращаясь к себе, он раз за разом проводит мозговую атаку на то таинственное, созданное природой — на собственный мозг...

Итак, мы вступаем в самую сокровенную область искусственного интеллекта.

Мы смелы и безапелляционны. Долго и систематически насаждаемая духовная нищета заставила нас позабыть о превалирующем духовном начале в человеке. Все больше людей понимают, что мозг — лишь инструмент духа, души. И как любой инструмент, он производится, тупится и приходит в негодность — умирает. Бессмертной остается душа — продукт тонкого мира, мира сверхвысоких частот и, по мнению некоторых философов, настойчиво проявляющейся психической энергии.

А раз мозг — инструмент, его надо тренировать, заполнять, совершенствовать. В природе царит принцип целесообразности, полезности в борьбе за главную установку — установку на развитие. Нужное существует и утверждается, ненужное отмирает, как отмирают неиспользованные, «лишние» нейроны, порождая склеротическую ткань и разъедающую интоксикацию. И человека ленивого, праздного, недеятельного охватывает преждевременное старческое слабоумие. «Душа обязана трудиться», — сказал поэт Н. Заболоцкий. Эту истину мы открываем вновь, с другой, неожиданной стороны...

Мы принимаем сказанное и успокаиваемся. Не вторгаемся мы в то, что для нас сейчас непостижимо, что пока нам не дано. Но кто знает? И на всякий случай мы вооружаемся иронией, пре-

одолевая «сопротивление материала». Ирония позволяет и сказать, и не сказать, предположить, но тотчас, прикрываясь шуткой, трусливо отступить. Только иронией можно защитить себя от мистических страхов, возникающих на трудном пути познания.

Внимание ученых к логическому моделированию процессов головного мозга [1—6] обусловили следующие причины:

- высокая скорость выполнения сложных логических конструкций — предикатов с высоким параллелизмом действий;
- простота алгоритмов логических действий мозга, основанная не на численном манипулировании, а на принципах ассоциативного мышления;
- возможность решения трудно формализуемых задач, в которых совместно используются данные логически несовместимой природы, противоречивые, неполные, «зашумленные», некорректные;
- устойчивость работы, совместимая с расширением, трансформированием и совершенствованием знаний;
- надежность, обеспечиваемая наличием многих путей логического вывода и способностью восстановления утраченных данных;
- возможность построения самообучающихся и самонастраивающихся систем;
- прекрасная сочетаемость с традиционными «вычислительными» алгоритмами обработки информации, позволяющая строить сложные системы управления, - с максимальной надежностью, адаптивностью и с минимумом расходуемых ресурсов;
- отсутствие требований к «традиционно» развиваемым вычислительным средствам. Единственный стимулируемый принцип — параллелизм. Для реализации параллельной системы — нейросети — необходима параллельная вычислительная система. Ведь для выявления параллелизма, в частности, мы и обращаемся к модели мозга! Наряду с разработкой параллельных вычислительных устройств — нейрокомпьютеров — стимул к развитию получают сети ЭВМ для реализации в них «больших» нейросетей.

Особый интерес ученых и разработчиков сложных управляющих систем к нейросетевым технологиям, как и к другим технологиям искусственного интеллекта, возник в начале 1980-х годов (известный «японский вызов» [7]), когда остро встала проблема сверхвысокой производительности вычислительных средств.

Приняв этот вызов, многие исследователи обратили внимание на принцип самообучения. Следует отметить краткое, но полное рассмотрение этой проблемы В.В. Игнатушенко [8].

Благотворное влияние на развитие нейросетевых технологий оказало создание методов параллельной обработки информации [9].

Необходимо выразить признательность замечательному хирургу, философу и кибернетику Н.М. Амосову [10], вместе с учениками систематизировавшему подход к созданию средств искусственного интеллекта (ИИ). Этот подход заключается в следующем.

В основе стратегий ИИ лежит понятие *парадигмы* — взгляда (концептуального представления) на суть проблемы или задачи и принцип ее решения. Рассматривают две парадигмы искусственного интеллекта.

1. *Парадигма эксперта* предполагает следующие объекты, а также этапы разработки и функционирования системы ИИ:

- формализация знаний — преобразование экспертом проблемного знания в форму, предписанную выбранной моделью представления знаний;
- формирование базы знаний (БЗ) - вложение формализованных знаний в программную систему;
- дедукция — решение задачи логического вывода на основе БЗ.

Эта парадигма лежит в основе применения экспертных систем, систем логического вывода, в том числе на языке логического программирования ПРОЛОГ. Считается, что системы на основе этой парадигмы более изучены.

2. *Парадигма ученика*, включающая следующие положения и последовательность действий:

- обработка наблюдений, изучение опыта частных примеров — формирование базы данных (БД) системы ИИ;
- индуктивное обучение — превращение БД в БЗ на основе обобщения знаний, накопленных в БД, и обоснование процедуры извлечения знаний из БЗ. Это означает, что на основе данных делается вывод об общности той зависимости между объектами, которую мы наблюдаем. Основное внимание здесь уделяется изучению аппроксимирующих, вероятностных и логических механизмов получения общих выводов из частных утверждений. Затем мы можем обосновать, например, достаточность процедуры

обобщенной интерполяции (экстраполяции), или процедуры ассоциативного поиска, с помощью которой будем удовлетворять запросы к БЗ;

- дедукция — по обоснованной или предполагаемой процедуре мы выбираем информацию из БЗ по запросу (например, оптимальную стратегию управления по вектору, характеризующему сложившуюся ситуацию).

Исследования в рамках этой парадигмы и ее разработка проведены пока слабо, хотя они лежат в основе построения самообучающихся систем управления (ниже будет приведен замечательный пример самообучающейся системы управления — правила стрельбы в артиллерии).

Чем база знаний, общий и обязательный элемент системы ИИ, отличается от базы данных? Возможностью *логического вывода!*

Теперь обратимся к «естественному» интеллекту. Природа не создала ничего лучшего, чем человеческий мозг. Значит, мозг является и носителем базы знаний, и средством логического вывода на ее основе независимо от того, по какой парадигме мы организовали свое мышление, т. е. каким способом заполняем базу знаний, — учимся!

Д.А. Поспелов в замечательной, единственной в своем роде, работе [11] освещает высшие сферы искусственного интеллекта — логику мышления. Цель данной книги — хотя бы частично препарировать нейросеть как средство мышления, тем самым привлекая внимание к низшему, начальному звену всей цепи методов искусственного интеллекта.

Отбросив мистику, мы признаем, что мозг представляет собой нейронную сеть, нейросеть, - нейроны, соединенные между собой, со многими входами и единственным выходом каждый. Нейрон реализует достаточно простую *передаточную функцию*, позволяющую преобразовать возбуждения на входах, с учетом весов входов, в значение возбуждения на выходе нейрона. Функционально законченный фрагмент мозга имеет входной слой нейронов — рецепторов, возбуждаемых извне, и выходной слой, нейроны которого возбуждаются в зависимости от конфигурации и величины возбуждения нейронов входного слоя. Предполагается, что нейросеть, имитирующая работу мозга, обрабатывает не сами данные, а их достоверность, или, в общепринятом смысле, вес, оценку этих данных. Для большинства непрерывных или

дискретных данных их задание сводится к указанию вероятности диапазонов, которым принадлежат их значения. Для большого класса дискретных данных — элементов множеств — целесообразно жесткое закрепление нейронов входного слоя.

Распределение величин возбуждения нейронов выходного слоя, а чаще всего нейрон, обладающий максимальной величиной возбуждения, позволяют установить соответствие между комбинацией и величинами возбуждений на входном слое (изображение на сетчатке глаза) и получаемым ответом (что это). Таким образом, эта зависимость и определяет возможность логического вывода вида «если — то». Управление, формирование данной зависимости осуществляются весами синаптических связей нейронов, которые влияют на направления распространения возбуждения нейронов в сети, приводящие на этапе обучения к «нужным» нейронам выходного слоя, т.е. служат связыванию и запоминанию отношений «посылка — следствие». Связь подструктур нейросети позволяет получать «длинные» логические цепочки на основе подобных отношений.

Отсюда следует, что сеть работает в двух режимах: в режиме обучения и в режиме распознавания (рабочем режиме).

В *режиме обучения* производится формирование логических цепочек.

В *режиме распознавания* нейросеть по предъявляемому образцу с высокой достоверностью определяет, к какому типу он относится, какие действия следует предпринять и т.д.

Считается, что в человеческом мозге до 100 млрд нейронов. Но сейчас нас не интересует, как устроен нейрон, в котором насчитывают до 240 химических реакций. Нас интересует, как *работает нейрон на логическом уровне*, как выполняет он логические функции. Реализация лишь этих функций должна стать основой и средством искусственного интеллекта. Воплощая эти логические функции, мы готовы нарушить основные законы физики, например закон сохранения энергии. Ведь мы рассчитываем не на физическое моделирование, а на доступное, универсальное — компьютерное.

Итак, мы сосредоточиваем внимание на «прямом» использовании нейросетей в задачах искусственного интеллекта. Однако их применение распространяется на решение и других задач. Для этого строят нейросетевые модели со структурой, ориентированной на данную задачу, используют специальную систему связей

нейроподобных элементов, определенный вид передаточной функции (часто используют так называемые сигмоидные связи, основанные на участии экспоненты при формировании передаточной функции), специально подобранные и динамически уточняемые веса. При этом используют свойства сходимости величин возбуждения нейронов, самооптимизации. При подаче входного вектора возбуждений через определенное число тактов работы нейросети значения возбуждения нейронов выходного слоя (в некоторых моделях все нейроны входного слоя являются нейронами выходного слоя и других нет) сходятся к неким величинам. Они могут указывать, например, на то, какой эталон в большей степени похож на «зашумленный», недостоверный входной образ, или на то, как найти решение некоторой задачи. Например, известная сеть Хопфилда [12], хоть и с ограничениями, может решать задачу коммивояжера - задачу экспоненциальной сложности. Сеть Хемминга [5] успешно реализует ассоциативную память. Сети Кохонена (карты Кохонена) [3] эффективно используют принцип кластеризации и широко применяются в экономике, финансах, бизнесе и т.д. Эффективно применяются нейросети для аппроксимации функций многих переменных в виде рекурсивного разложения в базисе передаточной функции.

В указанном выше применении нейросети выступают в роли *спецпроцессоров* для «быстрого» решения частных задач или классов задач. Это можно сравнить с применением аналоговых ЭВМ для решения систем дифференциальных уравнений, где программирование заключается в формировании электрической цепи из элементов заданного набора в соответствии с системой уравнений, а установившийся процесс позволяет на выходе снимать значения функций — решений.

Когда хотят подчеркнуть такие «вычислительные» применения нейросетей, то говорят о *нейроподобных* задачах, и это не должно отвлекать нас от действий в рамках ИИ, направленных на решение трудно формализуемых задач, на простоту и универсальность, свойственные мозгу.

# 1 ОСНОВЫ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

## 1.1 Модель мозга

Нейросеть содержит узлы — аналоги нервных клеток — *нейронов* (*нейроподобных элементов, НПЭ*) и их соединения — *синаптические связи* (рис. 1.1).



Ряс. 1.1. Нейрон

Модель нейрона во взаимодействии с другими нейронами нейросети представлена на рис. 1.2.

Здесь  $V_{ki}$  — энергетические доли импульсов  $V_k$ , выработанных другими нейронами и поступивших на дендриты нейрона  $i$ ;  $\omega_{ik}$  — веса дендритов;  $h_i$  — пороги. В свою очередь, выработанный импульс  $V_i$  также распространяется по дендритам нейронов, с которыми связан нейрон  $i$  с помощью ветвящегося аксона. В соответствии с законом распределения энергии величина  $V_i$  делится пропорционально значениям весов дендритов «принимających» нейронов. (В практических моделях такое распределение энергии обычно не реализуют.)

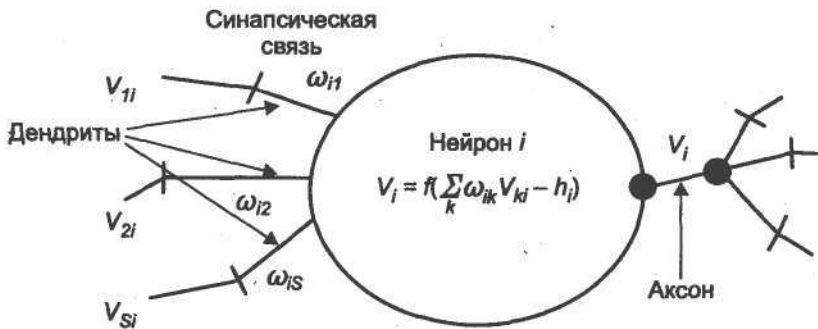


Рис. 1.2. Модель нейрона

Каждый нейрон управляем или извне, или по принципу самоуправления с помощью обратных связей. Можно регулировать значения *весов синапсических связей*  $\{\omega_{ij}\}$  и значения *порогов*  $h_i$ . Такое регулирование, реализованное в разных моделях, и определяет возможность обучения и самообучения сети. Оно задает пути прохождения возбуждений через сеть, простейшим образом формируя связи «посылка — следствие».

Изображенный на рис. 1.3 фрагмент нейросети позволяет представить следующее:

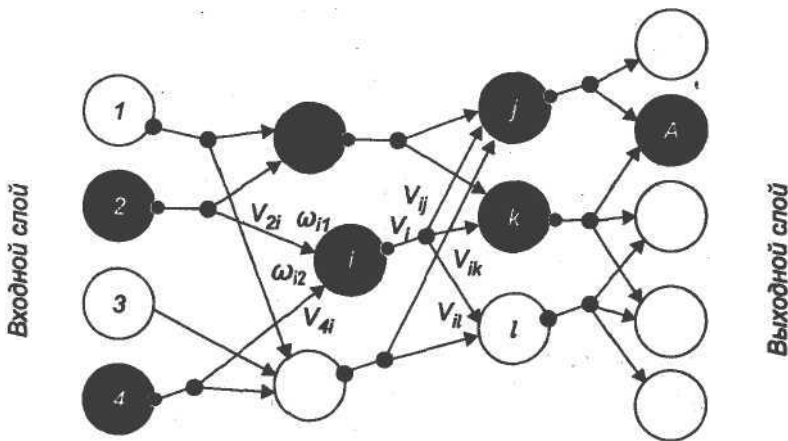


Рис. 1.3. Фрагмент нейросети



- 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 (см. рис. 1.3) производится суммирование и сравнение с порогом:

$$V_i = \xi((\omega_{i1}V_2 + \omega_{i2}V_4) - h), \quad \xi(x) = \begin{cases} 0 & \text{при } x < 0, \\ x & \text{при } x \geq 0; \end{cases}$$

- величина превышения порога является величиной возбуждения нейрона или определяет значение величины возбуждения. Например, в некоторых моделях величина возбуждения всегда равна единице, отсутствие возбуждения соответствует нулю. В других моделях допускают и отрицательную величину возбуждения. Значение возбуждения передается через ветвящийся *аксон* в соответствии со связями нейрона;

- 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;

- в сети распознают *входной (рецепторный) слой*, воспринимающий внешние возбуждения (на который, например, подается видеобразное изображение), и *выходной слой*, определяющий решение задачи. Работа сети тактируется для имитации прохождения по ней возбуждения и управления им;

- сеть работает в двух режимах: *режиме обучения* и *режиме распознавания (рабочем режиме)*.

Установим случайным образом начальные значения весов дендритов всей сети. Пусть нейросеть предназначена для распознавания рукописного текста. Тогда входной слой функционирует аналогично сетчатке глаза, которая воспринимает изображение. Его подача на входной слой возбуждает в некоторой конфигурации множество нейронов — рецепторов. Предположим, на входной слой подан и поддерживается некоторый эталон, например буква *A*. Спустя определенное время максимально возбуждятся некоторый нейрон выходного слоя (прохождение возбуждения отмечено на рис. 1.3 черным цветом). Закрепим этот нейрон за буквой *A*, т.е. его возбужденное состояние будем воспринимать как ответ: «Это буква *A*». Снова введем букву *A*, но с естественными искажениями, обусловленными почерком, дрожанием руки и т.д.

Возможно максимально возбудится тот же нейрон, а возможно — другой. Мы же хотим «научить» систему, заставить ее ответить, что это — тоже буква *A*, т.е. добиться максимального возбуждения того же нейрона выходного слоя.

Тогда необходимо изменить веса и, возможно, пороги в сети на пути прохождения возбуждения так, чтобы заставить возбудиться нужный нейрон. Такая настройка сети является основным, наиболее важным элементом ее обучением. Применяемые для этого алгоритмы называют *алгоритмами обучения*.

Одним из наиболее часто применяемых алгоритмов обучения является *алгоритм обратного распространения ошибки*. В нем воспроизводится подход, используемый в динамическом программировании и основанный на обратном и прямом проходах. Здесь мы предлагаем новый, более простой алгоритм "трассировки", основанный на прямом проходе при прокладывании путей; возбуждения от эталонов к нейронам выходного слоя.

Предъявляя множество *эталонов* и регулируя параметры, производим обучение сети данному образцу. Математические проблемы несовместимости управления параметрами для разных эталонов в данном случае не рассматриваются: в природе такой процесс проходит успешно, достаточно устойчиво при разумном отличии эталонов. Ясно, например, что заставлять один и тот же нейрон выходного слоя возбуждаться и на строчную, и на прописную букву *A* вряд ли разумно. В лучшем случае он определит, что это вообще буква, а не 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. Предъявляем сети различные буквы. Можно быть уверенным, что, если мы предъявим случайно искаженную и даже зашумленную букву *A* (конечно, в допустимых пределах), сеть ее с большой вероятностью распознает, т. е. максимально возбудится соответствующий нейрон выходного слоя.

Можно существенно облегчить обучение, предъявляя эталон «в полном смысле», т.е., например, показывая букву, регламентированную букварем. Тогда на вероятность распознавания этой буквы будет влиять предусмотренная степень отклонения от данного эталона, обусловленная почерком. Далее необходимо придерживаться именно такого способа обучения.

Расширим функцию входного слоя, связав его не с изображением, а в общем случае с некоторыми характеристиками исходной ситуации (входного вектора), по которой необходимо принимать решение — формировать выходной вектор. Тогда сеть обучается по эталонным ситуациям, решение которых известно, а затем в рабочем режиме выдает решение во всем диапазоне ситуаций и при этом автоматически определяет, на какую «знакомую» ситуацию более всего похожа введенная и, следовательно, какое решение надо принять.

Такая обработка входной информации при возможном использовании в сфере развлечений показана на рис. 1.4.

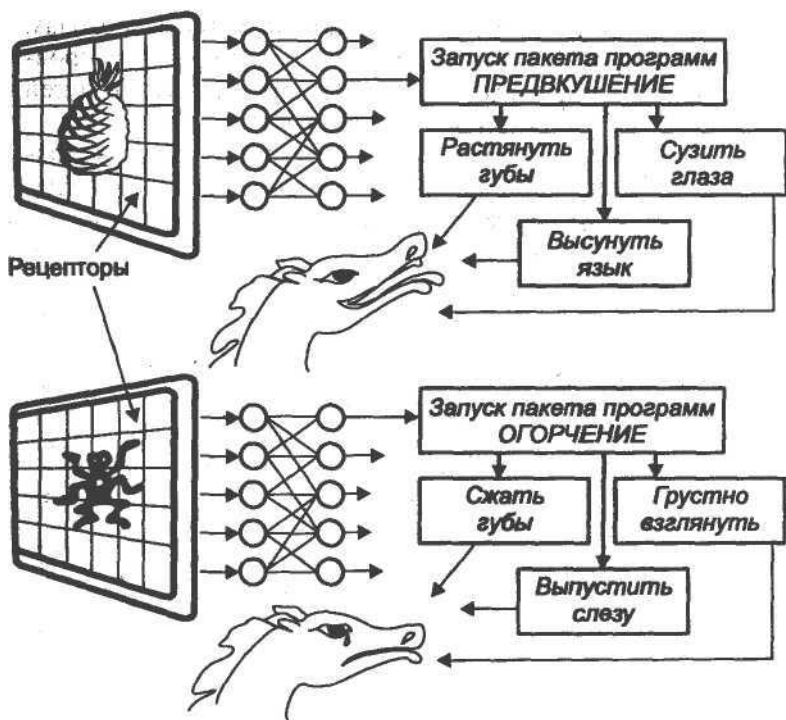


Рис. 1.4. Моделирование реакции на изображение

Отметим, что нейронная сеть выдает заключение в виде ответа на вопрос: «На какую ситуацию похожа данная и, следовательно»

но, какое должно быть принято решение?» Это еще раз подтверждает, что нейросеть *имитирует ассоциативное мышление*.

## 1.2 Устойчивость и помехозащищенность

В рассмотренном в разд. 1.1 примере относительно жесткого закрепления нейронов выходного слоя между образами принцип ассоциативности или ответ на вопрос: «На что более всего это похоже?», пока не просматривается. Как же он появляется?

Представим себе выходной слой нейросети (рис. 1.5). Рассматривая прохождение возбуждения, например, при подаче того же изображения буквы *A*, считаем, что в возбужденное состояние придет не один нейрон, строго соответствующий (по нашему желанию) этой букве, — в его окрестности возбудятся и другие нейроны. Причем возбуждение будет угасать по мере удаления. Нам же необходимо, чтобы максимальной величиной возбуждения в этой окрестности обладал именно отмеченный нейрон. Более того, желательно, чтобы эта величина возбуждения была существенно выше величин возбуждения других нейронов. Это обеспечит однозначность при последующем использовании полученного вывода в построении «длинных» логических цепочек для принятия решения. Такая локализация возбуждения позволяет с большой определенностью ответить на вопрос: «На что *более всего* похож предъявляемый образ, несмотря на случайные отличия его от эталона и помехи?»

Отметим существующую в природе (реализованную в мозге) непрерывность перерастания свойства похожести одних образов в другие. Следование этому принципу при принудительном закреплении нейронов выходного слоя способствует построению адекватных моделей. Говоря точнее, подавая первый раз букву *A* при случайном начальном выборе весов, 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 необходимо добиваться при предъявлении других эталонов.

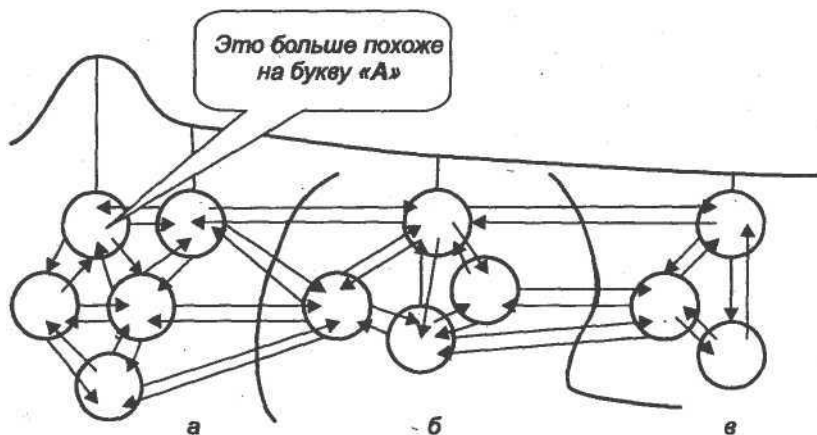


Рис. 1.5. Возбуждение выходного слоя:  
*а* — область возбуждения букв; *б* — область возбуждения цифр;  
*в* — область возбуждения специальны\* знаков

Далее при построении алгоритма обучения предположим, что первоначально сеть «пуста», т.е. все веса синаптических связей равны нулю. Путь возбуждения (связь «посылка — следствие») прокладывается («трассируется») с помощью присваивания некоторым весам положительного значения.

Способы максимизации и локализации уровня возбуждения основаны на нахождении экстремума функции возбуждения, построенной в области выходного слоя. Здесь не обойтись без обмена тормозящими воздействиями между нейронами выходного слоя. Суть идеи в следующем. Все нейроны выходного слоя имеют синаптические связи, такие, что каждый нейрон связан с ближайшими нейронами тормозящими связями, по модулю пропорциональными величине собственного возбуждения. В результате взаимодействия двух нейронов (один из алгоритмов такого взаимодействия будет рассмотрен ниже) более «сильный» сохранит положительный (хотя и меньший) потенциал, сообщив более «слабому» 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.

## 1.3 Вводи «разглядывание» эталонов и образов

Устройства ввода информации — эталонов, входных векторов, исходных ситуаций — имеют для нейросети определяющее значение. С их помощью формируются и поддерживаются возбуждения входного слоя. Однако связь модели живого организма с внешней средой сложнее. Это настоящий конгломерат ряда частных технических и алгоритмических проблем, к которой относится успешно решаемая проблема видеоввода. Но ввести в компьютер «картинку» — это лишь подделка. «Картинку» надо обработать в целом и по частям, чтобы получить полную и достоверную информацию. В результате имеем аналог нашего восприятия действительности.

Существуют три способа обзора представляемого изображения:

- сканирование сектора обзора, разбитого на элементарные сегменты;
- сканирование сектора обзора со «своим окном просмотра»;
- спонтанный обзор, обусловленный вниманием к цветовому или скоростному всплеску, быстрому увеличению размера (угрожающему приближению) объекта, указанию извне (целеуказанию) и т.д.

Третий способ также основан на сканировании сектора обзора, однако со значительно меньшими энергетическими затратами.

При первом и втором способах анализ сложнее, так как требует согласования виденного по сегментам, что, в свою очередь, приводит к включению высших уровней логического вывода (интеллекта).

Применив третий способ, можно добиться избирательности, чрезвычайности реакции, например, на резкие движения, на бег, на появление яркой расцветки в одежде и т.д., что может с успехом использоваться в игровых системах.

Все способы реализуются легче, если речь идет о единственном объекте единовременного распознавания - букве, хозяине квартиры, подписи и т.д. В случае, если существует множество объектов, например туристская группа, любующаяся «умным»

монстром, необходим не только детальный, но и совместный анализ этим монстром всех (многих) ее составляющих.

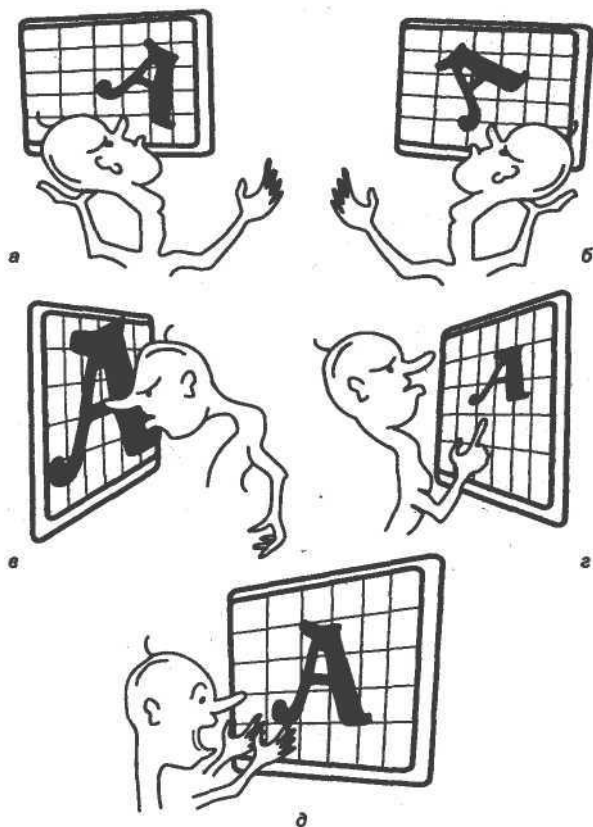
Впрочем, говоря о туристах, можно вести речь о конечной, усредненной реакции монстра на всю группу. Производя обзор, управляющая им нейросеть постепенно согласно критериям обучения «хорошо — плохо» воспринимает настроение как последовательное проявление радости и огорчения и приходит к некоторому окончательному состоянию, обусловленному количественным эквивалентом того и другого. Для разных групп туристов или экскурсантов это состояние будет разным: либо источником веселья и шутивого «поощрения» той группы, которая привела объект в радость, либо источником «осуждения» группы, ввергнувшей его в печаль.

Итак, на каждом такте обзора формируется сегмент, содержание которого необходимо распознать. Чаще всего целесообразно допущение о том, что в элементарном сегменте (или в «коне 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 разглядеть эту букву? По-видимому, следует разместить ее на входном слое так, чтобы она максимально соответствовала тому размещению эталонов, с помощью которых производилось обучение. Тогда распознавание пойдет корректно (рис. 1.6). Такой процесс «разглядывания» предполагает:

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

В результате таких пробных действий может вдруг начаться процесс распознавания, пусть и ошибочного.

Указанный процесс выделения и размещения в попытке инициировать распознавание сравним с концентрацией внимания и с фокусировкой: ведь зрение в каждый момент времени сконцентрировано на элементе изображения посредством фокусировки на нем глаз в процессе обзора и разглядывания всего изображения.



**Рис. 1.6.** Распознавание буквы:  
*a* — наклон головы влево; *б* — наклон головы вправо; *в* — близко;  
*г* — далеко; *д* — угадал

## 1.4 Пространство признаков

Рассмотрим подробнее процесс восприятия, например, строчной буквы *a* рецепторами — входным слоем нейросети. Конфигурация возбужденных рецепторов, порождая прохождение возбуждения через внутренние слои нейросети (образуя путь возбуждения), задавала возбуждение (не затухшее по пути) одного из нейронов выходного слоя, определяющего: "Это буква *a*".



Устойчивость подобной схемы по отношению к множеству конфигураций возбуждений рецепторов, соответствующих одной только букве *a*, вряд ли высока. Такое непосредственное распознавание осуществляется далеко не всегда, ибо наше образование базируется на *признаках и определениях (правилах вывода)*.

Как характеризуется строчная буква *a*? Это кружок, к которому справа примыкает палочка с хвостиком вправо. А прописная буква "А"? Две палочки с перекладиной. Но ведь буква *H* тоже соответствует этому определению. Тогда следует добавить: прописная "А" — это две палочки, соединенные вверху. (Кстати, соединение вверху может быть из-за небрежности лишь обозначено. Тогда о намеке на него можно судить по наклону палочек. Дальнейшая небрежность приведет к неразличимости букв *A* и *Я*.)

Значит, существует ряд признаков, лежащих в основе определений. Ясно, что такой способ распознавания гораздо устойчивее к искажениям и особенностям почерка, однозначен и надежен. Ведь при изображении буквы *A* возможны не только небрежность в верхнем соединении палочек, но и различный наклон буквы, длина боковых палочек, место расположения перекладины, ее наклон, и т.д. Искажение приведет к сомнениям лишь при сходстве с цифрой 4, с телеграфным столбом или с греческой буквой *Л*. Но даже в этом случае окончательный вывод может быть сделан на основе контекста, т.е. по использованию дополнительных признаков.

Значит, в нашем случае необходимо ввести такие признаки, как наличие кружка, палочек, хвостиков, их взаимного расположения, т.е. построить пространство признаков, преобразовать в это пространство входные изображения, и тогда получим более определенное и устойчивое к искажениям заключение.

Перевод входного изображения в пространство признаков значительно расширяет возможности обзора — масштабирования, размещения, поиска угла наклона, т.е. позволяет с более высокой достоверностью производить распознавание.

Например, изображение танка может в разных ракурсах накладываться на входной слой рецепторов и запоминаться: «и это — танк», «и это — тоже танк» и т.д. Но если ввести определение «танк — это массивный корпус на гусеничном ходу, а сверху — башня с дулом пушки и все такое зелененькое», то в

любом случае при распознавании будут приняты меры предосторожности.

## 1.5 Кора

Понятно, что примитивная оптимизация целевой функции, принимающей только два значения: «хорошо» и «плохо», может быть использована лишь на начальной стадии изучения и применения принципов нейросети. В частности, на таких принципах строится модель собачки Павлова, демонстрирующая условный рефлекс при предъявлении этикетки «Pedigree».

Цепочки логических выводов человека 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. Возникает вопрос о целесообразной структуре сети, где 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 умозаключение.

При какой же концентрации выводов совмещаются понятия входного и выходного слоев? Видимо, целесообразно существование универсального слоя — коры, со связями между отдельными нейронами и целыми областями, где сформировавшиеся выводы используются для получения других выводов.

## 1.6 Локализация максимального возбуждения на выходном слое

Проанализировав сказанное выше, попытаемся собрать некоторую универсальную модель нейросети с входным и выходным слоями. Картина возбуждений выходного слоя при подаче

изображения на входной слой после аппроксимации в непрерывную область определения из дискретной будет представлять собой непрерывную функцию, максимум которой определяет необходимое заключение. Однако на этапе обучения, а также в рабочем режиме с удовлетворительной достаточно малой вероятностью по ряду предъявляемых эталонов этот максимум не совпадает с нужным ответом. Следовательно, путь распространения возбуждений внутри сети необходимо скорректировать изменением синаптических весов задействованных нейронов.

Как было упомянуто выше, предварительная локализация максимума величины сигнала производится в результате взаимодействия нейронов в области или некоторой окрестности выходного слоя или коры. Области возбуждений «закрепляются» за типами объектов — за буквами, цифрами, продуктами питания и т.д. В развитой сети с актуальным понятием коры нейроны отдельных ее областей через нейроны внутренних слоев вновь порождают пути прохождения возбуждений в другие области коры и т.д.

Пытаясь проанализировать и воспроизвести универсальную нейросеть, видим, что отдельный нейрон может взаимодействовать с огромным числом нейронов (нейрон имеет до 10 тыс. дендритов), но это взаимодействие носит *локальный* характер. А именно, несмотря на случайность связей, вероятность связи с «ближним» нейроном значительно выше вероятности связи с нейроном «далеким». Об этом свидетельствуют исследования длины дендритов и аксонов: длина дендрита достигает миллиметра, а длина аксона — сотни миллиметров. При этом применяется, по-видимому, усредненная характеристика, вряд ли отражающая нейроны только головного мозга.

Такой принцип локальности, пронизывающий всю структуру сети, в сочетании с принципом *иерархии* — возможностью построения новых выводов на основе сделанных, позволяет реализовать связи «каждый с каждым». Никакой вывод не остается 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.

Данный принцип не отвергает существования маловероятных связей «каждого с каждым». Эта вероятность может быть высокой вследствие аномалий генетического характера. Например, некий уникум способен «видеть» кожей: нейроны, воспринимающие кожные ощущения, сильно связаны с нейронами выходного слоя, «отвечающими» за зрение. При слабых связях распространение возбуждения от кожных рецепторов в сторону «зрительных» нейронов гаснет. В данном же случае оно становится результативным, поскольку в целом все нейроны устроены одинаково.

Большое число связей способствует высокой надежности мозга. Ведь ежедневная гибель огромного числа нейронов, подхлестнутая 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.

Рассмотрим теперь 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, которые не соответствуют нейрону с максимальным возбуждением т.е. концентрация сигнала и выделение нейрона с максимальной величиной возбуждения достигаются с помощью подавляющих связей «соседних» нейронов выходного слоя.

Пусть нейронам выходного слоя, условно расположенного на плоскости  $(x,y)$ , соответствует непрерывная функция возбуждения  $P(x,y)$  (рис. 1.7), обусловленная прохождением сигналов возбуждений в сети на основе предъявленного эталона. Будем считать, что эта функция имеет один или более максимум. 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 регулируемые весами первоначальную величину возбуждения  $P_{ij}$ , полученную из сети.

Представим взаимодействие соседних нейронов, например,  $(i,j)$  и  $(i,j+1)$  разными значениями величин возбуждения. Пусть  $P_{ij} > P_{i,j+1}$ . Тогда в течение очередного периода тактовой частоты на входе нейрона  $(i,j)$  появится подавляющий сигнал  $P_{i,j+1}$ , а на

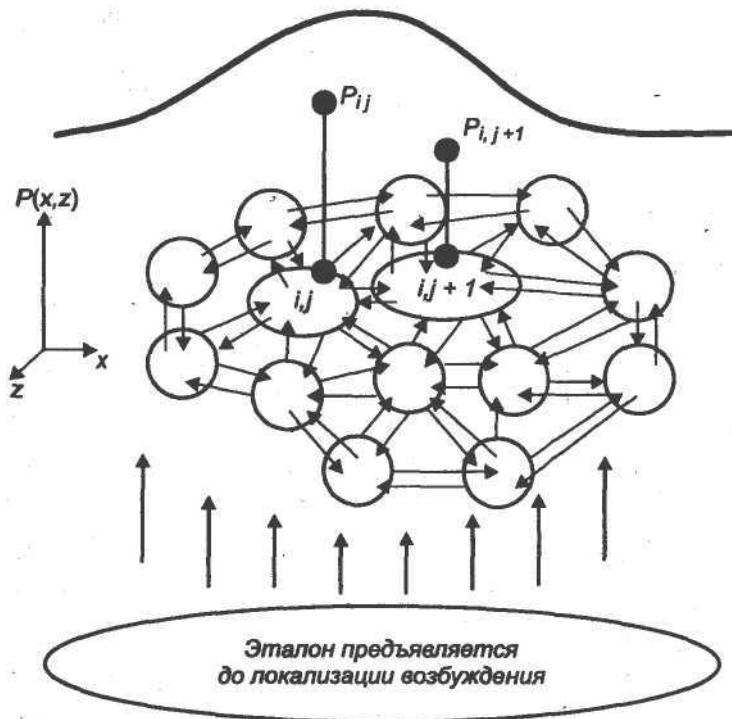


Рис.1.7. Локализация возбуждения нейронов выходного слоя

входе нейрона  $(i, j+1)$ , имеющего меньшее значение величины возбуждения, — большой подавляющий сигнал  $P_{ij}$ .

При предъявлении на входном слое эталона, т.е. при существовании в некоторый период времени характера и величины возбуждений, обуславливающих возбуждение выходного слоя, между нейронами  $(i, j)$  и  $(i, j+1)$  происходит перераспределение величины возбуждения. А именно значение  $P_{ij}$ , несомненно, уменьшится, но в еще большей степени уменьшится значение  $P_{i,j+1}$ . В следующем такте «сильный» нейрон еще более «ослабит» «слабый» нейрон, который, в свою очередь, сможет еще в меньшей степени «ослабить» более «сильный» нейрон и т.д. Так, при достаточно долгом показе эталона «слабый» нейрон может «слабеть» до тех пор, пока взвешенная сумма подаваемых ему сигналов не станет меньше его порога.

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 усиление сигнала наиболее возбужденного нейрона. Полное подавление сигнала возможно только 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 прежде имел более слабый сигнал возбуждения. Тогда возможно появление в указанной области возбуждения нового локального максимума. Таким образом, веса отрицательных связей должны способствовать максимизации возбуждения того нейрона, который первоначально продемонстрировал максимальное возбуждение при затухании возбуждения нейронов в сторону периферии.

Но всегда ли следует на выходном слое локализовать величину возбуждения?

По-видимому, такое усиление возбуждения используется для того, чтобы единственный нейрон выходного слоя преодолел некоторый порог, будучи ответственным за сделанный вывод или решение.

Локализация и максимизация возбуждения на выходном слое особенно важны тогда, когда действительно необходима высокая степень определенности, например в том случае, если получаемый вывод (решение) немедленно участвует в цепочке последующих выводов.

В конце концов все обусловлено назначением сети решаемой задачей. Можно представить себе аттракцион — реакцию фантастического чудовища на изображение, как это показано на рис. 1.4. По виду изображения инициируются те или иные программы — радости, гнева, поднятия лап, виляния хвостом и т.д. Возбуждение определенных нейронов выходного слоя связывается с запуском соответствующих программ, где основным параметром является величина возбуждения. Программы не исключают друг друга и могут запускаться в одном такте. При этом предпочтителен запуск только тех программ, значение параметра которых превышает некоторый порог.

В большинстве частных случаев, когда 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.

Отметим в заключение, что научный аспект проблемы развития и внедрения нейросетевых технологий базируется на философско-математическом представлении задач ИИ, выражающемся в формализации *процесса мышления*.

В работе [11] Д. А. Пospelов, в частности, исследует теорию Аристотеля - силлогистику и предлагает принципы ее моделирования. Тем самым указывается важность формализации *механизма мышления*. Этим механизмом уже давно, со времен Лейбница и до появления булевой алгебры (вплоть до наших дней), является математическая логика, отображенная во многих работах выдающихся математиков.

Математическая логика, ее важный раздел «Алгебра высказываний», действительно соединили принципы мышления и их автоматизированное воплощение.

Однако для реализации мышления природа не создала ничего лучшего, чем человеческий мозг. Он является гигантской нейросетью, фиксирующей причинно-следственные связи, создающей базу знаний и владеющей процедурами логического вывода.

Таким образом, нейронные сети реально являются основой формализации *средств мышления*. Поэтому справедливо считать, что исследование нейронных сетей опирается на достижения математической логики.

# 2 ПОСТРОЕНИЕ СОВРЕМЕННОЙ НЕЙРОСЕТЕВОЙ ТЕХНОЛОГИИ

## 2.1 Построение примитивной нейросети

Говоря о вещах несогласуемых, мы вспоминаем поговорку: «В огороде бузина, а в Киеве дядька». Как же связать объекты, между собой не совместимые, вызывающие усмешку, но, несомненно, влияющие на раздумья о смысле жизни?

Здесь мы сталкиваемся с действительно трудно формализуемой задачей, требующей мобилизации нашего ассоциативного мышления.

Существует мнение о том, 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 становится не коммерсантом, а, скажем, блестящим специалистом по реликтовым захоронениям мезозойского периода. Более того, если человек талантлив, то он талантлив во всем.

В книге Э. Мулдашева «От кого мы произошли?» в главе «Почему мы смотрим друг другу в глаза?» автор говорит об известных ему 22 параметрах, отражающихся на глазной области лица и динамически описывающих морально-психологическое состояние собеседника — партнера. Однако обчислить на компьютере удалось лишь два. Автор удивлен той колоссальной скоростью, с которой ведет описанную обработку человеческий мозг: «Воистину велик Бог, создавший такое компьютерное совершенство мозга!»

Мы уже не удивляемся высокой скорости работы мозга. Известно, что это — процесс распараллеливания обрабатываемой информации, и без параллельных вычислительных средств зани-



маться моделированием работы мозга, т.е. «больших» нейросетей, — бессмысленно.

Именно особенности параллельной обработки информации наряду с возможным отсутствием формальных расчетов (только на основе ассоциативного мышления) привлекли к себе внимание проектировщиков суперЭВМ в период так называемого японского вызова, в начале 1980-х годов.

Нейросетевые технологии основаны на моделировании деятельности мозга, а по классическому утверждению Р. Шеннона [14], «моделирование есть искусство». И как каждое искусство, модель нейросети базируется на примерах, опыте и общих рекомендациях.

Рассмотрим пример, навеянный временем перестройки.

Вася и Петя — друзья. Обозначим  $A$  — множество друзей:

$$A = \{A1, A2\} = \{\text{Вася}, \text{Петя}\}.$$

Вася и Петя создали «крышу», под которой успешно трудится ряд палаток:

$$C = \{C1, C2, C3, C4, C5\} = \{\text{«Оксана»}, \text{«Роксана»}, \text{«Марина»}, \text{«Регина»}, \text{«Св. Аполлинария»}\},$$

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

$$B = \{B1, B2, B3\} = \{\text{Красный киллер}, \text{Пират}, \text{Ночная бабочка}\}.$$

Фирма *Красный киллер* производит винно-водочную продукцию отличного качества. Фирма *Пират* специализируется на аудио- и видеопродукции и других культурных ценностях, а фирма *Ночная бабочка* — на французской косметике из мосластных 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 посетил 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 голо-

пузых апологетов трудного детства, которые с некоторой долей достоверности доставляют ему информацию.

Дядя Рамзай — прогрессивный бизнесмен, и оценки прибыли производит с помощью математического аппарата, обратившись за помощью к нам. А информации дяде Рамзаю приходится обрабатывать 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. Аполлинаруя получила покровителя в местной мэрии и разъезжает в длиннющем "линкольне". Тщетно пытаясь разрушить узы бескорыстной дружбы, фирма *Пират* подкупила Васю, снизив нагрузку вымогательства на свою продукцию. Петя, кажется, пошел на нарушение «закона» и вторгается в область, контролируемую конкурентами. Все такие обстоятельства прямо или косвенно влияют на долю прибыли дяди Рамзая.

А мы, принимая заказ на очень трудно формализуемую задачу, хотим уяснить, что нужно дяде Рамзаю, и устанавливаем:

- он хочет, задавая исходные данные на основе неполной или недостоверной информации своих агентов, все-таки распознать с наибольшей определенностью, что это за ситуация и на какую прибыль можно рассчитывать;
- он хочет, задавая исходную ситуацию с некоторой достоверностью, установить среднюю величину прибыли;
- он хочет сделать вывод о частоте появления различных ситуаций, чтобы перераспределить таксу за молчание о шалостях Васи и Пети.

Итак, абстрагируемся.

Пусть по стечению обстоятельств, которые мы будем называть событиями, принимаются решения  $R1, R2, R3, R4, R5$ .

События выражаются записями, например, вида  $\{A1, B2, C3\}$  или  $A1 \wedge B2 \wedge C3$ . Это означает, что Вася ( $A1$ ) отправился в палатку Марины ( $C3$ ), торгующей в этот раз продукцией фирмы *Пират* ( $B2$ )

Решения образуют конечное множество. Каждое решение носит сугубо личностный характер и определяет тактику дальнейшего поведения, надежду на существенное приобретение, турпоездку и т.д.

Одно решение соответствует некоторой, в общем случае не единственной, комбинации событий. Для изображения таких

комбинаций воспользуемся записями, например, вида  $A1 \wedge (B2 \vee \vee B3) \wedge (C4 \vee C5)$ . Это означает, что Вася отправился в палатку то ли к Регине, то ли к Аполлинарии, торгующей продукцией то ли фирмы *Пират*, то ли фирмы *Ночная бабочка*. Однако эта же запись означает, что все составляющие ее конъюнкции  $A1 \wedge B2 \wedge \wedge C4$ ,  $A1 \wedge B3 \wedge C4$  и другие 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:

$$\begin{aligned}
 & \text{if } A1 \wedge B1 \wedge (C1 \vee C2 \vee C3 \vee C4 \vee C5) \text{ then } R1; \\
 & \text{if } A1 \wedge (B2 \vee B3) \wedge (C1 \vee C2 \vee C3) \text{ then } R2; \\
 & \text{if } A1 \wedge (B2 \vee B3) \wedge (C4 \vee C5) \text{ then } R3; \\
 & \text{if } A2 \wedge B3 \wedge (C1 \vee C2 \vee C3 \vee C4 \vee C5) \text{ then } R4; \\
 & \text{if } A2 \wedge (B1 \vee B2) \wedge (C1 \vee C2 \vee C3 \vee C4 \vee C5) \text{ then } R5.
 \end{aligned}
 \tag{2.1}$$

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.

## 2.2 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, 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 — 100), [100 - 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. Но 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 (ведь интуиция основана на опыте), определим достоверность того, что интересующая нас величина принадлежит данному диапазону. С какой достоверностью она принадлежит второму диапазону? А третьему? Можно даже уйти от понятия достоверности как вероятностной категории. Все ли мы знакомы с понятием полноты множества событий? Ведь часто слышим: «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 рас-

считается 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 «чероки» 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 джипа «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.

## **«Схемотехнический» подход 2.3 к построению нейросети «подзадачу»**

Реализуем подход, используемый при построении схем устройств ЭВМ 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. 2.1) схему, реализующую алгоритм счета значения выражения (2.1). Предпо-

лагаем подавать на вход значения булевых переменных, обозначающих события.

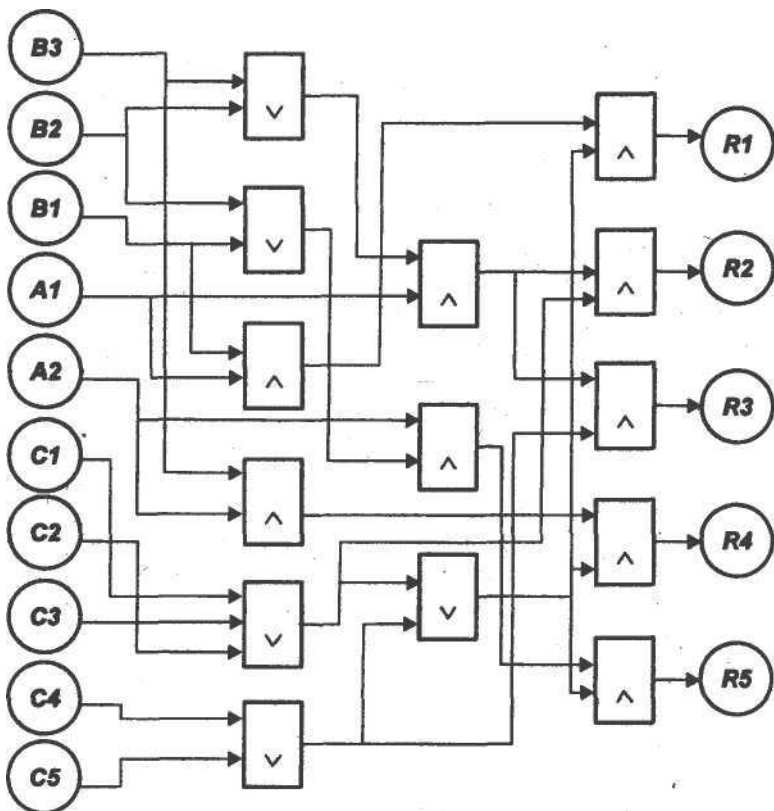


Рис. 2.1. «Схмотехническое» решение

Такая электронная схема могла бы нам верно служить, способствуя быстрому определению необходимой реакции на сложившуюся ситуацию, если мы предусмотрели все возможные ситуации, знаем, какое решение соответствует каждой из них, и всегда обладаем 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, пытаясь 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 помогает 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 (/ -го) нейрона с числом  $m$  входов — дендритов:

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

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

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

Тогда *нейрон-конъюнктор* может быть реализован с помощью существенно высокого порога (рис. 2.2), где значение  $\delta$  обусловлено некой поправкой, достаточной, чтобы для преодоления порога сигналы возбуждения поступали обязательно *по всем* входам.

При обучении предполагается, что входные сигналы — булевы переменные, принимающие значения 0, 1. Положим  $\omega_{ij} = 1/m$  выберем  $\delta < 1/m$ . Тогда для преодоления порога 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.

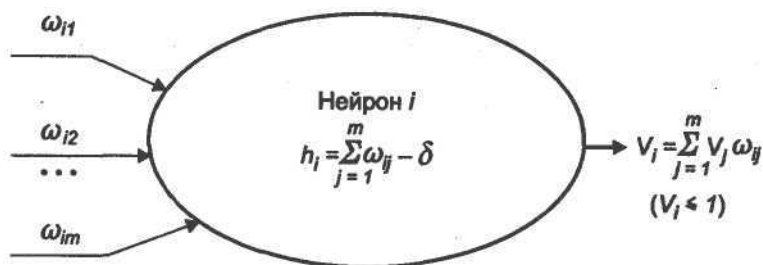


Рис. 2.2. Нейрон-конъюнктор

При переходе к действительным переменным, когда вместо событий рассматриваются, например, лишь предполагаемые вероятности их наступления, экспериментальный выбор значения  $\delta$  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 возник сигнал возбуждения на выходе. При этом сигнал на выходе не превышает «1» (рис. 2.3).

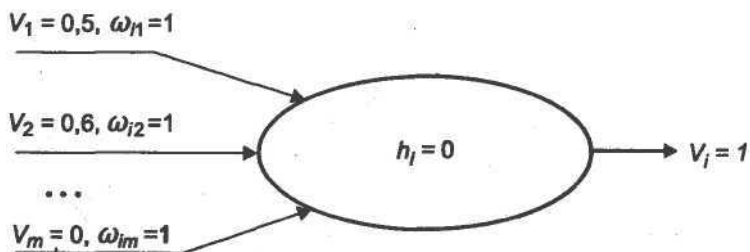


Рис. 2.3. Нейрон-дизъюнктор

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. 2.1. Получим нейросеть (рис. 2.4), где нейроны-конъюнкты заштрихованы.

Теперь позволим дяде Рамзаю поучиться, поэкспериментировать, задавая различные достоверности событий — возможных или невозможных. Например, зададим «правильную» и абсолютно достоверную ситуацию  $B3=1, A1=1, C4=1$  (Вася отправился к Регине, торгующей ямайским ромом). Легко проследить, что в первом такте возбуждятся 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 возбуждятся.

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 Аполлинарии, 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  $V_{B1} = 0,8$ ,  $V_{B2} = 0,8$ ,  $V_{C1} = 0,7$ ,  $V_{C3} = 0,8$ .

Напоминаем еще раз, что требовать полноты 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 конъюнктов равными 0,5, а дизъюнктов, — равными 1. Пороги конъюнктов определяются значением  $\delta = 0,4$ . Пороги дизъюнктов имеют нулевое значение.

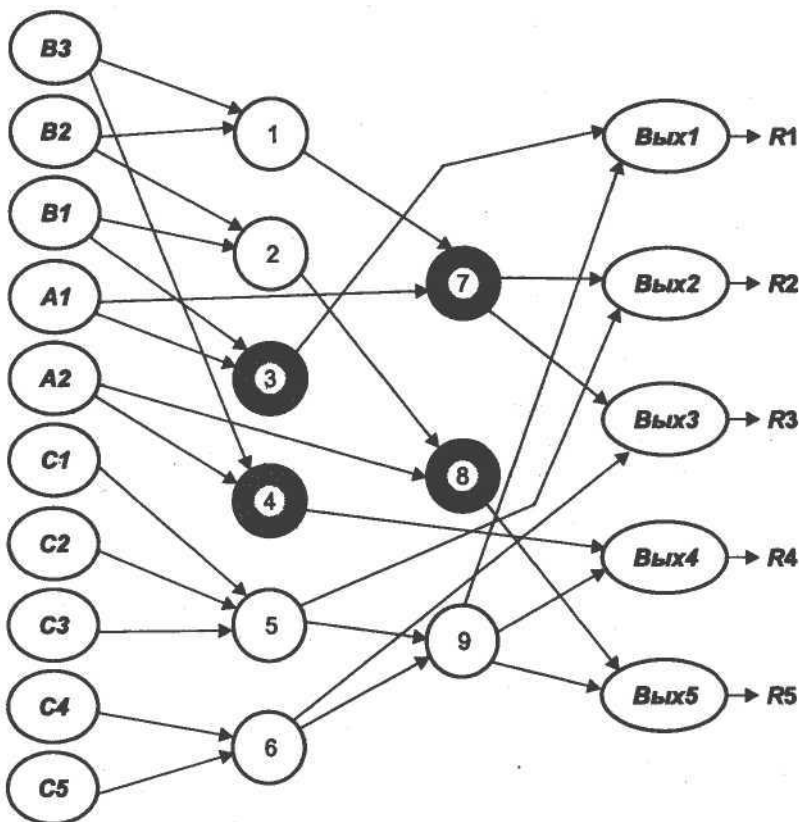


Рис. 2.4. Нейросеть с конъюнкторами (выделены) и дизъюнкторами

Важность данного примера требует повторения рисунка нейросети (рис. 2.5) с проставленными возле нейронов значениями сигналов возбуждения.

В итоге ситуация скорее всего имеет решение  $R_5$  и уж никак не  $R_4$ . Однако ситуация, соответствующая решению  $R_1$ , требует внимания и т.д.

Пусть при вполне определенной ситуации (все достоверности принимают значение «1») каждое решение  $R_i$  приносит прибыль  $M_i$ . В таком случае средняя величина ожидаемой прибыли для нашей неопределенной ситуации рассчитывается так:

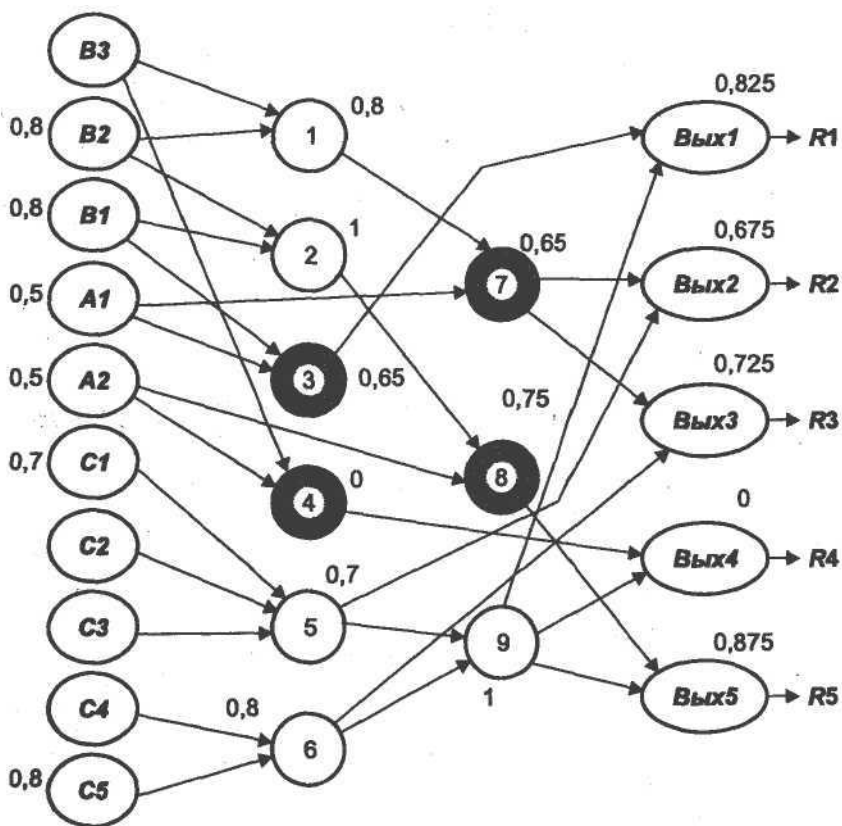


Рис. 2.5. Нейросеть с рассчитанными значениями возбуждений

$$M = \frac{\sum_j M_j V_{\text{Вых}j}}{\sum_j V_{\text{Вых}j}} = \frac{M_1 \cdot 0,825 + M_2 \cdot 0,675 + M_3 \cdot 0,725 + M_5 \cdot 0,875}{3,1}$$

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

## 2.4 Построение нейросети «под задачу»

Мы построили нейросеть с экзотическими (с точки зрения психотерапевта) конъюнкторами и дизъюнкторами.

Предположим теперь (см. рис 2.4, 2.5), 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

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

При  $\omega_{ij} = 0,8$ ,  $A = 0,2$  сформирована сеть, представленная на рис. 2.6.

Подадим на вход, например, ситуацию  $\{A1, B2, C3\}$ , требующую решения  $R1$ . Величины возбуждений нейронов показаны на рис. 2.6.

На основе расчетов по полученной сети составим табл. 2.1, 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 2.1

Примеры расчета принимаемых решений

Ситуация	Требуемое решение	$V_{Вых1}$	$V_{Вых2}$			$V_{Вых5}$
{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

Ситуация	Требуемое решение	$V_{Вых1}$	$V_{Вых2}$	$V_{Вых3}$	$V_{Вых4}$	$V_{Вых5}$
{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

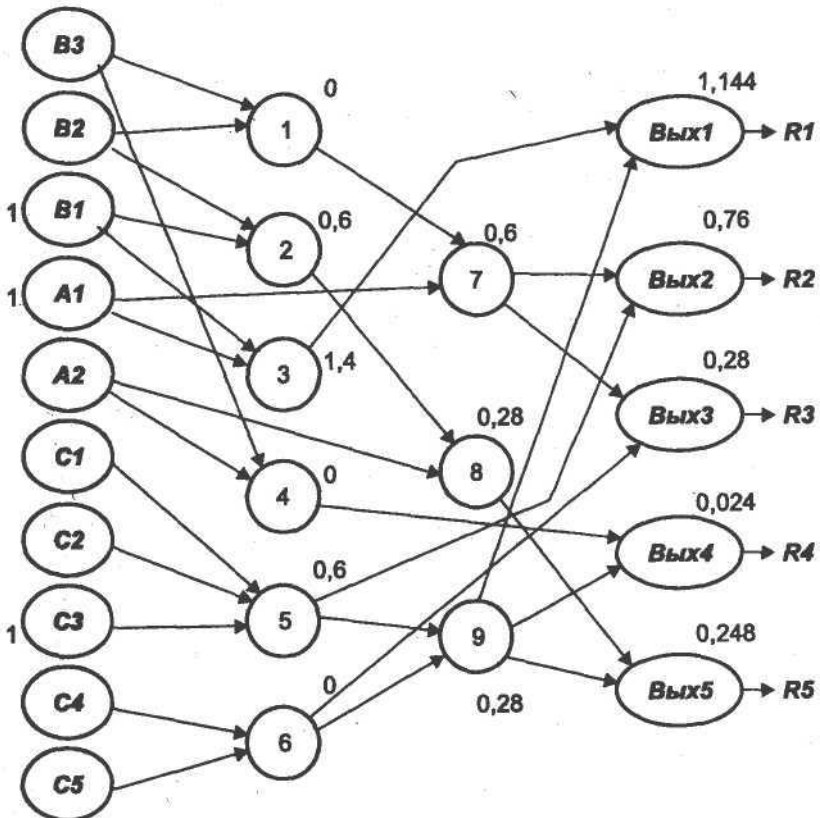


Рис. 2.6. Нейросеть с «универсальными» нейронами

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 эта ситуация более всего ситуации, когда Петя направился к Аполлинару, 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 (см. рис. 2.5), 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. Отсюда широкое распространение так называемых однослойных сетей. Построим такую сеть и для нашего примера (рис. 2.7).

Возьмем ту же передаточную функцию с теми же параметрами и рассчитаем те же примеры, отображенные в табл. 2.1. Составим для них табл. 2.2.

Таблица 2.2

Примеры расчета решений по однослойной нейросети

Ситуация	Требуемое решение	$V_{Вых1}$	$V_{Вых2}$	$V_{Вых3}$	$V_{Вых4}$	$V_{Вых5}$
{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

Ситуация	Требуемое решение	$V_{Вых1}$	$V_{Вых2}$	$V_{Вых3}$	$V_{Вых4}$	$V_{Вых5}$
{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

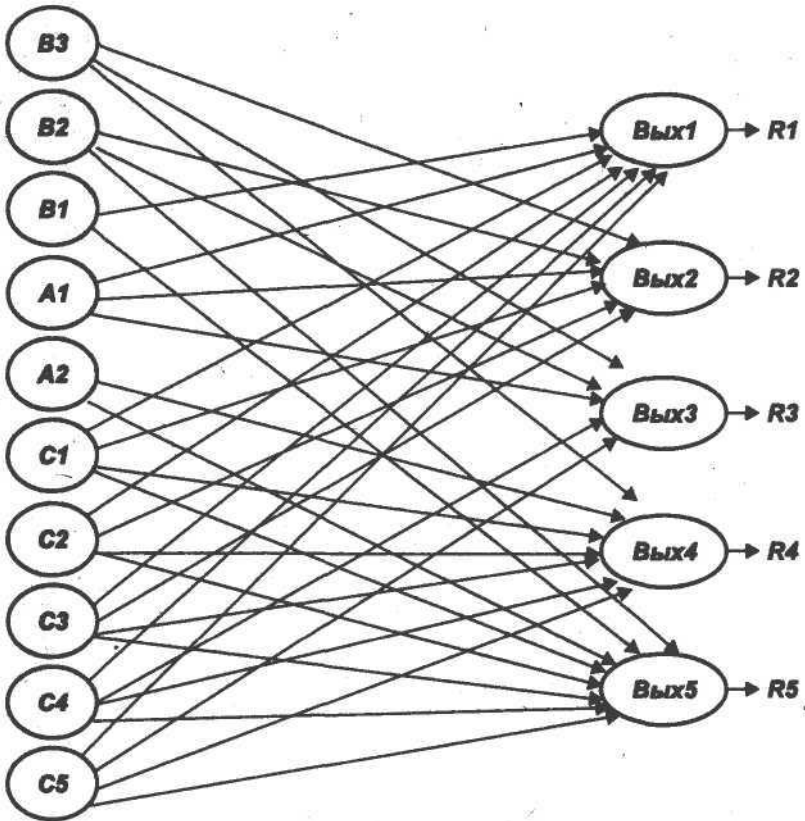


Рис. 2.7. Однослойная нейросеть

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.

Очевидно, что масштабирование порогов позволяет управлять величинами возбуждений, и при желании можно добиться аналогичных результатов расчетов по этим двум сетям. Но надо ли?

## 2.5 Формализация нейросети

Тактирование работы сети, столь характерное для каждой управляющей системы, отслеживающей ее состояние 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 возможны связи между нейронами только смежных слоев. В данном разделе мы также не рассматриваем обратные связи, но снимаем ограничение на «слоистость» нейросети, что обеспечивает 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*, анализируя 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.

Графмалопригоден для формальных исследований и компьютерных алгоритмов. Удобнее пользоваться матричным отображением нейросети. Данным способом представления можно отоб-



разить как структуру, конфигурацию, топологию графа, так и численные значения характеристик его синаптических связей.

Составим матрицу следования  $S$  (рис.2.8), число строк (и столбцов) которой равно числу нейронов сети, включая нейроны входного и выходного слоя. Каждая строка (и столбец с тем же номером) соответствует одному нейрону. Для удобства установления порядка следования нейронов диагональные элементы матрицы отмечены черным.

Элемент  $(i,j)$  этой матрицы - суть непустой объект, содержащий вес  $\omega_{ij}$  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 [9].

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 синаптические связи (впредь будем опускать слово «синаптические») 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  $j$ -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 опеределенный 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 не пустым элементам в ней. «Новые\*непустыеэлементызаменяемнанепустыеэлементы, обозначающие транзитивные связи.

В нашем примере матрица  $S$ , дополненная транзитивными связями (непустые клеточки), представлена на рис. 2.8.

Зафиксируем некоторое подмножество  $R$  нейронов входного слоя и единственный нейрон  $г$  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  $R \rightarrow г$ .

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

1. Вычеркиваем из матрицы следования  $S$  строки и столбцы, соответствующие нейронам, не принадлежащим  $R$ .
2. Вычеркиваем из матрицы следования  $S$  все строки и столбцы, отображающие нейроны выходного слоя, кроме нейрона  $г$ .
3. Вычеркиваем из матрицы следования  $S$  строки и столбцы, отображающие нулевые позиции строки нейрона  $г$ .
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 п. 4, пока не прекратится образование нулевых строк.

На рис. 2.9 матрицей  $S [В1, А1, С1 \rightarrow \mathbf{Вых1}]$  изображено множество цепочек, ведущих из нейронов  $\mathbf{В1, А1, С1}$  нейрон  $\mathbf{Вых1}$ , т.е. статический путь возбуждения  $[51, А\setminus, С1] \rightarrow \mathbf{Вых1}$ .

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  $m$*  будем называть статическую цепочку, составленную из нейронов, первые  $m-1$  которых обладают отличной от нуля величиной возбуждения.

<b>B1</b>								
<b>A1</b>								
<b>C1</b>								
<b>3</b>	0,8	0,8						
<b>5</b>			0,8					
<b>9</b>					0,8			
<b>Вых1</b>				0,8		0,8		

Рис. 2.9. Матрица следования для статического пути 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  $г$  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.

## 2.6 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 следующих тактах при последующих предъявлениях эталона импульс возбуждения увереннее одолева-

ет 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* [4, 5]: *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. Видимо, 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 лишь само возбуждение увеличивает вес связи? Зачем возбуждать дополнительные нейроны, если и этих достаточно?

Очевидно, 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 пытаясь накопить хоть какой-то опыт), мы будем увеличивать веса синаптических связей нейронов, составляющих *выделенные* цепоч-

ки 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 некоторых 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 [5].) Природа больше рассчитывает на авось, «на глазок» и другие мудрые и практические приблизительные ориентиры, мало привлекательные теоретически. В данном случае нас более обнадеживает «прилив крови» в нужном направлении, стимулируемый информационным раздражителем или легким подзатыльником. 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 приблизительному, неопределенному. Действительно, не могла эволюция, основанная на принципе «делай все, что можно», методом «проб и ошибок» привести к производству точных электронных схем, упрямых 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.

## 2.7 Применение типовых 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.

**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. 2.10. Пусть используется предложенная выше передаточная функция

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

Значения  $\omega_{ij}$  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. 2.11. Здесь отображен ее окончательный вид, так как сначала все веса связей принимаются равными нулю.

Для того чтобы сформировать решение  $R \setminus$  на нейроне  $Вых1 \setminus$ , 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 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.

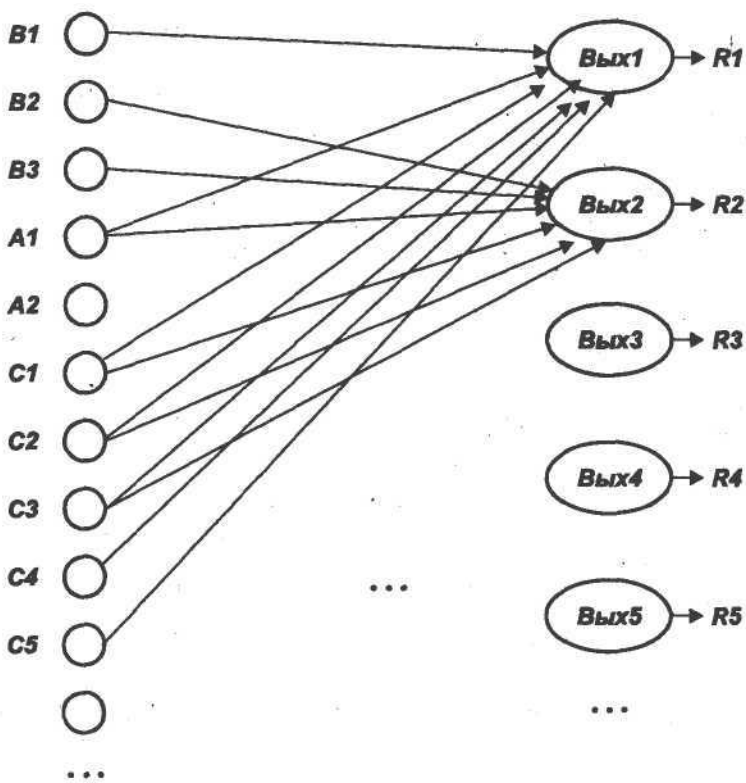


Рис. 2.10. Однослойная нейросеть после обучения

Здесь вновь прослеживается преимущество нейросети, способной элементарно 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 (см. рис. 2.11). Построенная нейросеть полностью соответствует специальной сети «под задачу», представленной на рис. 2.7.

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

**Нейросеть произвольной структуры.** Предположим, мы располагаем некоторым банком «красиво» изображенных графических



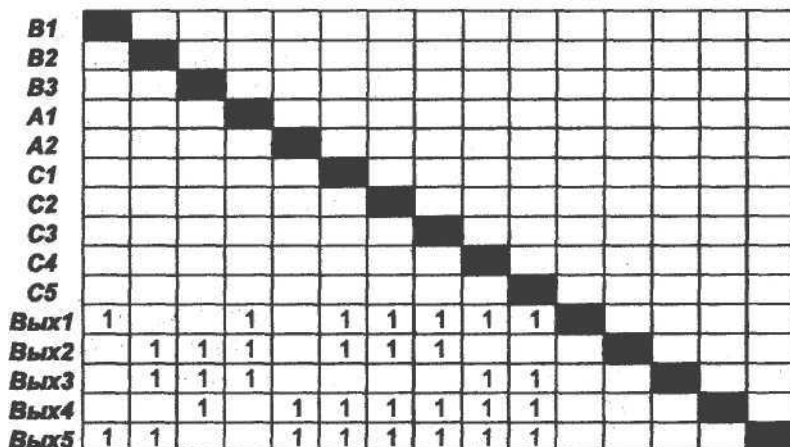


Рис. 2.11. Матрица следования для однослойной нейросети

схем, которые можно положить в основу структуры нейросети. Понравившуюся структуру интерпретируем как нейросеть, дополнив ее передаточной функцией и обучив решению задачи, поставленной дядей Рамзаем.

Пусть выбранная 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, в сети на рис. 2.6, как результат построения нейросети «под задачу».

Применим *метод опорных путей*, или *трассировки*, расширяющий использованный выше «схемотехнический» подход. Его можно изобразить схемой

рис. 2.1 → рис. 2.4 → рис. 2.6 → рис. 2.8.

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 события (0) или на его наступлении (1). После обучения сеть должна выдавать наиболее близкое решение при недостовер-

ной 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 условия 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. 2.12). На рисунке наглядно показано, какая нейросеть была предоставлена, насколько она «неповоротлива», скажем, по сравнению с однослойной «каждый с каждым».

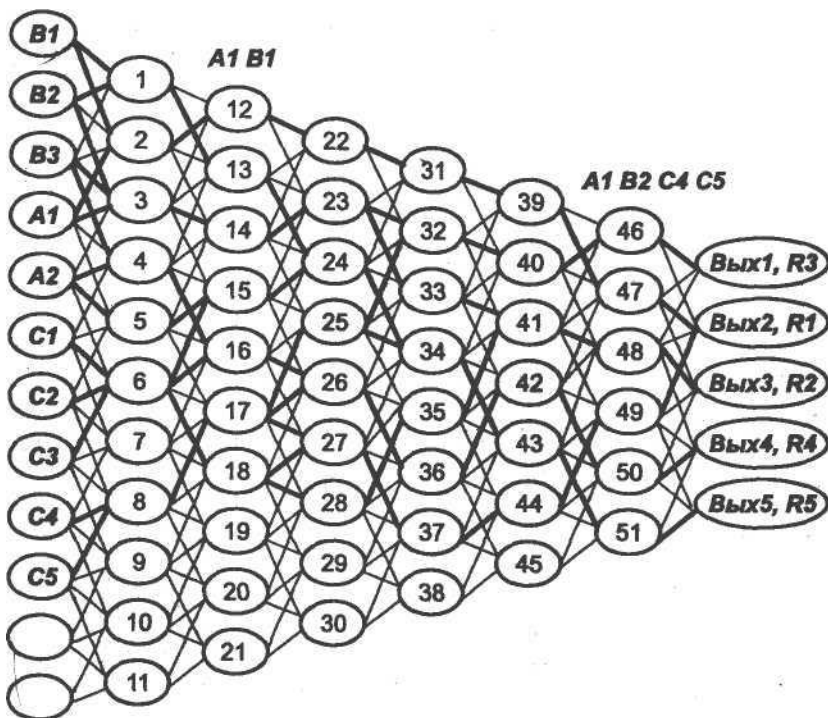


Рис. 2.12. Нейросеть после обучения

Далее рассмотрим формальный алгоритм трассировки. Однако чтобы настроиться на его понимание, необходимо проанализировать свои действия и сделать следующие выводы.

1. Мы исследовали слой за слоем, постоянно помня о цели — пять комбинаций ситуаций, каждая из которых должна возбудить один из нейронов выходного слоя. Причем решения за этими нейронами пока жестко не закреплены.

2. В каждом слое мы собирали частную комбинацию - терм, который можно использовать в последующем, — из доступных термов предыдущего слоя.

3. Термы, которые пока не используются при конструировании из-за их взаимной удаленности, мы запоминали без изменения на анализируемом слое, пытаясь «подтянуть» их в направлении возможного дальнейшего объединения.

4. Мы старались не «тянуть» термы «поперек» сети. В противном случае возникла бы проблема — 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. 2.12, если отметить элементы, соответствующие «тонким» линиям, нулевыми весами, а элементы, соответствующие «жирным» линиям, — весами, равными единице.

На рис. 2.13 отражен динамический путь возбуждения, приводящий к решению *RI*. Он строится по алгоритму, изложенному в подразд. 2.5. В данном случае динамические пути возбуждения совпадают со статическими. В общем случае из статического пути возбуждения необходимо исключить нейроны, которым соответствуют нулевые строки матрицы следования.

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»?

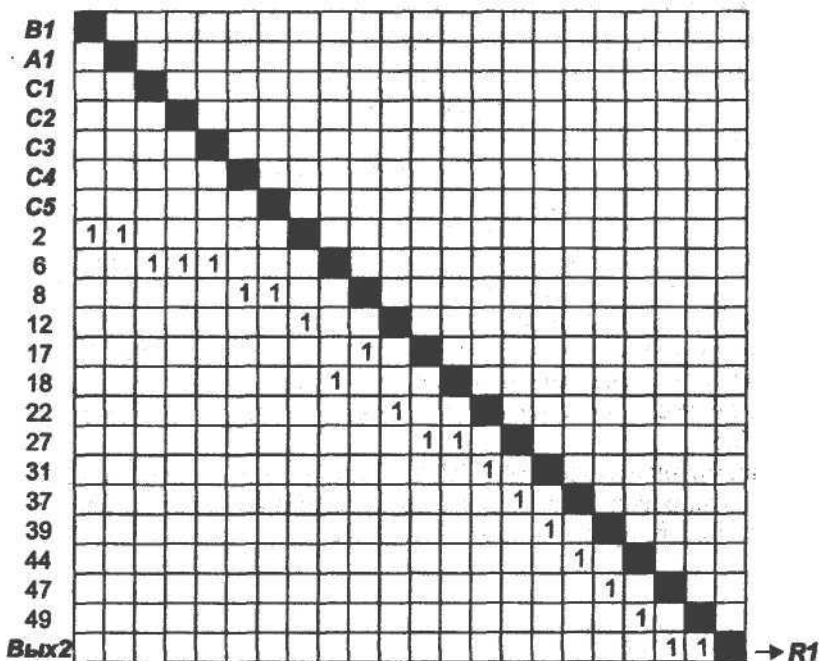


Рис. 2.13. Матрица, отображающая статический путь возбуждения

## 2.8 Энергетика нейросети

Пытаясь проанализировать, как энергия возбужденного нейрона воздействует на связанные с ним нейроны, мы вновь и вновь приходим к выводу, что она *распределяется* обратно пропорционально сопротивлениям синапсов «принимающих» нейронов.

Однако наша попытка последовать принципу распределения энергии, кроме существенного усложнения расчетов, привела к следующему.

После частичного обучения нейросети изменение весов для последующего обучения (при демонстрации других эталонов) приводит к перераспределению энергии, к ее оттоку от уже сформировавшихся динамических путей возбуждения. С точки зрения определенного нейрона выходного слоя «рассеянная» энергия

устремляется по «чужим» путям возбуждения, и тогда приходится с помощью порогов компенсировать возбуждение нужного пути, поддерживая необходимый уровень энергии.

Здесь лучше всего и проявляется известная традиция построения нейронных сетей, когда возбуждение с выхода нейрона в неизменном виде поступает на каждый вход других нейронов, не распределяясь между ними.

Ориентируясь на программную реализацию нейросети, не станем отступать от традиционных принципов, используя не «физический» подход, а информационный.

Для того чтобы сделать конкретный вывод, рассчитаем вариант той же «готовой» сети (см. рис. 2.12) с распределением энергии, увеличив веса связей, которые не приняли участие в трассировке, т.е. веса, первоначально не равные нулю. Положим их равными 0,5.

Рассмотрим эталон  $\{A1, B2, C3\}$ , требующий решения R2. Покажем, как рассчитывать сеть, предоставив возможность читателю установить правильность ее работы:

$$V_1 = 1 \cdot \frac{1}{2,5} = 0,4;$$

$$V_2 = 1 \cdot \frac{0,5}{2,5} \cdot 0,5 + 1 \cdot \frac{1}{3} = 0,43;$$

$$V_3 = 1 \cdot \frac{1}{2,5} + 1 \cdot \frac{1}{3} = 0,73;$$

$$V_{12} = V_1 \cdot \frac{0,5}{1,5} \cdot 0,5 + V_2 \cdot \frac{1}{2} + V_3 \cdot \frac{0,5}{2,5} \cdot 0,5 = 0,355;$$

$$V_{13} = V_1 \cdot \frac{1}{1,5} + V_2 \cdot \frac{0,5}{2} \cdot 0,5 + V_3 \cdot \frac{0,5}{2,5} \cdot 0,5 + V_4 \cdot \frac{0,5}{2,5} \cdot 0,5 = 0,4 \text{ и т.д.}$$

## 2.9 Рекомендации

Испытывая различные структуры нейросети и пытаясь проинвестировать обучение рассматриваемой задаче, мы, в частности, много усилий затратили на одну нейросеть. Выяснилось, что

имеет место пересечение путей возбуждения несовместимых ситуаций, требующих максимальных возбуждений разных нейронов выходного слоя. Нам так и не удалось разнести эти пути возбуждений. Коррекция весов одного пути немедленно вызвала коррекцию возбуждения другого. Кроме того, при «случайной» топологии сети величины возбуждения нейронов входного слоя оказывали различное влияние на возбуждение нейронов выходного.

В результате предпочтительнее оказалась регулярная структура сети, принудительное навязывание структуры и контроль взаимодействия ее элементов.

Другой важный вывод касается порочности излишней «экономии» при формировании нейросети.

Почему мы эвристически решали столь трудную комбинаторную задачу согласно рекомендациям, без всякого конструктивного алгоритма?

Наверное, из-за отсутствия должной избыточности. Умышленно создавая головоломку, мы теснились на маленькой сети с ограниченными связями. Таковы ли условия в живой природе, где складывающиеся связи помимо запоминания удовлетворяют требованиям устойчивости, надежности, дублирования — резервирования и хд.? Здесь уж либо жалеть и экономить, рискуя оказаться перед неразрешимостью задачи вообще, либо щедро и самым простым способом, по возможности независимо, прокладывать цепочки возбуждений от каждого нейрона входного слоя к каждому «заинтересованному» нейрону выходного слоя. Нет, ясность мышления основана на достаточности и даже избыточности мыслительной субстанции.

Снимем некоторые ограничения и рассмотрим другую типовую нейросеть. На рис. 2.14 тем же методом трассировки опорных путей показано возможное решение задачи. Здесь входной и первый внутренний слой сети связаны почти по принципу «каждый с каждым», что и позволило сформировать основные конструкции — термы, на основе комбинаций которых затем продолжается создание нужных путей возбуждения.

Все построения ведутся на основе рекурсивного подхода: сначала мы имеем термы самого низкого уровня — на входном слое, на их базе получаем конструктивные единицы первого уровня, их комбинируем в термы более высокого уровня и т.д. Конечно, это

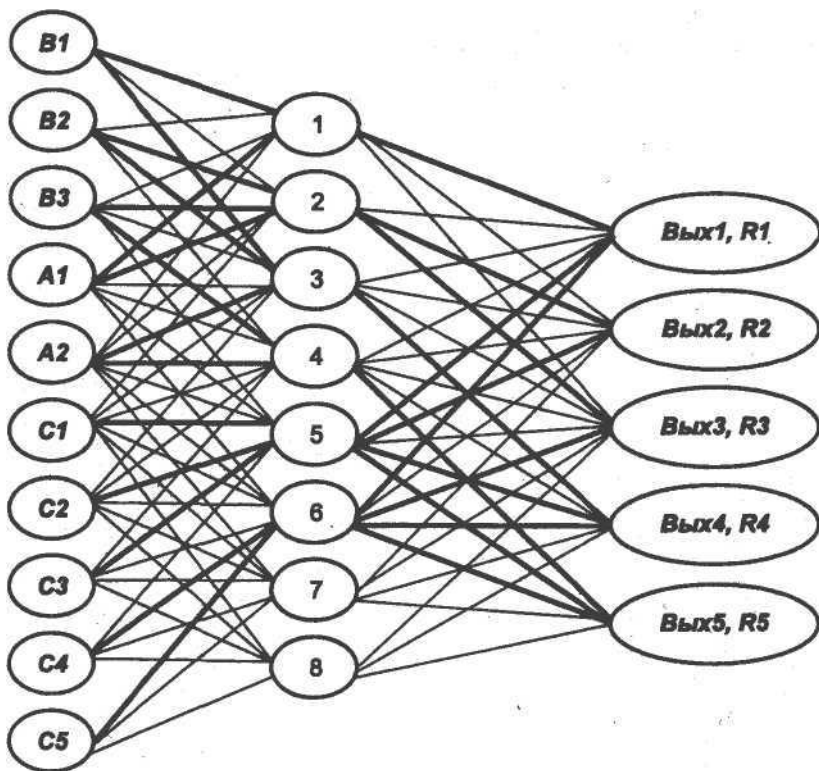


Рис. 2.14. Двухслойная нейросеть после обучения

самое конструктивное решение (мышление). Ведь термины могут формироваться в коре для многократного и многоцелевого использования, так сказать, впрок.

Но возможен и другой, более «щедрый» путь: «тянуть» цепочки «сильных» возбуждений от нейронов входного слоя к нейронам выходного слоя, избегая пересечений.

Пропагандируемая нами случайно-регулярная структура связей не должна зависеть от «слоистости» сети: связи «через слой» значительно упрощают пути возбуждения нейронов.

Вместе с тем «схемотехнический» подход и последующая замена булевых функций л и v одной передаточной функцией приводят к интересной гипотезе.

*Оптимальное число слоев нейросети или максимальная длина статической цепочки (если принцип «слоистости» не актуален) совпадает с максимальной длиной логической цепочки в аналогичной электронной схеме, реализующей систему булевых функций зависимости выходных сигналов — аналогов возбуждения нейронов выходного слоя от входных сигналов — аналогов возбуждения рецепторов.*

Сказанное позволяет ставить задачу минимизации длины статической цепочки при формировании нейросети для максимального упрощения ее структуры. Этим определяется широкая возможность применения однослойных сетей в том случае, когда задача сводится к обработке простых таблиц с однозначным соответствием принимаемых решений совокупности исходных ситуаций.

Однако дело в том, что любая булева функция представима дизъюнктивной нормальной формой, т.е. дизъюнкцией известных специалисту в области математической логики и схемотехники конъюнкций. Значит, соответствующая электронная схема состоит из логических цепочек, длина которых равна двум. При построении таких схем условно выделяют два уровня: на первом уровне формируются конъюнкции на основе комбинаций булевых переменных и их отрицаний, а на втором уровне — их дизъюнкция.

Таким образом, при использовании «схемотехнического» подхода для построения обученных нейросетей в самом общем случае мы можем довольствоваться не более чем двухслойными нейросетями.

Действительно, в нашем примере каждое событие, требующее определенного решения, интерпретируется как конъюнкция элементарных событий (например,  $A1 \wedge B1 \wedge C1$ ) в более доходчивой форме соответствует логической операции И. Комбинация событий, приводящих к одному решению, соответствует объединению дизъюнкции событий (например, к решению  $R1$  приводит дизъюнкция событий  $A1 \vee B1 \vee C1$   $\vee A1 \wedge B1 \wedge C2$   $\vee A1 \wedge B1 \wedge C3$   $\vee A1 \wedge B1 \wedge C4$   $\vee A1 \wedge B1 \wedge C5$ ), что соответствует логической операции ИЛИ.

Электронная схема на основе логических высказываний, приведенных выше, которую легко себе представить, отличается от изображенной на рис. 2.1. В ней все длины логических цепочек равны двум. Следовательно, при замене логических элементов нейронами, выполняющими пороговую передаточную



функцию (при замене булевых переменных действительными), получаем статические цепочки нейросети также с длинами, равными двум.

Значит, принципиально возможно априорное ограничение: «для построения схемы обученной нейросети для заданного применения изображаем два слоя (не считая рецепторного) нейронов, причем один из них выходной. Далее, только лишь выбором конфигурации связей, возможно, через слой, достигаем требуемой функциональной ориентации нейросети».

Неужели достаточно иметь дело с одно- и двухслойными нейросетями? Над этой проблемой советуем читателю подумать самостоятельно, обратив внимание на переход от схемы на рис. 2.1, преобразующей булевы переменные, к идентичной структуре нейронной сети на рис. 2.6, обрабатывающей действительные переменные.

## 2.10 Нейросетевые технологии и нейрокомпьютеры

Итак, располагая знаниями о данных на входе управляющей системы и реакциях на каждую ситуацию, 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 [15, 16] еще в 1970-х годах.

Для ее воплощения можно использовать 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 [17] в 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.

Получается, что мозг не может непосредственно хранить таблицы, а моделирует их с помощью нейросетевых механизмов. Исходные предложения, например, Кохонена [13] по применению нейросетей касались именно ассоциативной памяти.

Очевидно, такая реализация не очень удобна, поэтому следует определять области разумного применения каждого средства решения задач искусственного интеллекта.

Рассмотрим, где используется принцип нейросети без расчета, анализа баз знаний и т.д. По-видимому, там, где получена первичная информация для органов чувств. На этом же уровне проводим первичную классификацию и принимаем оперативное решение: убежать от стремительно приближающегося автомобиля, надеть противогаз и т.д.

Значит, в нашей жизнедеятельности существует такая ниша, где решение должно быть сверхоперативным, скорее рефлекторным, не допускающим анализа, чему и способствует высокий параллелизм сети. Именно высокий параллелизм наряду с исключением сложных расчетов обусловил взрыв интереса [7] к системам искусственного интеллекта в начале 1980-х годов, когда остро встала задача разработки вычислительных средств сверхвысокой производительности.

Здесь мы вновь затронули вопрос актуальности аппаратной реализации нейросети или нейрокомпьютеров [1, 2], так как программная модель на непараллельном компьютере лишена свойства высокого параллелизма мозга и ограничивает выход на «большие» нейросети.

Указанный 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 возбуждается, другой нет; связи нейрона индивидуальны и изменяются неидентично связям других нейронов и т.д.

Тогда при разработке параллельного вычислительного устройства - нейрокомпьютера, способного имитировать работу нейросети высокой производительности, следует учесть нижеперечисленное:

- необходимо распределять нейроны (точнее, нейроподобные элементы или соответствующие им программные процедуры) между процессорами НК, синхронизируя обработку нейронов в соответствии с матрицей следования, т.е. реализовать способ распараллеливания *по информации*;
- одинаковые для всех процессоров 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») [18 — 20], 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 требование: один нейроподобный элемент должен делить время между имитацией многих нейронов. Жесткая аппаратная имитация нейросети, соответствующая связи «один нейроподобный элемент — один нейрон», неэффективна, так как ограничивает возможную размерность моделируемой сети.

При применении сетевых технологий в рамках построения более сложных управляющих систем целесообразно использовать НК как сопроцессор под управлением мощного и универсального компьютера — монитора [21], что обусловлено разнообразными функциями «учителя» по формированию, заданию и регулировке параметров, по обучению и дальнейшему применению результатов. В рамках сегодняшних компьютерных технологий НК должен дополнять персональный компьютер (рабочую станцию) как его внешнее устройство и встраиваться в существующие ОС.

Итак, известны языки логического вывода, экспертные системы, самообучающиеся системы управления, а в природе первично воплощены лишь нейросети.

Почему же средства искусственного интеллекта, опирающиеся далеко не только на одни нейросети, не были первоначально заданы Природой, а явились плодом человеческого гения?

Да просто мозг — конструктивно целесообразен, развиваем, универсален и самодостаточен. С его помощью еще не то можно построить!

Приведем пример колеса: какова должна быть длина нерва, чувствующего обод колеса? Весьма велика. Значит, колесо должно быть отделено от живого существа. И потому человек как существо разумное колесо может сделать сам.

Обобщением «схемотехнического» подхода является подход к построению нейросетей с позиций алгебры высказываний. Логическое описание системы управления или принятия решений, в терминах алгебры высказываний, — «один в один», в результате формирования логической схемы ложится в основу нейронной сети. Остается заменить булевы переменные действительными (достоверная или другие оценки), а операции конъюнкции и дизъюнкции — единой операцией, осуществляемой передаточной функцией нейрона. Конечно, такая замена не столь безболезненна. Необходимы несложные преобразования логического описания, а также настройка порогов и весов связей, чтобы избежать неоднозначности выдаваемых решений.

Кроме того, читатель не должен считать, что каждая нейросеть сводится к однослойной. Значит, предложенный далее алгоритм трассировки нуждается в обобщении, направленном на структуризацию обучаемой нейросети.

# 3 ТРАССИРОВКА НЕЙРОСЕТИ

## 3.1 Подход: 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  $\&$ , объединяющей все посылки, ведущие к одному выводу,  $A \setminus \& B \setminus \& C1 \setminus \& C2 \setminus \& C3 \setminus \& C4 \setminus \& C5 \rightarrow R1$ .

Выберем нейросеть, отражающую все особенности, необходимые для построения алгоритма трассировки, и возьмем тот же пример, который выглядит теперь следующим образом:

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

Задавая обобщенный эталон на входе, например, из тех, что приведены выше, необходимо наиболее рациональным образом построить пересекающиеся пути возбуждений, ведущие от каждого возбужденного входного нейрона к соответствующему нейрону выходного слоя. Пути должны быть максимально объединены, совмещены согласно стратегии слияния и общего выхода.

Ори этом разрабатывается эвристический алгоритм, с минимальным использованием ресурсов сети — ее нейронов и связей, а также с минимальным развитием, если связей недостаточно.

Пусть выбрана нейросеть (рис. 3.1), где первоначально заданные связи обозначены тонкими стрелками. Строим соответствующий

ющую матрицу следования  $S$  и в ней транзитивные связи (рис. 3.2). Проверяем, все ли нейроны выходного слоя достижимы из любого нейрона входного слоя? Если нет, считаем, что сеть составлена некорректно.

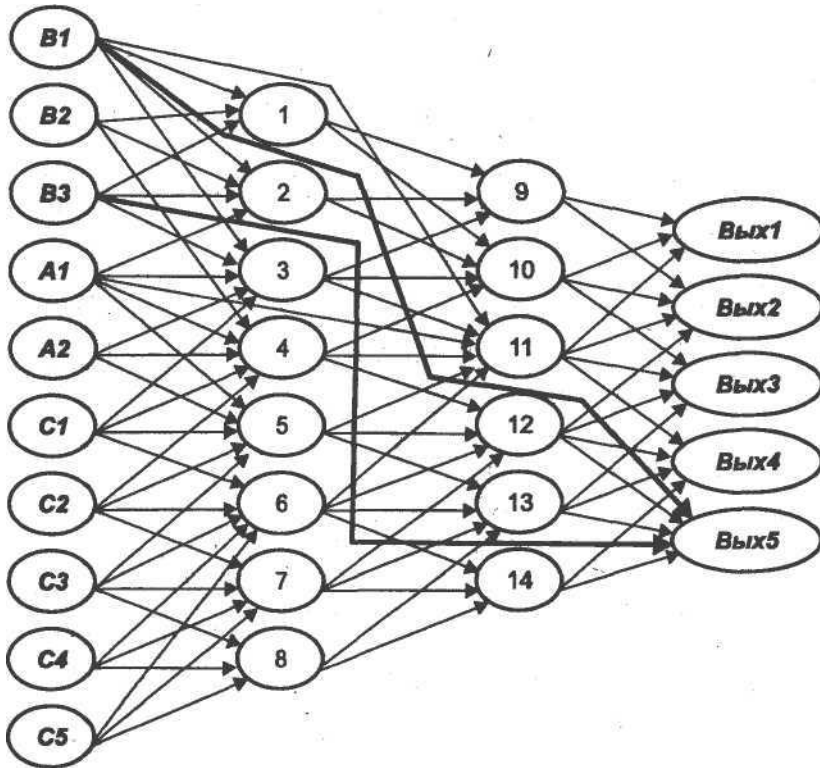


Рис. 3.1. Нейросеть, подлежащая обучению

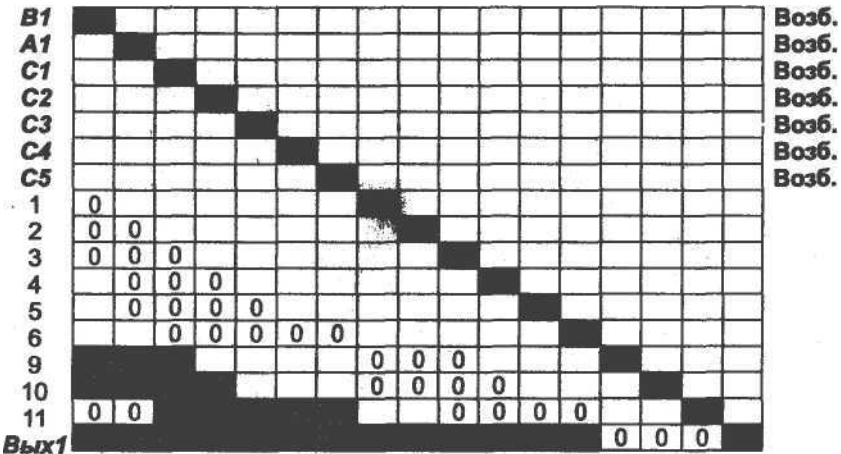
Из нейрона  $B \setminus$  не исходит ни одной статической цепочки, заканчивающейся нейроном  $Вых5$ , в связи с тем, что в строке, соответствующей нейрону  $Вых5$ , нет даже транзитивной связи в столбце, соответствующем нейрону  $B \setminus$ . Это относится и к нейрону  $B3$ .

Введем непосредственные синапсические связи  $B1 \rightarrow Вых1$  и  $B3 \rightarrow Вых5$ , что отмечено на рис. 3.1 жирной стрелкой, а на рис. 3.2 — весами элемента матрицы.

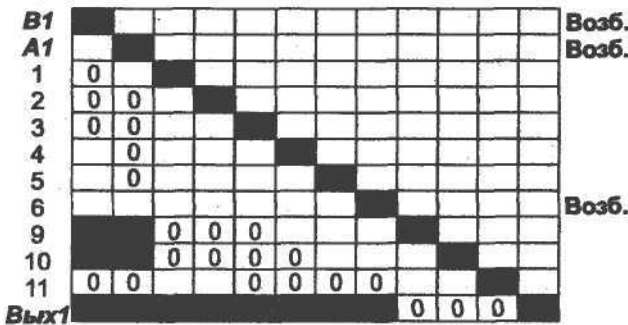


Сформируем (см. подразд. 2.5) статический путь возбуждения  $[B1, A1, C1, C3, C4, C5] \rightarrow B_{вых}$  и отобразим матрицей  $S_1[B1, A, C, C2, C3, C4, C5 \rightarrow B_{вых}]$  (рис. 3.3, а). Имитируя прокладываемые пути возбуждений, первоначально присваиваем нейронам входного слоя, отображенным нулевыми строками этой матрицы, признак «возбужден».

Найдем входы матрицы  $S_1$  (пустые строки), обладающие признаком «возбужден». Выделим в одну совокупность соответствующие им столбцы (в данном случае 1 — 7) и найдем в ней строку, содержащую максимальное число нулей (строка, соответствующая нейрону б). Заменяем в этой строке (в выделенной совокупности



а



б



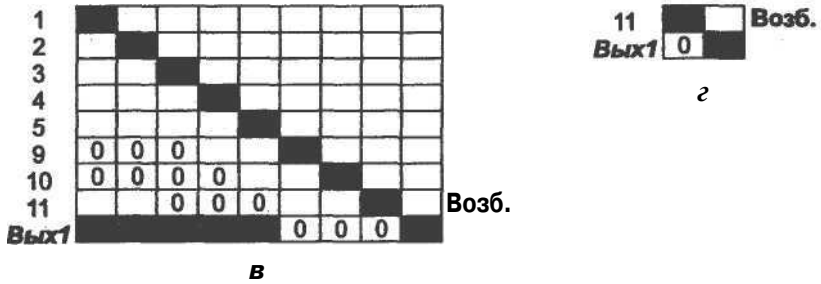


Рис. 3.3. Матрица следования при обучении первому эталону:  
*a* — общий вид; *b* — первый шаг преобразования матрицы следования;  
*в* — после исключения нейронов  $B1, A1$ , и  $б$ ;  
*г* — после исключения «невозбужденных» входов

столбцов) нули единицами, т.е. максимально увеличиваем соответствующие веса. Присвоим нейрону  $б$  признак «возбужден», а также значение  $m_б = 5$ , характеризующее то количество нейронов, «возбуждение» которых интегрируется на входе нейрона. При повторном использовании этого нейрона менять веса связей на его входе нельзя. Изменение весов отображаем в матрице  $S$ .

Исключим из матрицы  $S_1$  строки и столбцы, соответствующие нейронам  $C1, C2, C3, C4, C5$ . Новая матрица  $S_1$  имеет вид, представленный на рис. 3.3, *б*.

Вновь выделим столбцы, соответствующие «возбужденным» входам матрицы, т.е. нейронам  $B \setminus A \setminus б$ . Находим строку в выделенной совокупности столбцов с максимальным числом нулей. Строка соответствует нейрону 11. Заменяем в этой строке нули единицами, т.е. максимально увеличиваем веса. Присваиваем нейрону 11 признак «возбужден» и значение  $m_{11} = 3$ . Отражаем изменение весов в матрице  $S$ .

Исключим из матрицы  $S_1$  строки и столбцы, соответствующие нейронам  $B \setminus A \setminus б$ . Матрица принимает вид, изображенный на рис. 3.3, *в*. Исключаем из матрицы все входы, соответствующие нейронам 1, 2, 3, 4, 5, не обладающие признаком «возбужден», так как строящийся нами путь возбуждения их уже миновал. Исключение этих нейронов породит новые нейроны 9 и 10, подлежащие исключению по той же причине.

Последовательное исключение сверху вниз динамически учитывает появление таких нейронов и гарантирует полное исклю-

чение за один проход. Матрица  $S_1$  примет вид, представленный на рис. 3.3, г.

Теперь в первом столбце имеется единственная строка с нулем, соответствующая нейрону  $Вых\setminus$ . Изменим нуль на единицу, окончательно получим возможный динамический путь возбуждения по заданному эталону. Изменения отразим в матрице  $S$ .

Не будем приводить новый вид матрицы следования  $S$ , а на изображении самой сети (рис. 3.4) выделим построенную трассу (опорный или динамический путь возбуждения по предполагаемому эталону).

В результате имеем сеть, обученную реакции на эталон одной комбинации событий.

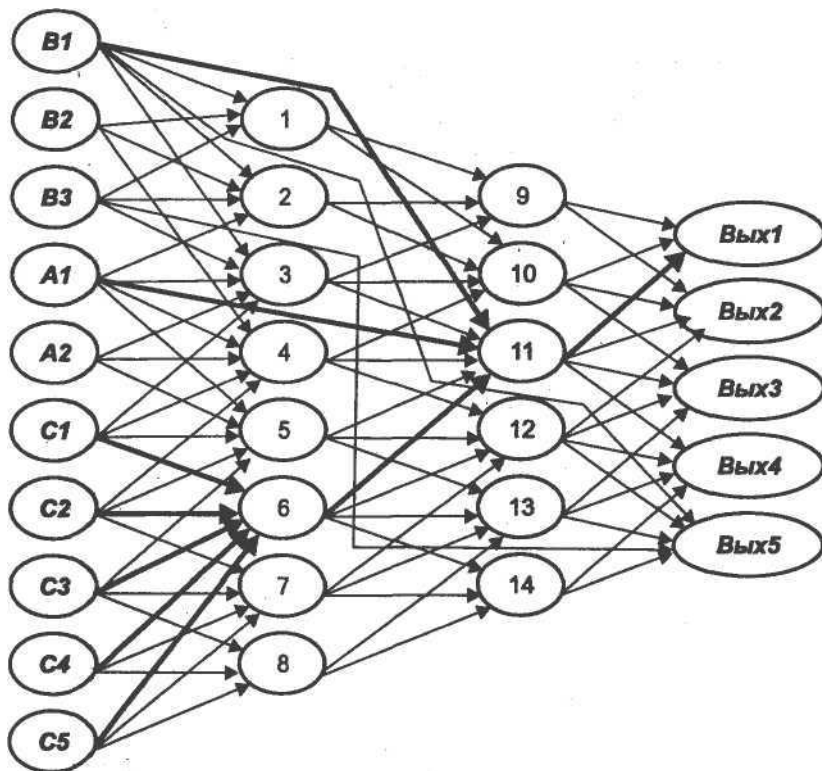


Рис. 3.4. Нейросеть, обученная первому эталону

Составим обобщенный эталон ситуации  $\{A\setminus, B2, 53, C\setminus, C2, C3\}$ , требующий решения  $R2$ . Свяжем это решение с нейроном  $Вых2$ .

По сети, представленной на рис. 3.4, или по матрице (см. рис. 3.2), построим с учетом частичного обучения матрицу статического пути возбуждения  $S_2 [B2, B3, A1, C1, C2, C3 \rightarrow Вых2]$  (рис. 3.5).

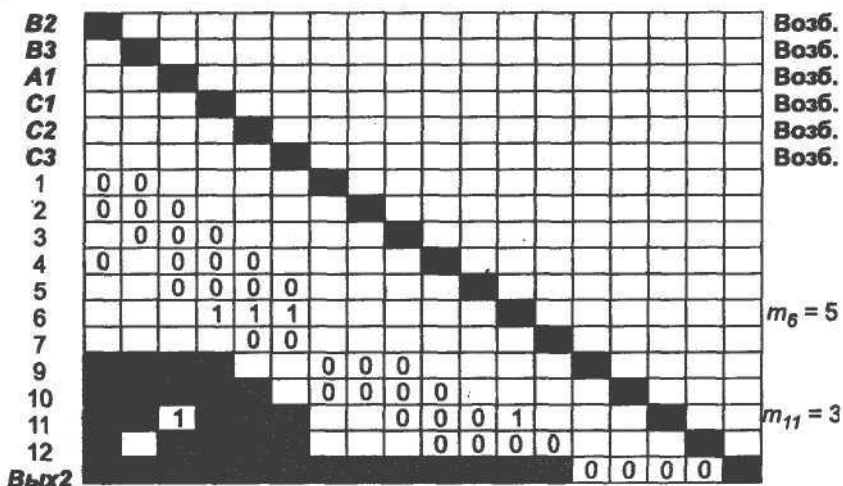


Рис. 3.5. Матрица следования для обучения второму эталону

На нейроне 6 ранее суммировалось возбуждение пяти нейронов:  $C1, C2, C3, C4, C5$ . Таким образом, и соответствующая строка матрицы  $S$  содержит пять единиц  $m_6 = 5$ , а соответствующая строка матрицы  $S_2$  содержит всего три единицы. Значит, этот нейрон не может использоваться для трассировки необходимого пути возбуждения, так как комбинация  $C1 \& C2 \& C3 \& C4 \& C5$  здесь не требуется. То же относится и к нейрону 11.

Исключим из данной матрицы строки и столбцы, соответствующие нейронам 6 и 11. Матрица примет вид, приведенный на рис. 3.6.

Объединим столбцы множества входов матрицы, обладающих признаком «возбужден», в данном случае первые шесть столбцов. Найдем в них первую строку, содержащую максимальное число нулей. Это строка, соответствующая нейрону 4.

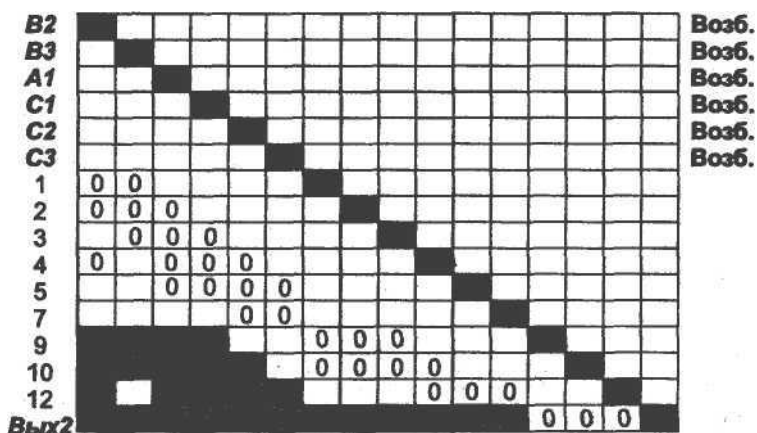


Рис. 3.6. Шаг преобразования матрицы следования

Изменим в ней все нули на единицы, положим  $m_4 = 4$ . Присвоим нейрону 4 признак «возбужден», отразим внесенные изменения весов в матрице  $S$ .

Исключим из матрицы  $S_2$  строки и столбцы, соответствующие нейронам  $B2, A1, C1, C2$ , определив возбуждение нейрона 4.

Вновь объединим столбцы множества входов, обладающих признаком «возбужден», соответствующие нейронам  $B3, C3, 4$ . В совокупности столбцов выберем первую строку, содержащую максимальное число нулей, — строку, соответствующую нейрону 1. Заменим нуль на единицу, положим  $m_1 = 1$ , нейрону 1 присвоим признак «возбужден».

Исключим из матрицы строку и столбец, соответствующие нейрону  $B3$ , передавшему энергию возбуждения нейрону 1, а также исключим строки и столбцы, соответствующие образовавшимся входам — «не возбужденным» нейронам 2 и 3.

В столбцах, соответствующих нейронам — входам с признаком «возбужден», найдем строку с наибольшим числом нулей. Такая строка соответствует нейрону 10. Заменим нули на единицы, присвоим нейрону признак «возбужден», полагаем  $m_{10} = 2$ .

В матрице  $S_2$  исключим строки и столбцы, соответствующие нейронам 1 и 4, передавшим возбуждение нейрону 10.

Нейрон 9, не обладающий признаком «возбужден», образует вход матрицы. Исключим соответствующие ему строку и столбец.

Выделим столбцы, соответствующие нейронам  $C3$  и  $10$ , и в них находим строку с максимальным числом нулей. Такая строка соответствует нейрону  $5$ . Заменяем в ней нуль единицей, полагаем  $m_5 = 1$  и присваиваем нейрону признак «возбужден». Отразим изменение веса в матрице  $S$ .

Исключим из матрицы  $S_2$  строку и столбец, соответствующие нейрону  $CB$ , передавшему возбуждение нейрону  $5$ . Далее исключим строку и столбец, соответствующие нейрону  $7$ , как порождающему вход матрицы, который не имеет признака «возбужден».

В совокупности выделенных столбцов, соответствующих нейронам  $5$  и  $10$ , строка, соответствующая нейрону  $12$ , имеет единственный нуль. Заменяем его на единицу, присвоим нейрону  $12$  признак «возбужден», положим  $m_{12} = 1$ . Отразим внесенное изменение веса в матрице  $S$ . Исключим из матрицы  $S_2$  строку и столбец, соответствующие нейрону  $5$ . И наконец, на последнем шаге, заменив нули в строке, соответствующей нейрону  $Вых2$ , единицами, окончательно получим искомый путь возбуждения (рис. 3.7).

Обучим сеть ситуации, требующей решения  $R3$ . Пусть этому решению соответствует нейрон  $Вых3$ . Матрица  $S_3[B2, B3, A1, C4, C5 \rightarrow Вых3]$  представлена на рис. 3.8, *a*. Исключим из матрицы  $S_3$  те нейроны, для которых значение  $m$  превышает количество единиц в соответствующей строке, т. е. — нейроны  $4, 5, 6, 11$ , которые в данном пути возбуждения не используются. Вновь исключим из матрицы  $S_3$  те нейроны, для которых в результате предыдущего исключения значение  $m$  превышает количество единиц в соответствующей строке. Таким образом действуем до исчерпания этой возможности. В результате будут исключены из рассмотрения (вследствие исключения нейронов  $4$  и  $5$ ) нейроны  $10$  и  $12$ . Окончательный вид матрицы  $S_3$  изображен на рис. 3.8, *b*.

Присвоим всем нейронам, образующим входы матрицы, признак «возбужден». Выделим и объединим столбцы, соответствующие входам матрицы  $S_3$ .

Однако вследствие ранее выполненных построений — в результате частичного обучения — появляются новые особенности. Поэтому расширим последующие действия.

В выделенных столбцах найдем строку, содержащую максимальное число единиц. Строка представляет тот нейрон, кото-

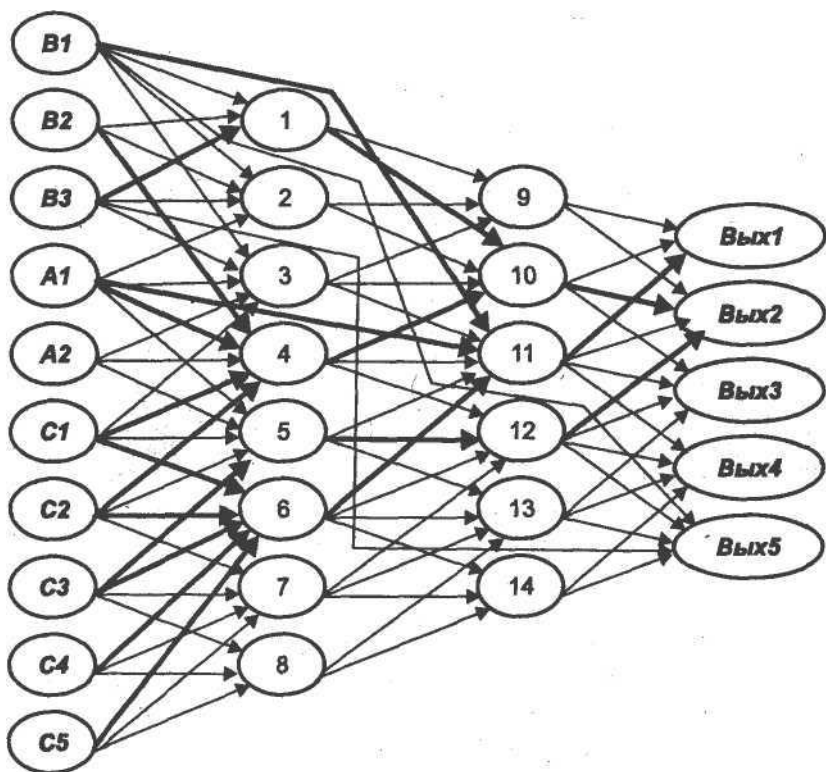
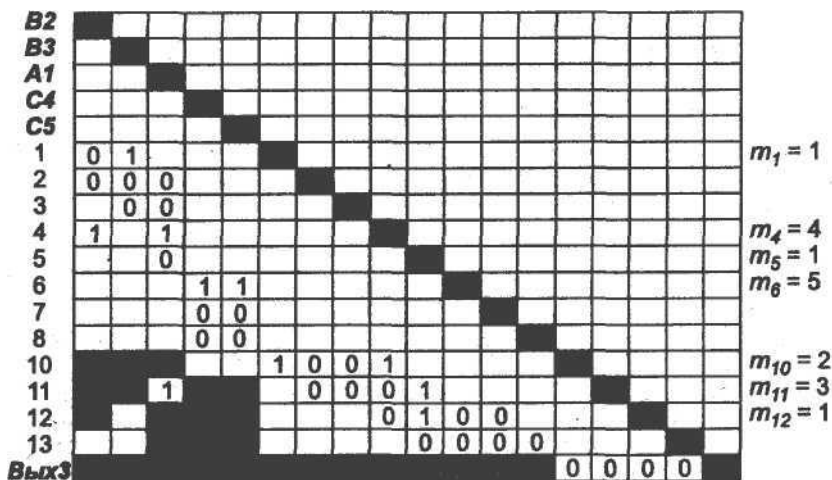


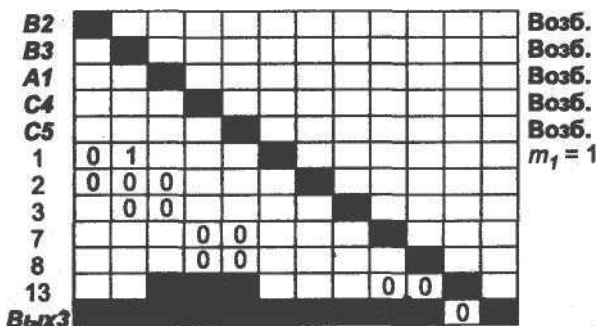
Рис. 3.7. Нейросеть после обучения двум эталонам

рый приобрел энергию возбуждения одного или нескольких нейронов, образующих входы матрицы. Эти нейроны могут быть исключены из рассмотрения. Так, нейрон 1 возбужден единственным нейроном, принадлежащим текущему множеству входов матрицы, — нейроном B3, который может быть исключен из рассмотрения. Матрица  $S_3$  принимает вид, представленный на рис. 3.8, в. Подтверждаем у нейрона 1 признак «возбужден» и превращаем его во вход матрицы  $S_3$ , уничтожив веса в его строке.

Вновь найдем множество входов матрицы и выделим соответствующие столбцы. Отыщем строку, содержащую единицы в этих и только этих столбцах. Таковых больше нет. Тогда найдем стро-



а



б

Рис. 3.8. Шаги преобразования матрицы следования для обучения третьему эталону:  
 а — общий вид; б — матрица следования после исключения неиспользованных нейронов.

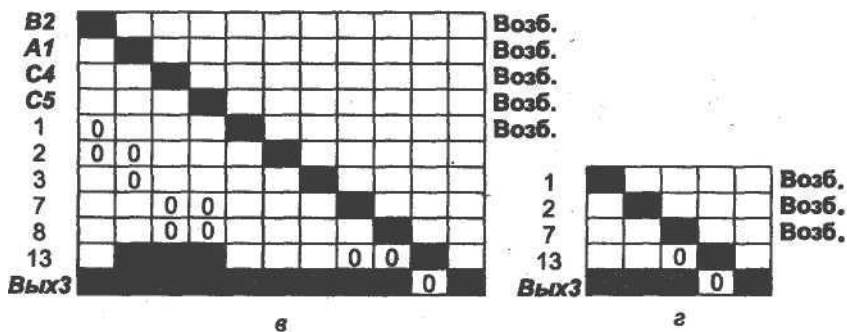


Рис. 3.8. Продолжение  
 а — после исключения нейрона В3; з — после возбуждения нейрона 7

ку, содержащую максимальное число нулей. Первая такая строка соответствует нейрону 2. Заменяем нули единицами, положим  $m_2 = 2$ , исключим нейроны В2 и А1 из рассмотрения, присвоим нейрону 2 признак «возбужден». Отразим сделанные изменения весов в матрице S.

Исключим входы, не обладающие признаком «возбужден», — нейроны 3 и 5.

Выделим столбцы, соответствующие входам. В них нет строк, содержащих единицы. Тогда найдем строку, содержащую максимальное число нулей. Такая строка соответствует нейрону 7. Заменяем нули (в выделенной совокупности столбцов) на единицы, положим  $m_7 = 2$ , присвоим нейрону 7 признак «возбужден», отразим изменение весов в матрице S. Исключим нейроны С4 и С5 из рассмотрения. Сформировался «невозбужденный» вход, соответствующий нейрону 8. Исключим то рассмотрение и этот нейрон (рис. 3.8, з).

Выделим столбцы, соответствующие «возбужденным» входам и не находим строк, содержащих единицы. Тогда найдем первую строку с максимальным числом нулей. Эта строка, соответствующая нейрону 13. Заменяем нуль на единицу, нейрону 13 присвоим признак «возбужден», полагаем  $m_{13} = 1$ . Исключим нейрон 7 из рассмотрения.

Выделим столбцы, соответствующие множеству «возбужденных» входов. В них нет строки с единицами, но единственная строка с максимальным числом нулей соответствует нейрону



*Вых3*. Заменяем нуль на единицу. Однако статус этого нейрона особый, и значение  $m$ , как и признак «возбужден», ему не присваивается.

Исключим нейрон 13 из рассмотрения.

Выделим множество столбцов, соответствующих «возбужденным» входам, т. е. нейронам 1 и 2. Ищем в этих столбцах строку с единицами, затем с максимальным числом нулей, но таковых нет. Следовательно, необходимо ввести дополнительные связи. Для этого в каждом столбце найдем первую сверху транзитивную связь и заменим ее на единицу. Тогда данная сеть дополнится динамическими цепочками возбуждения  $1 \rightarrow \text{Вых3}$  и  $2 \rightarrow \text{Вых3}$ .

Построение трассы решения  $R3$  закончено, и сеть имеет вид, представленный на рис. 3.9.

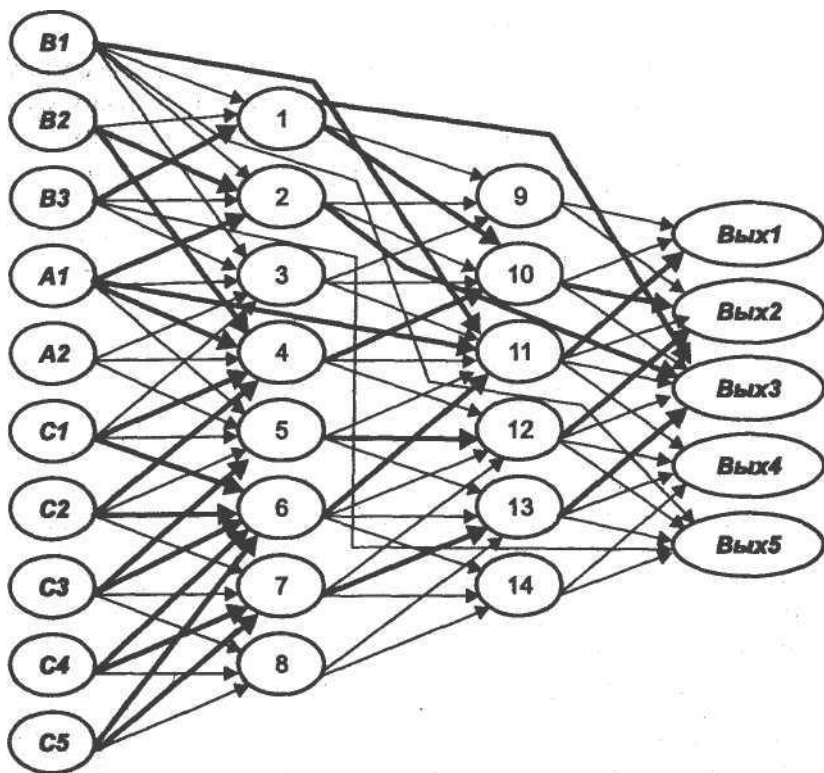
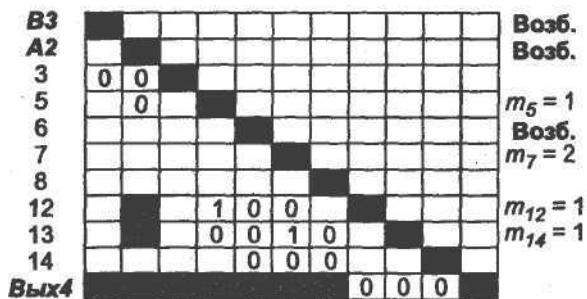
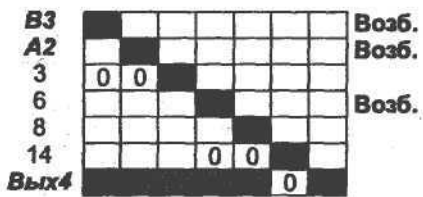


Рис. 3.9. Нейросеть после обучения третьему эталону

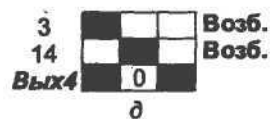




а



б



д



е

Рис. 3.10. Матрица следования для обучения четвертому эталону:  
 а - общий вид; б - после исключения нейронов 4 и 11; в - после возбуждения нейрона 6; г - после исключения неиспользуемых нейронов;  
 д - после возбуждения нейронов 3 и 14;  
 е - после исключения нейрона 14

Присвоим всем нейронам-входам признак «возбужден». Выделим столбцы, соответствующие входам, это первые семь столбцов. Найдем строку с максимальным числом единиц в этих столбцах при отсутствии единиц в других. Строка соответствует нейрону 6. Теперь нейрон 6 представляет нейроны С\, С2, С3, С4, С5, передавшие ему энергию возбуждения. Присвоим ему признак «возбужден», исключим из рассмотрения перечисленные нейроны, матрица  $S_4$  примет вид, представленный на рис. 3.10, в.

Для «невозбужденных» нейронов исключим в матрице  $S_4$  строки (и столбцы), число единиц в которых меньше  $m$ . Они соответствуют нейронам 5 и 7. Повторим этот шаг до полного исключения таких нейронов - исключаются нейроны 12 и 13. Матрица примет вид, представленный на рис. 3.10, г.

Исключим из рассмотрения множество «невозбужденных» входов. К таковым относится нейрон 8. Выделим множество столбцов, соответствующих входам матрицы. В их совокупности найдем первую строку с максимальным числом нулей (единиц нет во всей матрице). Такую строку образует нейрон 3. Нули в ней соответствуют нейронам  $B\bar{B}$  и  $A2$ . Исключим эти нейроны из рассмотрения, положим  $m_3 = 2$ , присвоим нейрону 3 признак «возбужден».

Вновь выделим множество столбцов, соответствующих входам матрицы, и так как в их совокупности нет строк, содержащих единицы, найдем строку с максимальным количеством нулей, соответствующую нейрону 14. Заменим в ней ноль на единицу, положим  $m_{14} = 1$ , присвоим нейрону 14 признак «возбужден». Исключим нейрон 6 из матрицы, и она примет вид, представленный на рис. 3.10,  $\delta$ .

Выделим множество столбцов, соответствующих входам матрицы, и так как в их совокупности нет строк, содержащих единицы, найдем строку с максимальным количеством нулей. Эта строка соответствует нейрону  $Вых4$ . Заменим в ней ноль на единицу. Поскольку это нейрон выходного слоя, не станем присваивать ему признак «возбужден» и значение  $m$ . Исключим нейрон 14 из рассмотрения. Вид матрицы представлен на рис. 3.10,  $e$ .

В этой матрице множество «возбужденных» входов представлено единственным нейроном 3. Однако в соответствующем ему столбце нет ни единичных, ни нулевых элементов. Введем в сеть дополнительную связь, найдя в этом столбце первую сверху транзитивную связь и положив ее вес равным единице. Эта связь порождает динамический путь возбуждения  $3 \rightarrow Вых4$ . Все изменения отразим в матрице  $S$ .

Построение трассы решения  $R4$  закончено. На рис. 3.11 приведен вид нейросети, полученной после данного этапа трассировки.

Теперь приступим к трассировке последнего пути возбуждения  $A2 \& B1 \& B2 \& C1 \& C2 \& C3 \& C4 \& C5 \rightarrow R5$ . Решению  $R5$  соответствует нейрон выходного слоя  $Вых5$ .

На основе текущей матрицы  $S$  построим матрицу  $S_5[B1, B2, A2, C1, C2, C3, C4, C5 \rightarrow Вых5]$  путем исключения «ненужных» строк и столбцов с сохранением транзитивных связей. Исключим из рассмотрения нейрон 4, для которого количество единиц в строке меньше  $m_4 = 4$ . Получившаяся матрица  $S_5$  представлена на рис. 3.12,  $a$ .

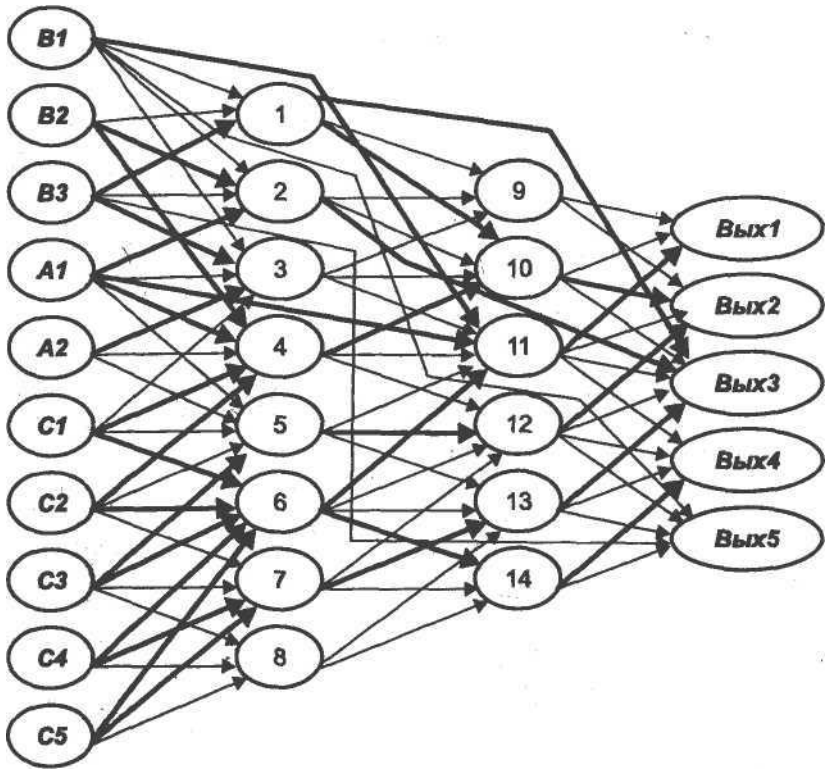


Рис. 3.11. Нейросеть после обучения четырем эталонам

Присвоим всем входам признак «возбужден». Выделим столбцы, соответствующие «возбужденным» входам. В совокупности этих столбцов найдем строку, содержащую максимальное число единиц в этих и только этих столбцах. В данном случае такая строка соответствует нейрону 6. Присвоим ему признак «возбужден» и исключим из рассмотрения нейроны  $C1, \dots, C5$ .

Далее исключим из рассмотрения нейроны 5 и 7, так как в соответствующих строках отсутствуют единицы при отличных от нуля значениях  $m$ , после чего придется исключить по той же причине нейроны 12 и 13. Получившаяся матрица представлена на рис. 3.12, б.

Исключим из рассмотрения «невозбужденный» вход, соответствующий нейрону 8. Выделим столбцы, соответствующие «воз-



нейрону выходного слоя  $Вых5$ , содержащую максимальное число нулей (строк с единицами больше нет). Заменяем нули на единицы. Исключим нейроны  $B1$  и  $14$  из рассмотрения. Отразим введенные изменения в матрице  $S$ . На рис. 3.12,  $z$  приведена полученная матрица  $S_z$ .

Выделим множество столбцов — входов матрицы. В этих столбцах не находим строк, содержащих единицы, а также строк, содержащих нули. Тогда в каждый столбец занесем единицы на места транзитивных связей, т. е. введем связи  $B2 \rightarrow Вых5$  и  $2 \rightarrow Вых5$ , а веса этих связей положим равными единице.

Трассировка нейросети закончена. Окончательный ее вид представлен на рис. 3.13, где единичные веса соответствуют жирным стрелкам.

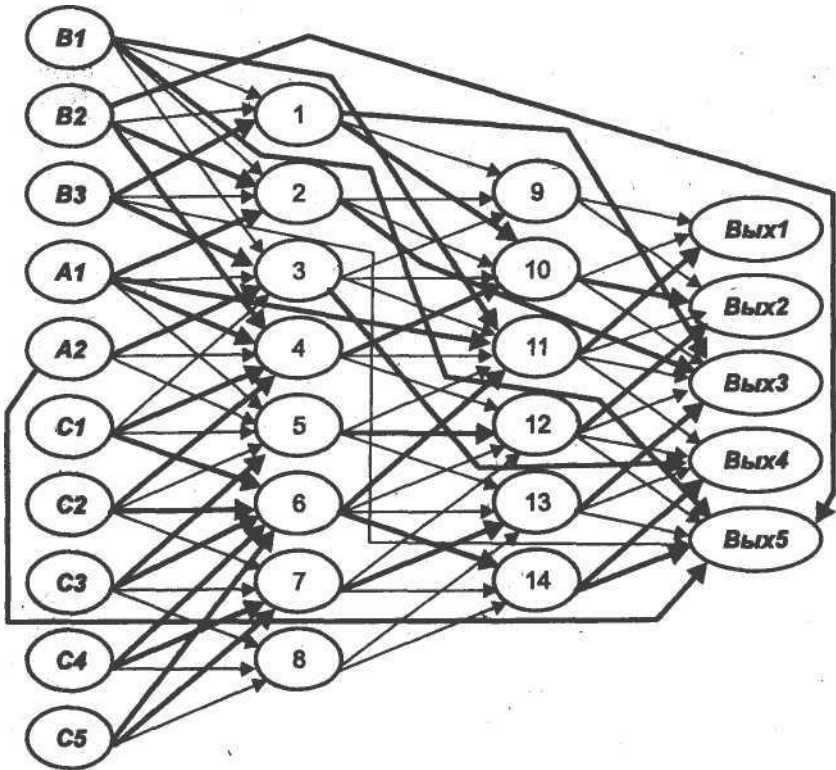


Рис. 3.13. Нейросеть после обучения всем эталонам

## 3.2 Алгоритм трассировки нейросети

В результате решения задачи найден *алгоритм трассировки нейросети*. Рассмотрим его по шагам.

Шаг 1. Дополняем матрицу следования  $S$  транзитивными связями по алгоритму, представленному в подразд. 2.4.

Шаг 2. Строки, соответствующие нейронам выходного слоя, не должны иметь пустых элементов в позициях, соответствующих нейронам входного слоя. Пустой элемент указывает на отсутствие пути возбуждения от нейрона входного слоя. В таком случае будем считать нейросеть построенной некорректно и нуждающейся во внесении дополнительных связей, например, непосредственно от нейрона входного слоя к нейрону выходного.

Шаг 3. Организуем перебор всех эталонов обобщенных ситуаций (обобщенных эталонов), закрепив нейроны выходного слоя  $Выxi$  ( $i = 1, \dots, m$ ) за соответствующими решениями. Для каждого обобщенного эталона выполняем шаги 4—15. Если все обобщенные эталоны обработаны, выполнение алгоритма заканчиваем.

Шаг 4. Для обобщенного эталона  $i$  ( $i = 1, 2, \dots, m$ ) строим матрицу следования  $S_i[D_{1i}, D_{2i}, \dots, D_{ir} \rightarrow Выxi]$ , где  $D_{1i}, D_{2i}, \dots, D_{ir}$  - нейроны входного слоя — рецепторы, возбуждающиеся при подаче обобщенного эталона, т.е. характеризующие ситуацию. Величина возбуждения равна единице.

Шаг 5. В матрице  $S_i$  последовательно, сверху вниз, вычеркиваем строки (и соответствующие столбцы), число единичных элементов в которых меньше указанного ранее при строке  $m$ .

*Примечание.* Отличные от нуля значения счетчиков  $m$  формируются не ранее чем после обучения первому эталону. Поскольку в процессе вычеркивания возможно порождение новых строк с числом единиц, меньше  $m$ , а матрица  $S$  треугольная, то следует производить это вычеркивание за один проход построчно сверху вниз. В противном случае не избежать повторного, а возможно, и многократного анализа появления новых строк, подлежащих вычеркиванию.

Шаг 6. Присваиваем признак «возбужден» всем нейронам входного слоя матрицы  $S_i$ .



Шаг 7. Проверяем количество строк матрицы  $S_r$ . При наличии более одной строки выполняем следующий шаг, в противном случае — шаг 3.

Шаг 8. Исключаем строки (и соответствующие им столбцы) матрицы  $S_r$ , являющиеся ее входами и отображающие нейроны входного слоя, не обладающие признаком «возбужден».

Шаг 9. Исключаем из матрицы  $S_r$  строки (и столбцы), которые содержат количество единичных элементов меньше параметра  $m$ , указанного при строке.

*Примечание.* Такое действие, уже предусмотренное шагом 5, необходимо производить после каждого вычеркивания строк и столбцов.

Шаг 10. Выделяем множество строк матрицы  $S_r$  с признаком "возбужден" и множество соответствующих им столбцов.

Шаг 11. В совокупности выделенных столбцов находим первую строку, содержащую максимальное число единиц и не содержащую единиц в других столбцах, т. е. число единиц в найденной строке и выделенных столбцах не должно быть меньше  $m$ . Соответствующий этой строке нейрон может быть использован повторно. Если таковой строки найти не удастся, выполняем шаг 13.

Шаг 12. Исключаем из рассмотрения нейроны (вычеркиваем строки и столбцы), которым соответствуют единицы в найденной строке. Присваиваем нейрону, соответствующему выделенной строке, признак «возбужден». Уничтожаем в этой строке все нули и символы транзитивных связей, если они имеются. Строка преобразуется во вход матрицы  $S_r$ . Далее выполняем шаг 7.

Шаг 13. В совокупности выделенных столбцов находим первую строку, содержащую максимальное число нулевых элементов. Если такой строки нет, выполняем шаг 15.

Шаг 14. Меняем значения возбуждения соответствующих связей, т.е. заменяем нули единицами. Присваиваем нейрону, соответствующему выделенной строке, значение  $m$ , равное количеству единиц в строке, и признак «возбужден». Вычеркиванием строк и столбцов исключаем из рассмотрения нейроны, передавшие энергию возбуждения найденному.

*Примечание.* Значения весов связей одного нейрона могут корректироваться лишь однажды. При обучении другим эталоном нейрон может использоваться повторно, если в обучении участвуют все нейроны, передающие ему возбуждение с весом, равным единице. При этом достаточно учитывать лишь число единиц в строке. В процессе такого обучения эталоны друг другу не мешают.

Внесенные изменения весов учитываем в матрице  $S$ . Выполняем шаг 7.

Шаг 15. В каждом выделенном столбце находим первый сверху непустой элемент, соответствующий транзитивной связи. Вводим в нейросеть дополнительную связь, присваивая найденному элементу единичное значение. Исключаем из рассмотрения (вычеркиваем строки и столбцы) нейроны, соответствующие обработанным столбцам. Отражаем внесенные изменения в матрице  $S$ .

*Примечание.* Указанная транзитивная связь (см. выше) требуется на последней стадии обучения обобщенному эталону и может обнаружиться лишь в строке, соответствующей нейрону выходного слоя, поэтому формировать значение  $m$  уже излишне. Выполняем шаг 7.

Построенный алгоритм трассировки, несомненно, эвристический, т.е. 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 решений.

Так, в нашем примере удачно сложился терм в результате связи  $[C1, C2, C3, C4, C5 \rightarrow 6]$ . Он использовался при получении трех решений по эталонным ситуациям —  $R1, R4, R5$ . По-видимому, целесообразны термы в результате объединения  $C \setminus, C2, C\bar{B}$ , а также  $C4$  и  $C5, B1$  и  $B2$  и др.

Предлагаем читателю самостоятельно провести трассировку сети, представленной на рис. 2.14, по предложенному алгоритму.

### 3.3 Приведение нейросети после трассировки

Рассмотрим простейшую передаточную функцию

$$\begin{aligned} V &:= \sum_j \omega_j V_j; \\ V &:= \text{if } V > h \text{ then } V \text{ else } 0. \end{aligned} \tag{3.1}$$

Задавая конкретные эталоны в исследованном выше примере, будем наблюдать за изменением величины возбуждения соответствующего нейрона выходного слоя. Мы видим, что каждый

эталон приводит к одинаковой величине возбуждения нейрона выходного слоя.

Однако, например, при распознавании знаков алфавита величины возбуждения нейронов выходного слоя в ответ на предъявление различных эталонов могут быть различными. Так, эталон латинской / «засветит» на входном слое (на сетчатке) меньшее число нейронов, чем, скажем, буква  $Q$ . При рассмотренной передаточной функции величины возбуждения на выходе будут существенно разными.

Возникает вопрос: не приводит ли некоторое незначительное изменение (зашумление) ситуации  $A$ , логично требующее решения  $R_A$ , к ситуации  $B$ , т.е. к принятию решения  $R_B$ ?

Разная величина возбуждения нейронов выходного слоя указывает на возникновение нежелательной приоритетности ситуаций или эталонов. Очевидно, это возбуждение следует как-то выровнять по эталонным ситуациям, *привести* к одной или достаточно близкой величине.

Введем постоянную величину  $U$ , например, как максимальное значение возбуждения некоторого нейрона выходного слоя или среднюю оценку, не превышающую такой максимум. Введем для каждого нейрона выходного слоя *коэффициент приведения*:

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

Значение  $V_j$  находим не по обобщенному эталону, а по каждой конкретной эталонной 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) находим величину возбуждения  $V_{Вых1}$  нейрона  $Вых1$ :

$$k_{Вых1} = \frac{U}{V_{Вых1}}.$$

В дальнейшем в режиме распознавания будем использовать произведение каждого значения возбуждения нейрона выходного слоя на его коэффициент приведения. Таким образом, мы поставим все эталонные ситуации на входе в равное положение.

### 3.4 Трассировка двухслойной нейросети

В подразд. 2.9 указывалось, что любой предикат алгебры логики может быть записан в виде дизъюнктивной нормальной формы как дизъюнкция конъюнкций значений предиката и утверждений, принимающих значения ИСТИНА — ЛОЖЬ, а также их отрицаний. Поэтому будем считать, что каждый предикат системы, на основе которого строится нейронная сеть для принятия решений, представляет собой дизъюнкцию конъюнкций событий, приводящих к общему решению. Например, система предикатов (2.1) в записи на Алголе имеет вид:

```
if (A1∧B1∧C1)∨(A1∧B1∧C2)∨(A1∧B1∧C3)∨(A1∧B1∧C4)∨  
(A1∧B1∧C5) then R1;  
if (A1∧B2∧C1)∨(A1∧B2∧C2)∨(A1∧B2∧C3)∨  
(A1∧B3∧C1)∨(A1∧B3∧C2)∨(A1∧B3∧C3) then R2;      (3.2)  
if (A1∧B2∧C4)∨(A1∧B2∧C5)∨(A1∧B3∧C4)∨  
(A1∧B3∧C5) then R3;  
if (A2∧B3∧C1)∨(A2∧B3∧C2)∨(A2∧B3∧C3)∨(A2∧B3∧C4)∨  
(A2∧B3∧C5) then R4;  
if (A2∧B1∧C1)∨(A2∧B1∧C2)∨(A2∧B1∧C3)∨(A2∧B1∧C4)∨  
(A2∧B1∧C5)∨(A2∧B2∧C1)∨(A2∧B2∧C2)∨(A2∧B2∧C3)∨  
(A2∧B2∧C4)∨(A2∧B2∧C5) then R5.
```

Тогда трассировка нейросети заданной структуры по каждому предикату производится в два этапа. На первом этапе формируются термы — конъюнкции событий, за которыми закрепляются нейроны. Эти нейроны «собирают» возбуждение нейронов входного слоя. На втором этапе исключаются из рассмотрения нейроны входного слоя, на основе которых сформированы термы. Входным слоем нейронов становятся нейроны входного слоя сети, соответствующие событиям, не вошедшим в состав термов, а также нейроны, реализующие термы. Формируется связь этих нейронов с нейроном выходного слоя, закрепленным за решением.

Как ранее говорилось, для выполнения такой трассировки достаточно взять двухслойную сеть. Однако в такой сети перво-

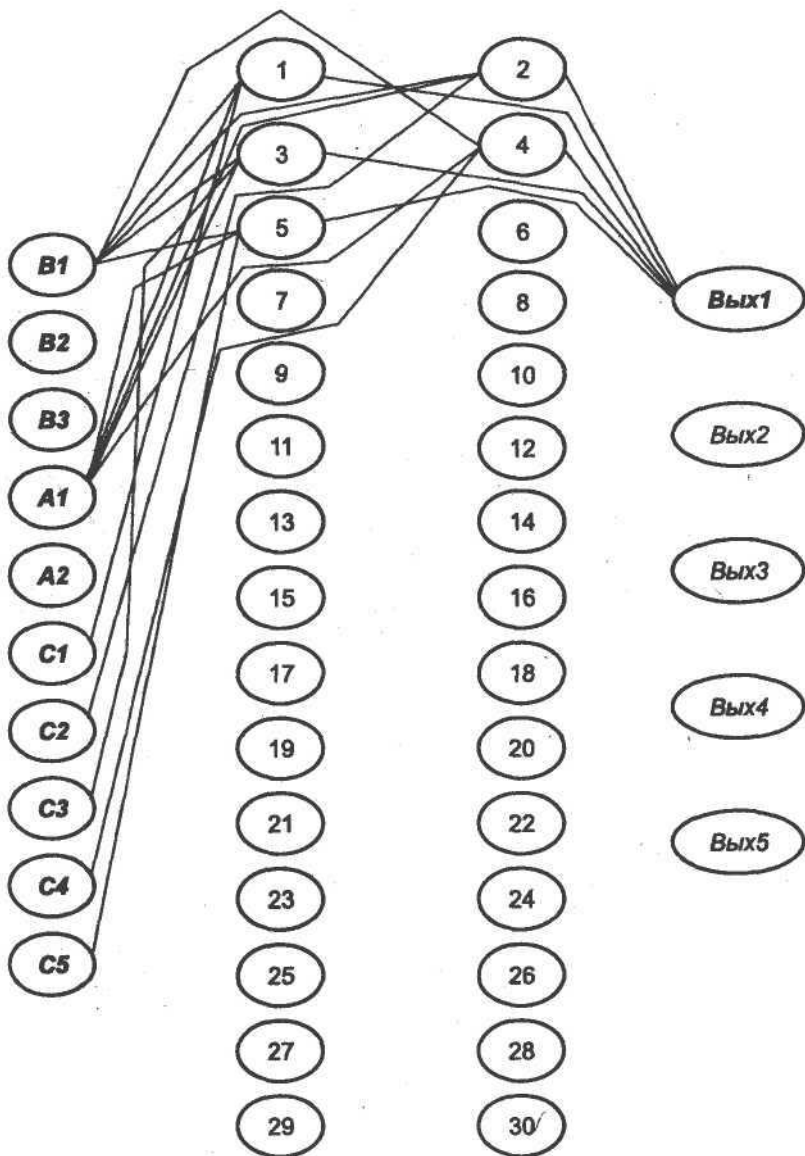


Рис. 3.14. Двухслойная нейросеть после обучения (показана трассировка одного решения)

начально может быть недостаточно заданных связей, поэтому целесообразно дополнить сеть необходимыми связями. Ведь из сказанного выше следует, что могут понадобиться связи «через слой», т.е. возможен отход от традиционной «слоистости» сети.

На рис. 3.14 показана трассировка двухслойной сети, реализующей систему предикатов (3.2). Связи между слоями первоначально заданы по принципу «каждый с каждым». Выделены связи с весами, равными единице.

В дополнение к сказанному отметим следующее.

Если ситуация, приводящая к некоторому решению, поглощается обобщенным эталоном, приводящим к другому решению, то нейросеть не сводится к однослойной и дает неоднозначный ответ (см. подразд. 2.9). При этом правильное логическое описание системы принятия решений состоит из описания логических функций, каждая из которых представляет собой скобочную запись с единственным входением переменных. Трассировка должна производиться в соответствии с вложенностью скобок.

Таким образом, необходимо обобщение предлагаемого здесь алгоритма трассировки.

# 4 СТРАТЕГИИ ОБУЧЕНИЯ И САМООБУЧЕНИЯ

## 4.1 Динамизм обучения

На самом деле обучение не бывает внезапным, как мы это представили выше, рассматривая сразу обобщенные эталоны.

Можно учить по «чистым» эталонам, объединив их в обобщенные, но реально используются «частные» эталоны, приводящие к одному решению. Например, последовательно применяют эталоны  $B1\&A1\&C2$ ,  $B1\&A1\&C1$  и т.д. и получают решение  $R1$ . Можно предъявлять попеременно разные эталоны, формируя одновременно разные решения.

Например, дядя Рамзай не мог ранее предвидеть некоторых комбинаций событий, предварительно имея ошибочное суждение о распределении работ своих клиентов. Ему даже пришлось употребить свои связи для того, чтобы осуществить вожделенную мечту: включить в орбиту своих действий восхитительную высокоприбыльную продукцию фирмы *Ночная бабочка*.

Во всех случаях на этапе обучения целесообразно определять события  $A$ ,  $B$ ,  $C$  с максимальной достоверностью, равной единице, рассчитывая тем самым реакцию нейросети на вполне определенные ситуации. Ведь даже малая вероятность события, приводящая в рабочем режиме к малой величине возбуждения нейрона входного слоя, указывает на то, что событие возможно и, следовательно, нуждается в рассмотрении и выработке решения. Технология трассировки — «прокладывания» опорных путей в сети также предполагает максимальную достоверность событий.

Сеть, очевидно, должна эволюционировать - пополняться и развиваться. Лишь на этой основе можно в дальнейшем ставить вопрос не только об обучении с учителем, но и о самообучении.

, Таким образом, мы не сразу предъявляем сети, например, весь обобщенный эталон  $A1\&B1\&C1\&C2\&C3\&C4\&C5$ , приводящий к решению  $R1$ , а последовательно используем частные эталоны как единичные значения соответствующих булевых переменных или целых. Возможно, мы не все эталоны даже можем

перебрать практически. Например, спустя год после успешной деятельности компании, Петя впервые направился к Марине, торгующей тибетским бальзамом. Анализ этой ситуации потребовал трассировки решения  $A2 \& B3 \& C3 \rightarrow R4$ .

Конечно, проще всего 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 по всему множеству обобщенных эталонов, включая уточненные. Однако это противоречит динамике совокупного процесса обучения и распознавания, непрерывному участию сети в системе управления, увеличению трудоемкости обучения. Обучение должно быть столь же оперативным и динамичным, как и 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. В результате использованные ранее нейроны выводятся в ресурс. Затем трассировка выполняется заново по уточненному обобщенному эталону, что гораздо короче, если не учитывать временных затрат на ликвидацию следов присутствия устаревшего обобщенного эталона.

Рассмотрим проблему динамического включения новых частных эталонов в состав обобщенного на фоне уже произведенного частичного обучения нейросети.

Пусть предъявление эталонов  $A1 \& B1 \& C2$  и  $A1 \& B1 \& C4$  (обобщенный эталон  $A1 \& B1 \& C2 \& C4$ ), требующих решения  $R1$ , а также предъявление 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, представленной на рис. 4.1. Здесь выделенные связи обладают максимальным значением веса (единичным). Нейросеть получена с помощью алгоритма трассировки, изложенного выше. Матрица следования  $S$ , соответствующая получившейся сети, показана на рис. 4.2.

Указанную нейросеть весьма утяжеляют дополнительные связи. Обучение всем обобщенным эталонам сразу (см. разд. 3) вызывает термы, использующиеся при получении различных решений. Здесь же термы не складывались, нейроны почти не использовались повторно. Это и привело к формированию большого числа дополнительных связей.



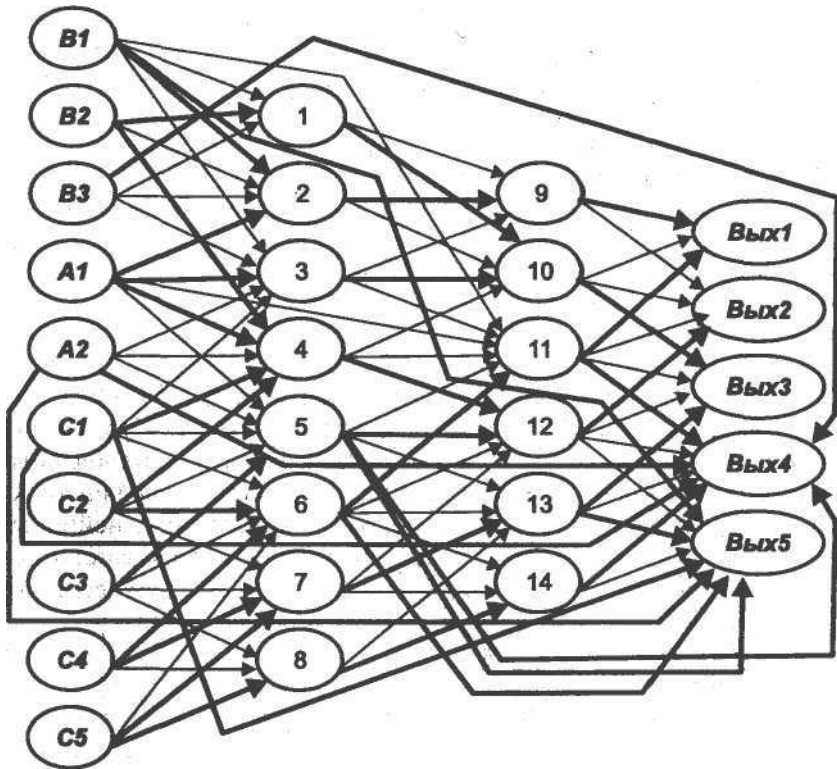


Рис. 4.1. Частично обученная нейронная сеть

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  $A1 \& B1 \& C2 \& C4$  следует расширить до эталона  $A1 \& B1 \& C1 \& C2 \& C5 \& C4$ , требующего решения  $R1$ , т. е. необходимо построить путь возбуждения (трассу)  $B \setminus, A \setminus, C \setminus, C2, C3, C4 \rightarrow \text{Вых}1$ . Причем при построении нельзя изменять уже ранее построенный путь  $B \setminus, A \setminus, C2, C4 \rightarrow \text{Вых}1$ . Значит, надо проложить трассу, дополнив сеть путем возбуждения  $C1, C3 \rightarrow \text{Вых}1$ , при этом максимально объединяя, пересекая, сливая его с уже построенным ранее путем возбуждения, ведущим к  $\text{Вых}1$ .

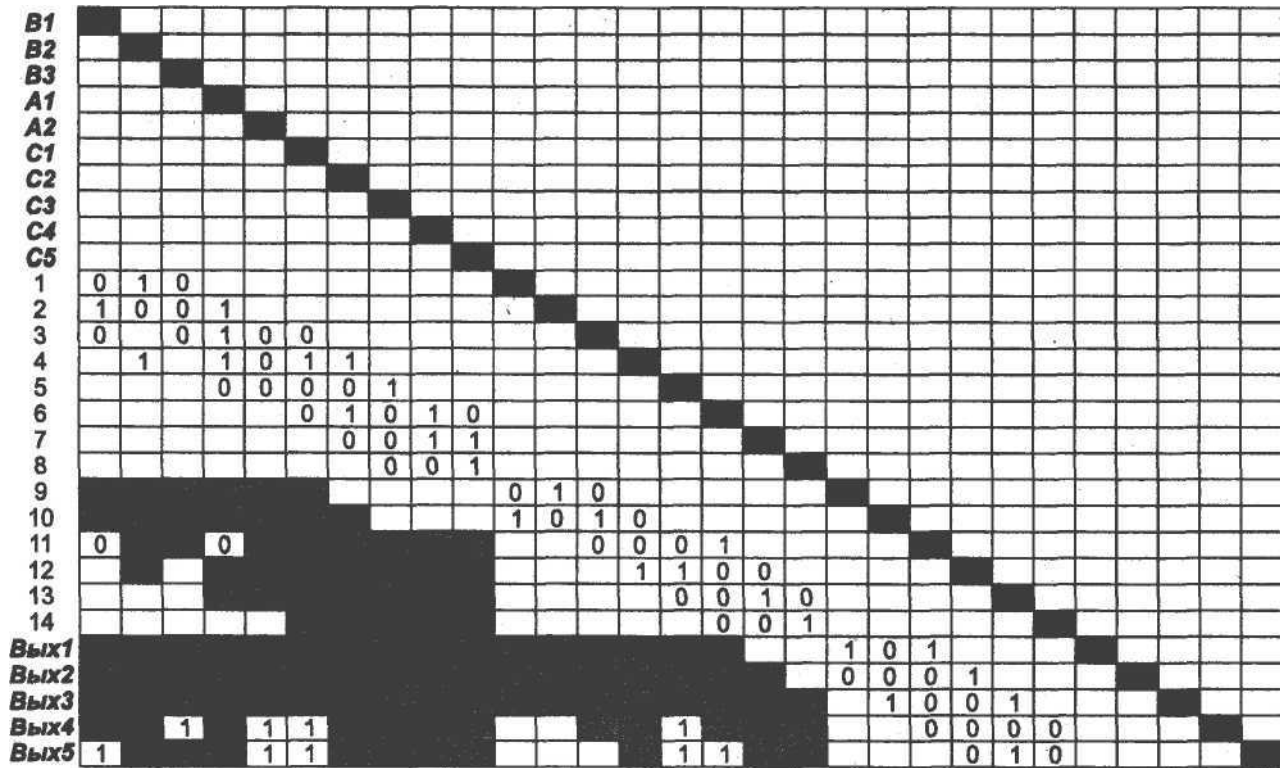


Рис. 4.2. Матрица следования для частично обученной нейронной сети

Нельзя, однако, допускать слияния, влияния строящейся трассы на пути возбуждения, ведущие к другим нейронам выходного слоя, что происходит при повторном использовании нейронов. Ведь слияние двух путей возбуждения заключается в увеличении весов синаптических связей некоторых нейронов. Таким образом, некоторые нейроны могут получать сигналы возбуждения большего, чем прежде, числа нейронов. А если эти нейроны использовались для получения других решений?

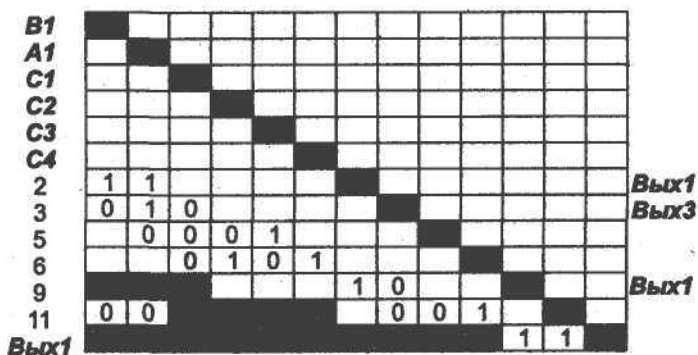
В дополнение к алгоритму трассировки заметим следующее. Целесообразно при повторном использовании нейронов в процессе последовательного анализа обобщенных эталонов проследить за тем, участвует ли данный нейрон лишь в одном пути возбуждения к единственному нейрону выходного слоя и к какому именно или к более чем одному нейрону выходного слоя. Этого вполне достаточно.

Для нового обобщенного эталона на основе матрицы  $S$  построим матрицу  $S^*$  [ $B1, A1, C1, C2, C3, C4 \rightarrow Vых1$ ] (рис. 4.3, а). В соответствии с правилом построения в ней представлены лишь те нейроны, для которых число единиц в строке равно соответствующему значению  $m$ . Для каждой строки, содержащей единичные элементы, указано использование нейрона для получения какого-либо единственного решения (признак  $Vых1$ ) или не единственного (признак отсутствует).

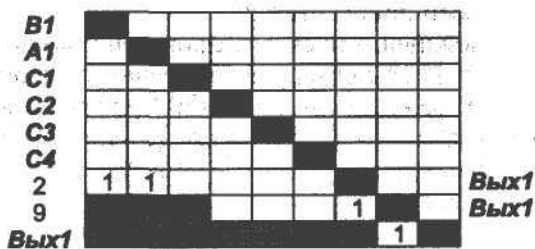
Исключим из построенной матрицы строки и столбцы, соответствующие нейронам, участвующим в путях возбуждения, не ведущих к  $Vых1$  (рис. 4.3, б).

В столбце, соответствующем нейрону  $C1$ , и в строке, не являющейся входом, находим первый сверху нулевой элемент. Полагаем его равным единице и отражаем введенное изменение в матрице  $S$ . Если нулевого элемента в столбце не нашлось, находим первый «пустой» или отмеченный транзитивной связью элемент и полагаем его равным единице. В результате получаем дополнительную связь  $C1 \rightarrow 2$  с единичным весом, отраженную в матрице  $S$ . То же проделываем с нейроном  $C3$ , введя дополнительную связь с единичным весом  $C3 \rightarrow 2$  (рис. 4.3, в).

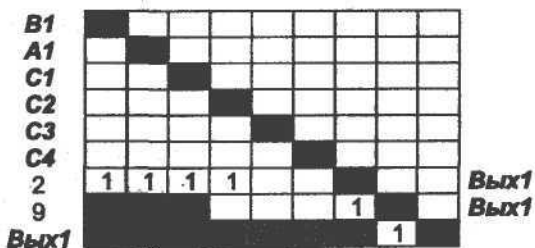
Отметим, что наше решение основано на эвристике, и место появления новых возбужденных связей может быть оспорено. Например, могли быть построены связи  $C1 \rightarrow 2, C3 \rightarrow 9$ , или  $C1 \rightarrow Vых1, C3 \rightarrow 9$ , или  $C1 \rightarrow 9, C3 \rightarrow Vых1$ , и т.д.



а



б



в

Рис. 4.3. Матрица следования для обучения уточненному эталону:  
 а — общий вид; б — после исключения лишних нейронов;  
 в — после введения дополнительных связей

Предлагаем читателю самостоятельно получить уточненную сеть, дополнив рис. 4.1.

## 4.2 Не задавайте глупых вопросов!

Мы учим сеть по обобщенным эталонам, несомненно, опираясь на здравомыслие пользователя. Так, применяя обобщенный эталон  $A1 \& B2 \& B3 \& C1 \& C2 \& C3$ , мы предполагаем возникновение логичных вопросов типа «Какое решение принять, если Вася отправился в павильон "Роксана", куда накануне завезена большая партия косметики из Китая?» При  $A1 = B3 = C2 = 1$  (на самом деле истинное происхождение косметики известно) задаем ясный вопрос и получаем столь же ясный ответ: «Принимай решение (или значение твоей прибыли)  $R2$ !»

Но что, если при сложившихся связях сформулировать вопрос:  $B3 = C1 = C2 = C3 = 1$  ? Максимального и равного возбуждения на выходе достигнет даже не один нейрон! Судить ли по неоднозначности ответа о том, что на входе произошло «объединение» вопросов? (Каков вопрос, таков ответ?) А всегда ли получается неоднозначный ответ?

Положим  $A2 = B1 = B2 = 1$ . 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 обусловлена теми задачами и правилами, которые лежат в основе ее разработки. Если мы предположили, что события  $A$ ,  $B$ , С обязательно должны участвовать в формировании вопросов, то при появлении 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 трассировки пути возбуждения  $B3, C1, C2, C3 \rightarrow \text{Вых6}$ .

Однако возможности неоднозначного ответа (получения одинаковой величины возбуждения нейронов выходного слоя) таким образом не избежать. Придется анализировать всю картину возбуждения нейронов выходного слоя и выдавать все ответы типа: «Продукция фирмы  $B3$  поступила в палатки  $C1, C2, C3$ »; "Это впоследствии скорее всего приведет к решениям  $B2$  и  $R4$ ".

Нетрудно видеть, что такая ситуация приводит к необходимости надстройки сети вторым *логическим уровнем*, где выходной слой нейронов первого уровня становится (возможно, в совокупности со своим входным слоем) входным слоем второго уровня. Так могут быть продолжены логические цепочки наших умозаключений.

## 4.3 Познание нового - основа самообучения

Мы научили сеть множеству эталонов — вплоть до сформированных в обобщенные. Теперь, подавая на вход даже искаженные образы, можно ответить на вопрос «На какой эталон в большей мере он похож?» С помощью максимально возбужденного нейрона выходного слоя ответ будет найден. Рассмотрим далее более внимательно пороги.

Что если установить ограничение на величину возбуждения нейронов выходного слоя? 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 необходимо найти свободный нейрон выходного слоя и по вышеприведенному алгоритму произвести трассировку.

Так проводится запоминание эталона. Для последующего использования новых знаний трудно обойтись без внешнего воздействия, без учителя, ибо неизбежен вопрос «Что *это*, и что из *этого* следует?» Ведь длителен путь эволюции, основанный на самообучении, без поддержки извне. Таким образом, решение, принимаемое при предъявлении нового эталона, выбирается на основе расчетов, моделирования, опыта, цели, экспертных оценок и т.д.

Представим себе возможный диалог между нейрокомпьютером (НК) и учителем (У):

*НК(Величина возбуждения ни одного нейрона выходного слоя не преодолела порог):* Это что-то новенькое! Повтори вопрос, может, величины возбуждения на входе малы?

У: Повторяю вопрос.

*ИК(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 *RS*.
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. Провожу трассировку к выходному нейрону *Выхб*, соответствующему решению *R6*.

Здесь идет самообучение системы «учитель — нейрокомпьютер». Полное или близкое к нему самостоятельное обучение нейрокомпьютера возможно тогда, когда сеть многоуровневая, т. е. одни выводы являются посылками для других. При этом сеть должна быть знакома хотя бы с понятиями «хорошо» и «плохо», т. е. учитывать критериальную функцию, необходимую для моделирования. Это высшие сферы обучения (самообучения), которые мы здесь не рассматриваем.

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 истине.

Например, диалог в процессе медицинской диагностики формируется, несомненно, на основе отработанной стратегии «допроса». Предполагается, что вопросы структурированы и даже соблюдают порядок следования, отображенный графически, так что одни вопросы исключают другие.

В то же время принцип нейросети применим и тогда, когда вопросы нелогичны, несовместимы. Возбуждения всех нейронов 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 устраним трассировкой нового решения. Правда, новым решением — реакцией на 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, палец.

*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):* Молились ли Вы на сон грядущий?

Л: 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 Пенсионный фонд Российской Федерации!



# 5 НЕЙРОННЫЕ СЕТИ СОБРАТНЫМИ СВЯЗЯМИ

## 5.1 К вопросу о происхождении человека

В рассказе "Пиймо к ученому соседу» А.П. Чехов от лица своего героя пишет: «...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 бы 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 бы от нее хоть немножко пахло бы обезьяной, которую мы каждый вторник видим у Предводителя Дворянства?»

Попробуем построить проект, эскиз, набросок фрагмента нейросети, отражающей глубину умозаключений чеховского героя и наше горячее желание поправить и дополнить его (рис. 5.1).

Посылка «человек произошел от обезьяны» обязательно должна привести к абсурдному выводу «у человека есть хвост». Диким голосом современного студента не удивить. Обучение — трассировку, приводящую к образованию динамических путей возбуждения, при которой могут быть использованы многие нейроны, обозначим пунктиром. Тогда должна существовать отрицательная обратная связь, ведущая к нейрону входного слоя, «ответственному» за указанный тезис. Значит, этот нейрон должен получать возбуждение нейрона выходного слоя с помощью дендрита с отрицательным синапсическим весом. Отрицательная связь помечена на рисунке знаком "—".

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 не-

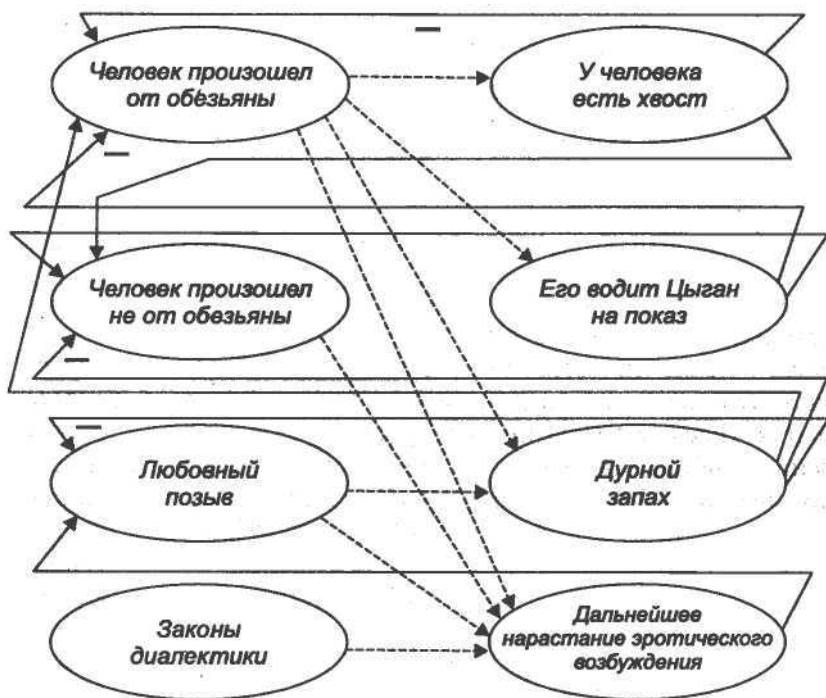


Рис. 5.1. Нейросеть для решения вопроса о происхождении человека

которые их дендриты получают возбуждение извне, а некоторые — от других нейронов сети.

Однако сделанный вывод усиливает важность посылки «человек произошел не от обезьяны». Величина возбуждения этой посылки справедливо должна быть увеличена, что достигается положительной обратной связью. То же самое относится к не менее одиозному выводу «его водит Цыган на показ».

Но одновременно действует и третий довод: «То, что дурно пахнет, недостойно любви!» Дурной запах давит, угнетает позывы любви и способствует сомнению о происхождении человека. Мы видим действенную роль отрицательной обратной связи «запах — любовь», преодолеваемую лишь с помощью крайне важной продукции фирм *Ночная бабочка* и *Красный киллер*.

## 5.2 Как же вводить обратные связи?

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, выпрашивая пряник, сообщает, что только что видел Васю у Аполлинарии, то дядя Рамзай, корректируя текущую информацию, должен игнорировать или учесть незначительно (в случае неверия собственным глазам) заявление Вовочки.

Желательно, чтобы достоверность подобной информации (про Васю и Оксану) автоматически снижалась либо, наоборот, достоверность информации о ситуациях без их участия увеличивалась.

Напоминаем, что не стоит гоняться за достоверностью, требуя принадлежности ее величины диапазону  $[0, 1]$ , а использовать более общее понятие *веса информации*. Очевидно, вес следует рассматривать как функцию времени. Ведь логично, что с течением времени вероятность участия Васи и Оксаны в последующих ситуациях должна восстанавливаться.

Что необходимо для работы сети подобным образом?

1. Надо с большей определенностью дифференцировать ситуации на выходном слое. Например, если в решениях  $R1$  и  $R2$  замешаны не только Вася и Оксана, но и другие участники, то это затрудняет принятие конкретного решения только по Васе и только по Оксане. Необходимо корректировать принимаемое решение по вхождению комбинации  $A1 \& C1$  в предикат, определяющий это решение, и не связывать, не объединять эту комбинацию, например, с комбинацией  $A \setminus \& C1$ , входящей в тот же предикат. Следует расширить множество  $\{Ri\}$  принимаемых реше-

ний, повторяя при необходимости одно и то же решение для разных эталонных комбинаций.

2. Необходимо ввести обратную связь, влияющую на решения относительно Васи ( $A1$ ) и Оксаны ( $C1$ ), в части достоверности или веса информации, т. е. на величины возбуждения нейронов  $A1$  и  $C1$ , даже вопреки попыткам искусственно задать высокие значения этих величин.

Приведем конкретные рекомендации.

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

2. Установление отрицательной обратной связи, ведущей от каждого нейрона выходного слоя, отвечающего за решение с участием Васи, к нейрону  $A1$  входного слоя.

3. Установление отрицательной обратной связи, ведущей 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 увеличивает достоверность участия Пети в дальнейших исследованиях дяди Рамзая.

5. Установление положительной обратной связи от нейронов выходного слоя, отвечающих за решения по Оксане, к нейронам  $C2, \dots, C5$ .

Строго следуя данной рекомендации, строим более подробную систему предикатов, которая воплощается нейросетью:

$$A1 \& B1 \& C1 \rightarrow R1$$

$$A1 \& B1 \& C2 \rightarrow R1$$

$$A1 \& B1 \& C3 \rightarrow R1$$

$$A1 \& B1 \& C4 \rightarrow R1$$

$$A1 \& B1 \& C5 \rightarrow R1$$

$$A1 \& B2 \& B3 \& C1 \rightarrow R2$$

$$A1 \& B2 \& B3 \& C2 \rightarrow R2$$

$$A1 \& B2 \& B3 \& C3 \rightarrow R2$$

$$A1 \& B2 \& B3 \& C4 \rightarrow R3$$

$$A1 \& B2 \& B3 \& C5 \rightarrow R3$$

$$A2 \& B3 \& C1 \rightarrow R4$$

$$A2 \& B3 \& C2 \rightarrow R4$$

$$A2 \& B3 \& C3 \rightarrow R4$$

$$A2 \& B3 \& C4 \rightarrow R4$$

$$A2 \& B3 \& C5 \rightarrow R4$$

$$A2 \& B1 \& B2 \& C1 \rightarrow R5$$

$$A2 \& B1 \& B2 \& C2 \rightarrow R5$$

$$A2 \& B1 \& B2 \& C3 \rightarrow R5$$

$$A2 \& B1 \& B2 \& C4 \rightarrow R5$$

$$A2 \& B1 \& B2 \& C5 \rightarrow R5$$

Заметно резкое возрастание числа нейронов выходного слоя.

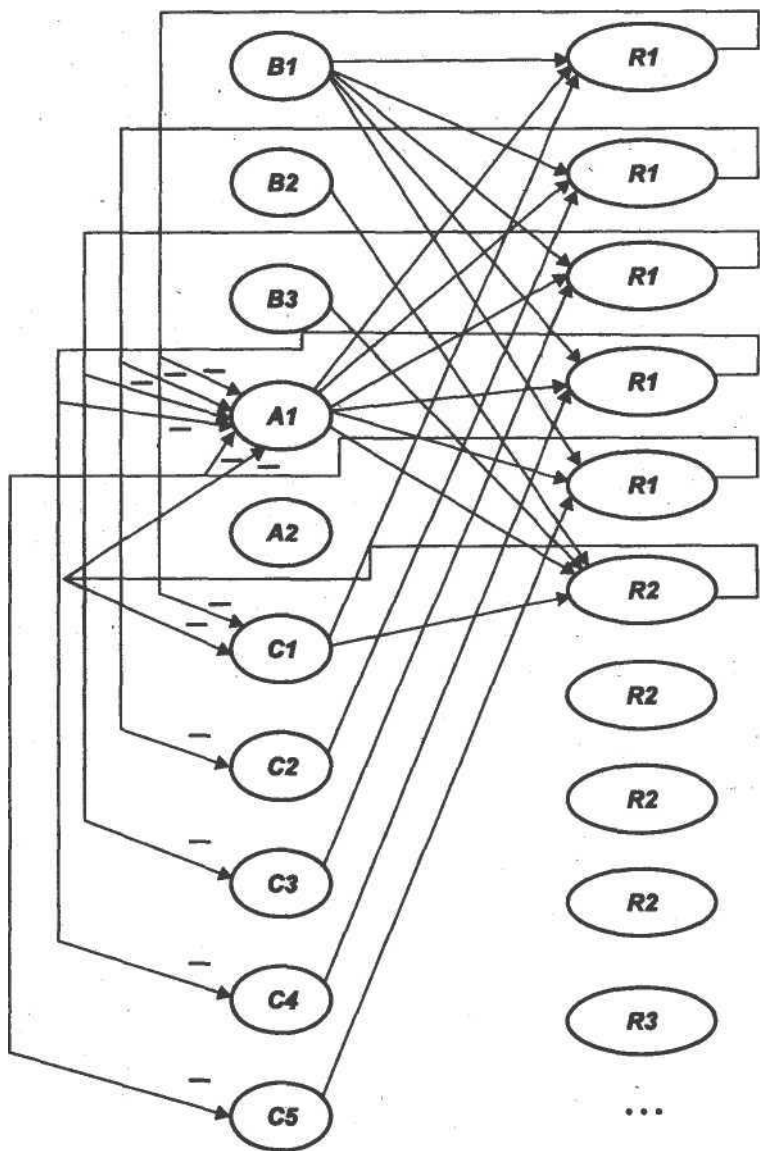


Рис. 5.2. Нейросеть с обратными связями

Что ж, нас теперь интересуют не просто выводы, но и то, каким путем каждый из них получен.

Это еще раз свидетельствует о том, что не всегда целесообразно использовать обобщенные эталоны, объединяя ситуации, ведущие к одинаковому выводу. Весьма часто, особенно при введении обратных связей, необходимо различать ситуации. Это отражает принцип мозга, где царит избыточность, обеспечивающая обилие путей и вариантов, развиваемость, резервирование, устойчививость и надежность.

На рис. 5.2 представлен фрагмент начала построения возможной нейросети, которую мы вправе считать окончательным результатом своих исследований и готовы предъявить Заказчику. Во избежание путаницы положительные обратные связи не указаны.

Следует обратить внимание на то, что, вводя обратные связи, мы ограничились только проблемой повышения достоверности исходных данных на основе текущего состояния событий.

Модуль веса обратной связи в нашем случае должен быть убывающей функцией времени. Пользуясь допустимыми условностью и приближенностью, преследуя цель скорее качественную, нежели количественную, можно довольно просто рассчитать величину отрицательной обратной связи. Например, если считать, что на один визит Вася расходует не менее 25 мин., то вес  $\omega$  одного из дендритов нейрона  $A1$  (аналогично другим нейронам входного слоя) определяем по формуле

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

- где  $k$  — экспериментально подобранный демпфирующий коэффициент, возможно, зависящий от величины подаваемого возбуждения при вводе информации для корректной компенсации обратной связи;
- $t$  — текущий момент поступления новой информации, требующей использования нейросети;
- $t_0$  — последний момент времени вовлечения соответствующего объекта в действие (хранится в составе информации о данном объекте —  $A1, A2, C1$  и др.);
- $t_0 + 25$  — ожидаемое время повторного допустимого «использования» особы в деле.

При этом используется приведенная ранее передаточная функция (3.1) для нахождения величины  $V_i$  возбуждения  $i$ -го нейрона.

# 6 НЕЙРОСЕТЕВЫЕ САМООБУЧАЮЩИЕСЯ СИСТЕМЫ УПРАВЛЕНИЯ

## 6.1 Самообучение на основе ситуационного управления

Известной основой аппаратной поддержки самообучающейся системы управления является ассоциативная вычислительная система (ВС).

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 к данным 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. Запросные поля обрабатываются по ключевому слову, производится поиск на основе сравнений и выдается результат из одного (или более) ответного поля. При помощи маскирования выделяются только те поля ключевого слова, которые используются в процессе поиска для сравнения.

Типичными операциями сравнения ассоциативной памяти являются следующие: «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 они обладают обычными свойствами, производят сложные преобразования данных и принадлежат типу ОКМД (STARAN, PEPE) или

МКМД [9]. Для параллельного обращения с целью ускорения поиска АП разбита на модули (в STARAN 32 модуля).

Когда в 1980 г. был провозглашен так называемый японский вызов относительно построения ВС сверхвысокой производительности, то в одном из важнейших пунктов предусматривалась необходимость самого широкого использования принципов *самообучающихся систем* — систем, способных накапливать опыт и выдавать результат решения задачи без расчетов — 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, это всего лишь привычный способ использования интерполяционных таблиц, например таблицы логарифмов.

Пусть мы пользуемся значениями функции  $y = f(x)$ . Можно запрограммировать ее вычисление на персональном компьютере, и, при необходимости задавая значение  $x$ , запускать программу, получая значение  $y$ .

Предположим, та же функция довольно сложна, а ее вычисление — важный элемент алгоритма управления в реальном времени. В таком случае зададим эту функцию таблично, а для ускорения выборки включим в ВС ассоциативную память. Предусмотрим операции, позволяющие производить простейшую интерполяцию, а именно для данного значения  $x$  найти наибольшее  $x_1 \leq x$  и наименьшее  $x_2 > x$ . Выбрать для них  $f(x_1)$  и  $f(x_2)$ . Тогда вместо вычисления точного значения  $f(x)$  процессор производит линейную интерполяцию с заданной точностью:

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

, В жизни очень часто приходится решать трудно формализуемые задачи, подобно предложенной дядей Рамзаем. Но в результате обучения и приобретения опыта складывающиеся в дальнейшем ситуации оцениваются на «похожесть» и принимается решение.

Как, например, определить, на сколько повернуть рулевое колесо, чтобы остаться на дороге? Ведь первый раз мы въезжали в бордюры! Видимо, в нашем сознании постепенно сложилась и зафиксировалась некая таблица.



Пожалуй, наиболее полно и с точки зрения практики 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 пристрелянные по ней установки. Такая работа продлевается и по фиктивным целям — реперам. Постепенно на данной местности запоминаются пристрелянные установки по многим целям и реперам. Тогда подготовка данных по вновь появляющимся целям резко упрощается, так как сводится к переносу огня от ближайшей цели или репера, т.е. к внесению поправок по дальности и направлению. Доказано, что при этом достаточно сразу назначить захват цели в узкую вилку, что сокращает время пристрелки. Правила стрельбы существуют века, однако вряд ли кто-то осознал, что речь идет о реализации самообучающейся системы, аналога которой не существует.

Для *трудно формализуемых задач управления* или для *увеличения производительности* вычислительных средств известно так называемое *ситуационное управление*, предложенное Д.А. Поспеловым [16]. Оно заключается в том, что для каждого значения вектора, характеризующего сложившуюся ситуацию, известно значение вектора, описывающего то решение, которое следует принять. Если все ситуации отразить невозможно, необходимо задать *правило интерполяции* (аналог интерполяции, обобщенной интерполяции).

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

Компоненты векторов  $X$  и  $Y$  могут быть целыми и вещественными числами, а также булевыми. Изначально они даже могут иметь нечисловую природу: «темнее — светлее», «правее — левее», «ласковее — суровее» и т.д. Проблема численной оценки качественных или эмоциональных категорий здесь не рассматривается.

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 (АП) и общий вид ее обработки показаны на рис. 6.1.

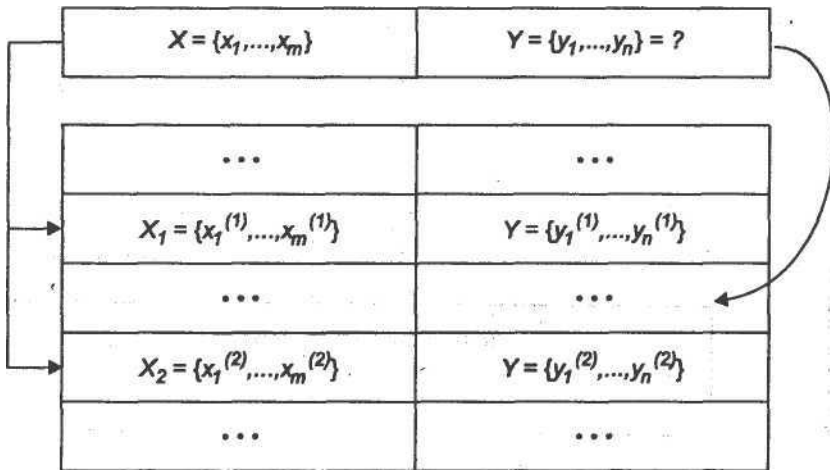


Рис. 6.1. Пример использования ассоциативной памяти

Пусть для входной ситуации  $X$  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 известны 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  $x_i \geq x_i^{(2)}$  ( $x_i \leq x_i^{(2)}$ ),  $i = 1, \dots, m$ . Это обусловлено преимуществом интерполяции по сравнению с экстраполяцией. Таким образом, определяется «вилка», которой принадлежит входная ситуация. Тогда, опираясь на известные граничные решения, необходимо получить промежуточное решение для данной ситуации, что достигается методом той же обобщенной интерполяции:

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

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

Где  $r(X, X_2)$  (аналогично  $r(X_1, X_2)$ ) — расстояния между векторами  $X$  и  $X_2$ , а также  $X_1$  и  $X_2$ .

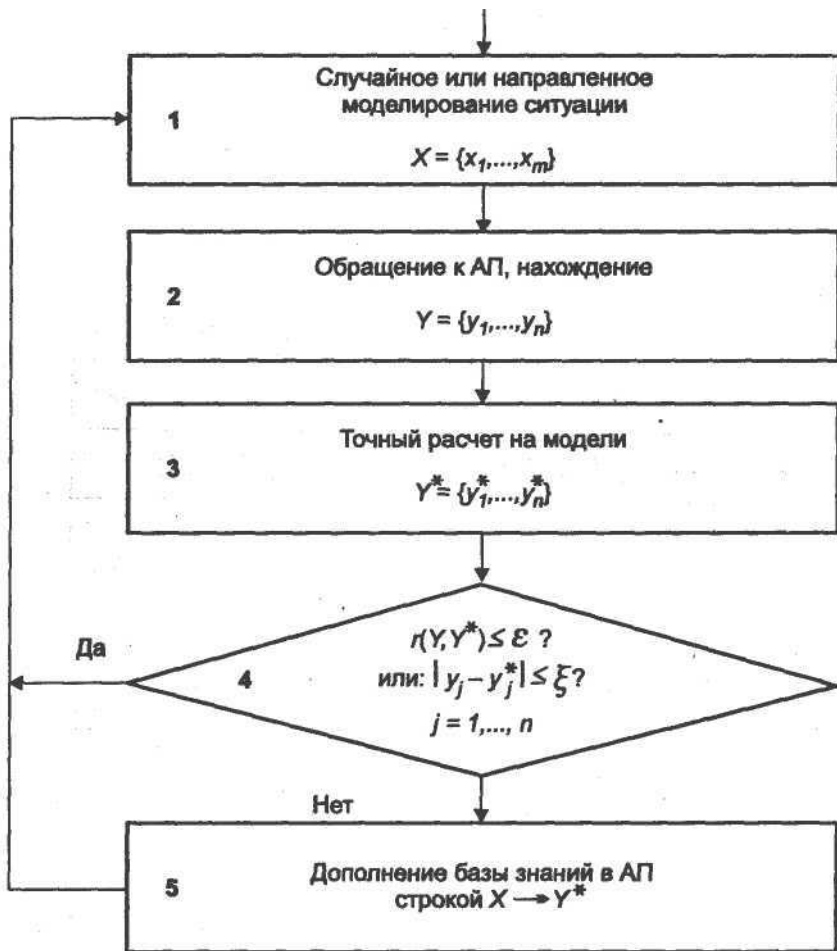


Рис. 6.2. Схема самообучения с ассоциативной памятью и моделью

Если точность  $Y$  достаточна, принципиально возможно дополнение АП новой строкой " $X - Y$ ", т.е. информацией о новом опыте.

Впрочем, динамика развития и уточнения АП, как *базы знаний*, представляется иной. Далеко не всегда целесообразно развивать базу знаний только тогда, когда принято ошибочное реше-

ние. «Учение на ошибках» может привести к трагедии (или к срыву процесса управления), особенно на этапе обучения системы.

Поэтому обучение системы целесообразно проводить на достаточной точной *модели*, максимально использующей точные вычисления компонент решения. Модель 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. Схема *алгоритма работы* системы представлена на рис. 6.2.

## 6.2 Нейросетевое воплощение

Мы уже построили нейросеть для важной задачи принятия решений и вручили ее заказчику — дяде Рамзаю.

Теперь еще раз тщательно проследим за применением принципа динамического совмещения обучения системы управления с получением решений и установим в схеме на рис. 6.2 целесообразность, место и способ использования нейронной сети как основного реального средства искусственного интеллекта. Прежде всего заметим, что основным естественным и целесообразным местом применения нейросети здесь является база знаний, ранее реализованная с помощью ассоциативной памяти (см. рис. 6.1). Надо только позаботиться о том, чтобы там находились не численные значения информации, а ее вес или достоверность, т. е. не сама информация должна обрабатываться нейросетью, а ее логический эквивалент. Ведь принцип искусственного интеллекта направлен на параллельную обработку логических высказываний, а вовсе не на числовую обработку информации, как это делается при решении «нейроподобных» задач с использованием нейросети в роли спецпроцессора.

Пусть нейросеть в качестве базы знаний имеет вид, представленный на рис. 6.3.

Нейроны-рецепторы закреплены за значениями элементов входного вектора. Значит, в режиме обучения установилось соответствие между величиной возбуждения *p-go* нейрона входного слоя и достоверностью того, что  $x_i = x_i^p, i=1, \dots, m$ . Здесь (см. подразд. 2.2) могут фигурировать не точные значения параметров, а некоторые диапазоны их изменения.

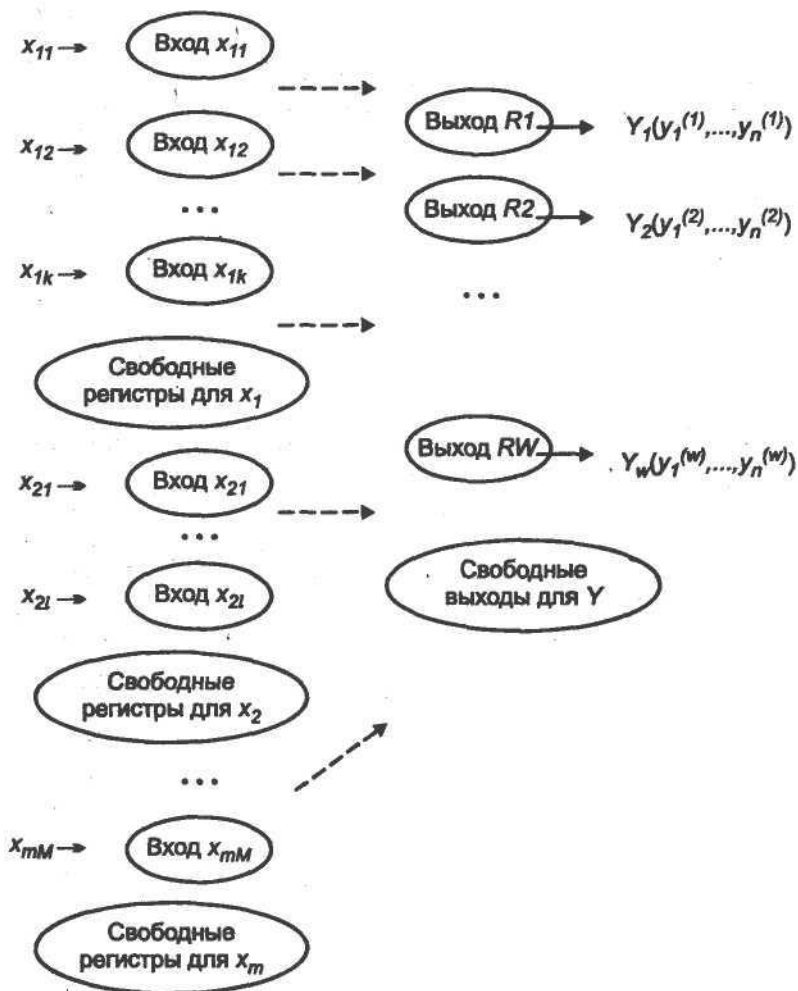


Рис. 6.3. Управляющая система с нейросетью

Как указано выше, при обучении положим эту достоверность равной единице, а в процессе распознавания она может быть какой угодно, даже не удовлетворяющей свойству полноты событий.

После обучения с помощью различных эталонов методом трассировки можно добиться соответствия вида  $\{Входы\ x_i\} \rightarrow Rq$ ,

наполнив тем самым базу знаний. Каждое решение  $Rq$  при этом тоже указывает на определенное значение вектора - результата  $Y_q$ , компоненты которого — суть управляющие воздействия системы, т.е. нейроны  $R_q$  фактически закрепляются за ситуациями, относительно которых известны векторы характеристик или значений параметров принимаемых решений, т.е. векторы значений управляющих воздействий.

Для нахождения этих значений в зависимости от исходных данных применяется моделирование наряду с экспериментом или с экспертными оценками. Таким образом, модель, эксперимент или эксперт играют роль учителя.

Первоначально обученная таким образом нейросеть используется в рабочем режиме распознавания и в режиме совместной работы с моделью. Конечно, справедливо считать, что нейросеть обучена недостаточно, и подобно «пристрелке реперов» любой удобный случай используется для того, чтобы с помощью модели испытать и в случае необходимости дополнить знания нейросети.

Для этого модель случайно или целенаправленно — по обоснованному плану, генерирует некоторую ситуацию, характеризующуюся значением компонент входного вектора  $X$ . По каждой компоненте определяется вес или достоверность того, что ее значение совпадает с подмножеством значений представленных входным слоем нейросети или с диапазонами значений.

Например, известна реакция сети на значения  $x = 2$ ,  $x = 5$ ,  $x = 6$ , а модель сгенерировала значение  $x = 5,7$ . Это может означать необходимость (реализуется некоторая процедура) формирования значений возбуждения  $V_{x=2} = 0$ ,  $V_{x=5} = 0.3$ ,  $V_{x=6} = 0,7$ . Здесь индексы указывают нейроны входного слоя, соответствующие данному значению параметра. Такая процедура выполняется по всем компонентам, отображенным входным слоем.

Пусть возбуждения входного слоя привели к преодолению порога возбуждения нейронов выходного слоя  $R5$  ( $V_{R5} = a$ ),  $R7$  ( $V_{R7} = b$ ),  $R12$  ( $V_{R12} = c$ ). При этом нейрону  $R5$  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 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 точность. Если удовлетворяет, делаем положительный вывод об обученности нейросети и продолжаем испытание по другим исходным данным. Если нет, сеть необходимо «доучить», продемонстрировав высокий уровень обратной связи.

Для этого придется ввести в действие новые рецепторы в соответствии с теми значениями исходных данных или их диапазонами, которые ранее не были представлены. Например, придется ввести рецептора соответствии со значением  $x = 5,7$ .

Далее, выделим нейрон выходного слоя в соответствии с правильным решением, полученным в результате моделирования. Затем выполним трассировку для того, чтобы появление 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 прекратятся сбои, что маловероятно. Следовательно, в таком режиме она должна работать в течение всего жизненного цикла, реализуя известную пословицу «Век живи — век учись».

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

Такая аппроксимация выполняется и в более явном виде, ибо каждая **компонента**  $y_j$  решения  $Y = \{y_1, \dots, y_n\}$  может быть найдена отдельно в результате предварительной трассировки (рис. 6.4). Следовательно, сеть строится и обучается так, чтобы заданное значение  $X = \{x_1, \dots, x_m\}$  приводило к максимальному (или усредненному) значению возбуждения нейрона выходного слоя, указывающего на соответствующее значение  $y_1$ , к максимальному (или усредненному) значению возбуждения другого нейрона вы-

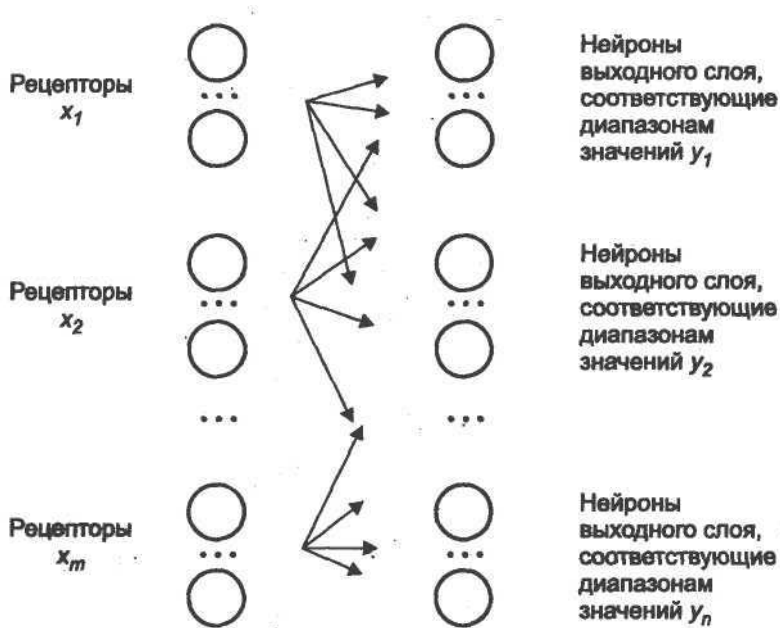


Рис. 6.4. Раздельное нахождение управляющих параметров с помощью нейросети

ходного слоя, указывающего на значение  $y_2$ , и т.д. В результате выходной слой разбивается на области, каждая из которых закреплена за своим **параметром**  $y_i, i = 1, \dots, n$ . Тогда полученное преобразование можно условно записать  $X \rightarrow \{Вых(y_1), вых(y_2), \dots, x(y_n)\}$ .

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



# 7 ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ НЕЙРОСЕТИ

## 7.1 ПРОЛОГ-программа

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

Язык представляет базу знаний как совокупность фактов и правил вывода; процедурная структура позволяет включать конструкции любых других алгоритмических языков, т.е. ПРОЛОГ является логической надстройкой, объединяющей лишь операции вывода. Формулируется *цель* логического вывода, и если она не противоречива, выявляются факты, ее породившие.

Рассмотрим упрощенную задачу в виде ПРОЛОГ-программы, содержащую характерные элементы проблемы достижения сложной цели на основе фрагмента базы знаний, 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*. Пусть база знаний имеет следующий вид.

*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),

ХОУ

**Процедура «сестра»:**

**сестра** (X, Y): женщина (X), родитель (P, X), родитель (P, Y),

X<>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 (запись цели образует *фрейм*), с которой мы обратились к этой БЗ, например

дядя(X, Y),

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

Используем прием трансформации цели, заключающийся в рекурсивном переборе различных вариантов подстановки вместо предикатов, составляющих сложную цель, правых частей кловов соответствующих процедур. В результате производится фиксация варианта связывания переменных и унификация, при которой отбрасываются несовместимые варианты, т.е. противоречашие

фактам и правилам. Варианты связывания всех переменных, прошедшие все этапы унификации, являются решением.

Изложим последовательность действий.

Находим первый (и единственный) предикат цели *дядя* ( $X, Y$ ). Заменяем найденный предикат правой частью процедуры с этим именем, записанной в БЗ. Получим трансформированную цель — фрейм:

$$\text{брат}(X, P), \text{родитель}(P, Y).$$

С первым предикатом этого фрейма действуем аналогично, получаем фрейм:

$$\text{мужчина}(X), \text{родитель}(Q, X), \text{родитель}(Q, P), X \langle \rangle P, \\ \text{родитель}(P, Y).$$

При этом во избежание коллизии, развивая фрейм цели, вводим новые переменные, отличные от тех, которые уже были использованы в записях БЗ.

Вновь входим в процедуру с именем первого предиката цели и начинаем первый уровень ветвления, используя первый клоз процедуры — факт *мужчина* (*иван*). С его помощью производим первое связывание переменных, т.е. подстановку конкретного значения. Предикат цели, породивший это связывание, из трансформируемой цели исключается, т.е. заменяется своей «пустой» правой частью:

$$\text{родитель}(Q, \text{иван}), \text{родитель}(Q, P), \text{иван} \langle \rangle P, \\ \text{родитель}(P, Y).$$

Обращаемся к процедуре "*родитель*", начиная второй уровень ветвления. Первый клоз этой процедуры *родитель* (*марья*, *иван*) определяет дальнейшее связывание переменных:

$$\text{родитель}(\text{марья}, P), \text{иван} \langle \rangle P, \text{родитель}(P, Y).$$

Вновь входим в процедуру "*родитель*" (третий уровень ветвления), находим клоз *родитель* (*марья*, *иван*). Трансформируем цель, получаем новый фрейм:

$$\text{иван} \langle \rangle \text{иван}, \text{родитель}(\text{иван}, Y).$$

Возникает противоречие, т.е. *унификация не проходит*.

Ищем на данном шаге ветвления другой вариант связывания, находим следующий клоз:

*родитель(марья, василий).*

Трансформируем цель:

*иван <> василий, родитель(василий, Y) → родитель(василий, Y).*

Вновь входим в процедуру "родитель", но не находим клоза, в котором *василий* указан как чей-либо *родитель*, т.е. вновь унификация не проходит.

Реализуем стратегию поиска с ветвлением и возвращением назад — *backtracking*. На втором уровне ветвления исследуем клоз, в котором *иван* указан как сын: *родитель(петр, иван)*. Цель трансформируется в следующий фрейм:

*родитель(петр, P), иван <> P, родитель(P, Y).*

Вновь (на третьем уровне ветвления) обращаемся к процедуре "родитель" и выбираем первый клоз, в котором *петр* указан как отец — *родитель(петр, ирина)*.

Цель трансформируется:

*иван <> ирина, родитель(ирина, Y) → родитель(ирина, Y).*

Входим в процедуру "родитель", но в ней нет клоза, в котором *ирина* указана как родитель. Унификация не проходит.

Возвращаемся на второй уровень ветвления и не находим больше клозов, где *иван* указан как сын. Далее на первом уровне ветвления в процедуре «*мужчина*» выбираем для дальнейшего испытания следующий клоз *мужнина(василий)*.

Цель принимает вид (фрейм):

*родитель(Q, василий), родитель(Q, P), василий <> P,  
родитель(P, Y).*

На втором уровне ветвления находим единственный клоз, в котором *василий* указан как сын. Цель трансформируется в соответствии с новым связыванием переменных, обусловленным найденным клозом *родитель(марья, василий)*:

*родитель(марья, P), василий <> P, родитель(P; Y).*

На третьем уровне ветвления находим первый среди клозов, где *марья* указана как родитель: *родитель (марья, иван)*. Связываем тем самым переменные, цель трансформируется:

*василий* < *иван, родитель (иван, Y)* → *родитель (иван, Y)*.

В процедуре «*родитель*» находим клуз, 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, найдем 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 *backtracing'a* (ветвление есть, а возврата нет), простоту 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, так как каждый раз обрабатывается лишь первый предикат каждого фрейма).

Однако представляется, что нейросетевая технология, основанная на естественном параллелизме, может оказаться эффективнее.

## 7.2 Нейросеть для задачи логического вывода

Рассмотрим предварительно один аспект обучения нейросети методом трассировки.

Обобщив подход, изложенный в разд. 2, на логику передаточных функций, мы заявили, что нашли способ построения *уже обученных* нейросетей.

Впоследствии (разд. 3 и 4) при отсутствии некоторых связей в типовых сетях обосновано их принудительное введение в случае целесообразности.

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, т. е. мы сразу можем строить обученную нейросеть, синапсические связи в которой имеют единичный (максимальный) вес.

Последуем этим путем и, *рассчитав все возможные выводы*, составим (рис. 7.1) нейросеть, соответствующую фрагменту беспредельной базы знаний о жителях далекого таежного села.

Для того чтобы максимально сохранить информацию о иерархии выводов, переменные — значения возбуждения рецепторов будем вводить только на первый уровень логического анализа в качестве параметров процедур-фактов. Ведь как ни формализуй, а окончательных рекомендаций по формированию нейросети не получить, так как это формирование — моделирование, а моделирование — искусство. Впрочем, каждый раз это ярко демонстрирует выбор передаточной функции. Здесь мы рекомендуем использовать передаточную функцию:

$$V = \sum_j V_j;$$

$$V_j = \begin{cases} V - h, & \text{если } h \leq V < 1; \\ 1 - h, & \text{если } V \geq 1; \\ 0, & \text{если } V < h. \end{cases}$$

Порог  $h$  подбирается экспериментально, в данном случае целесообразно положить  $h = 0,4$ .

Мы видим, что сеть отображает только факты, а где же правила? Они использованы и забыты? И да, и нет. Вот тут-то и проявляется вторая парадигма обучения — парадигма ученика. Ибо правила вывода оказались использованными при обучении и скрыты. Только путь индуктивного мышления способен их восстановить, т.е. восстановить закономерность по сформированным связям.

Как же тогда работать с этой сетью? Представим себе два режима работы: для фискального органа - налоговой инспекции, требующей общей картины социального состояния, и для информационно-аналитической службы безопасности, интересующейся гражданами индивидуально, по выбору и привлечению.

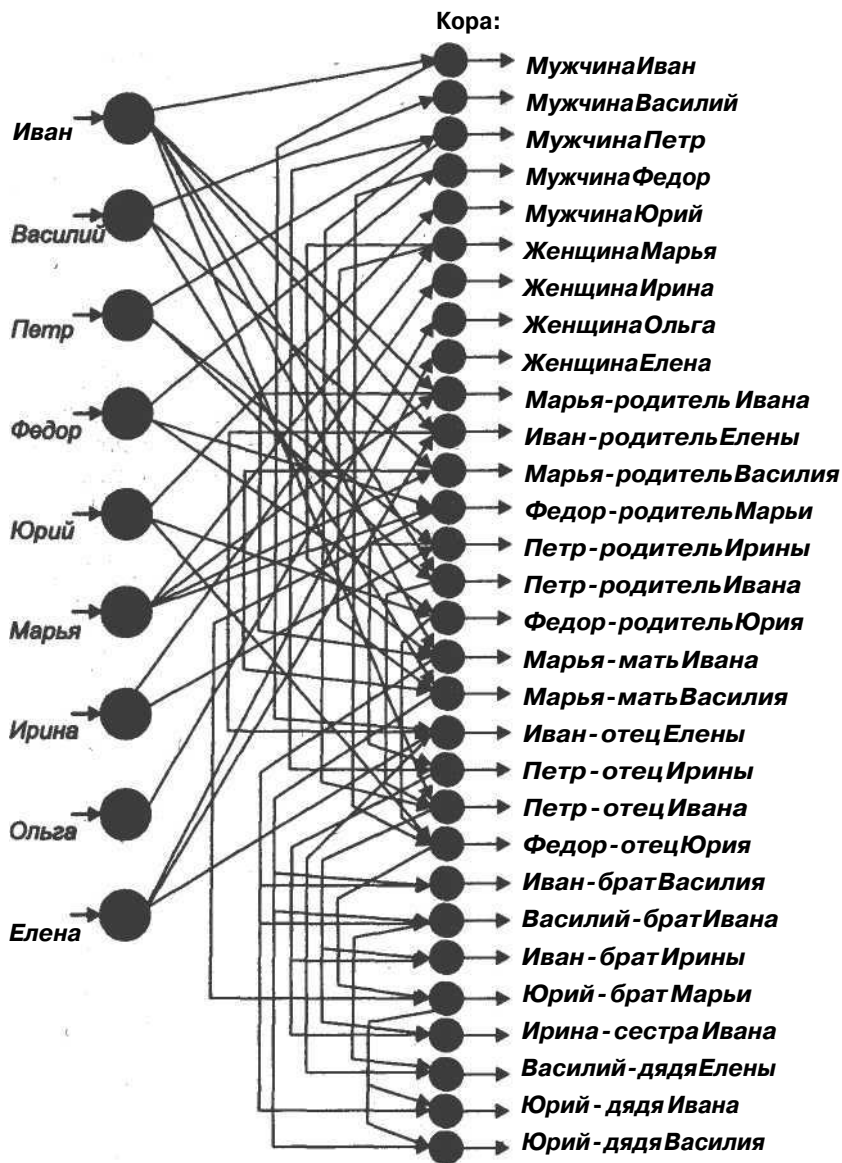


Рис. 7.1. База знаний антрополога-исследователя

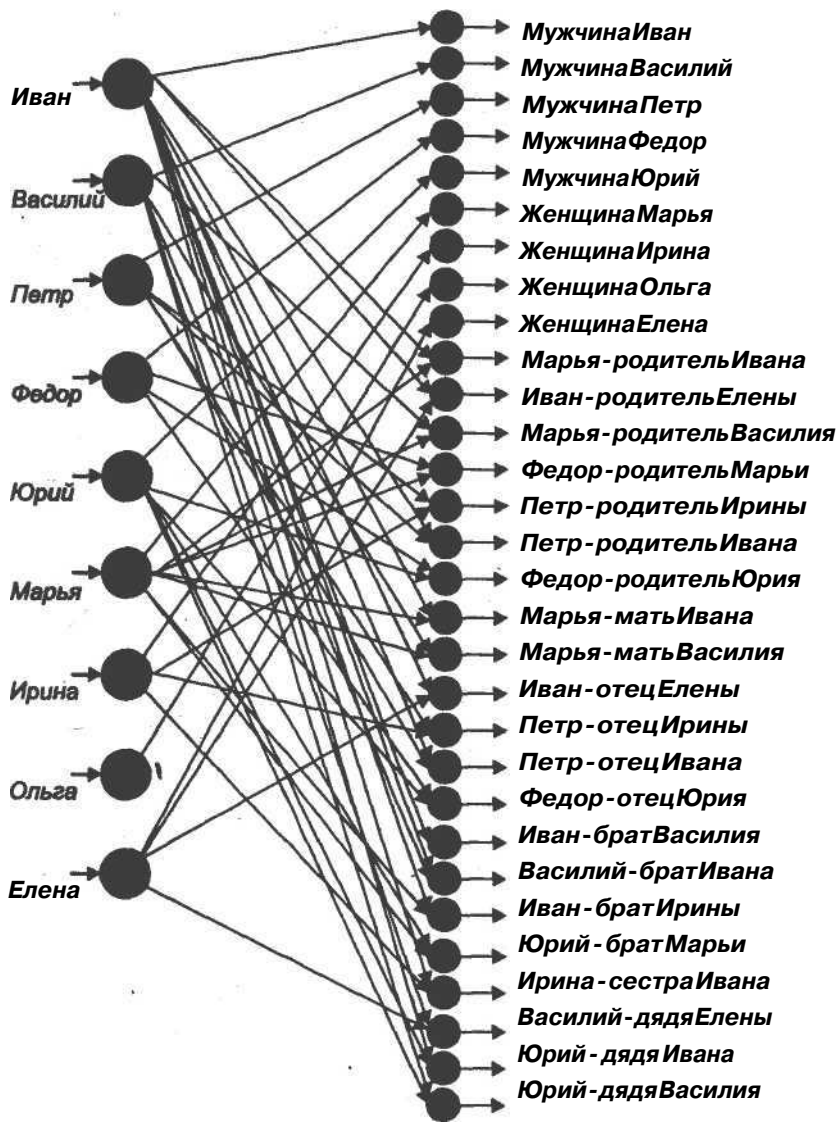


Рис. 7.2. База знаний участкового уполномоченного



А именно если максимально возбудить все нейроны-рецепторы, то можно извлечь полную информацию о клиентах: кто из них мужчина, кто — женщина, в каком родстве они состоят.

В то же время при возбуждении на входе нейронов-рецепторов, соответствующих *Ивану* и *Василию*, 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 возбудим нейрон-рецептор, соответствующий *Марье*, и посмотрим на реакцию сети. Сеть высоким возбуждением своих нейронов нам сообщает: "Да, действительно, *Марья* — мать *Ивана* и *Василия*".

Заметим, что можно подавать на вход сети и не единичные значения возбуждений. Например, установив, что в деле участвовал не то *Иван*, не то *Василий* (уж очень они похожи!), мы зададим соответствующие значения возбуждений равными 0,5 и установим высокую величину возбуждения нейрона, соответствующего дяде и того, и другого — *Юрию*. И тогда логично возложить на него ответственность за поведение племянников.

Таким образом, составляя нейросеть, мы руководствовались желанием сохранить причинно-следственные логические связи, обусловленные правилами. Этим обуславливается возможность использования метода индуктивного мышления для установления, например, изначальных фактов, достаточных для вывода: *Юрий* — дядя *Ивана*. Мы даже назвали эту нейросеть базой знаний антрополога-исследователя. Таким образом, имитируется организация *коры*, где все выводы равноправны и служат как формированию внутризаданных, промежуточных выводов, так и для решения некоторой внешней 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 специального формирования величин возбуждения нейронов, что и отражено в предлагаемой передаточной функции.

С другой стороны, зачем нам знать и делать вывод о том, что *Юрий* — дядя *Ивана*, потому что он — брат *Марьи*? Мы можем эти факты задать независимо, предварительно вычислив на языке ПРОЛОГ

Тогда получим фрагмент базы знаний участкового уполномоченного (рис. 7.2), что в самой простой форме отражает анкетирование, не подлежащее глубокому исследованию.

# 8 НЕЙРОСЕТИ И УПРАВЛЕНИЕ ФИНАНСАМИ

## Табличный метод - 8.1 основа искусственного интеллекта

В общем-то, принципы мозговой деятельности известны и активно используются. Мы применяем незримые таблицы в нашей памяти, принудительно и вольно заполняемые 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: и школьник, проводящий бессонные ночи за букварем, и умудренный опытом профессор. Ибо с теми же таблицами мы связываем не только принятие решений, но и двигаемся, ходим, играем в мяч.

Если противопоставить ассоциативному мышлению математические вычисления, то каков же их вес в жизни человека? Как шло развитие человека, когда он вообще, не умел считать? Пользуясь ассоциативным мышлением, умея интерполировать и экстраполировать, человек накапливал опыт. (Кстати, вспомним тезис Д. Менделеева: «Наука начинается тогда, когда начинают считать».) Можно спросить читателя: «Сколько раз сегодня Вы считали?» Вы водили автомобиль, играли в теннис, торопились на автобус, планируя свои действия. Представляете, сколько бы Вам пришлось высчитывать (да еще где взять алгоритм?), для того чтобы поднять ногу на тротуар, минувя бордюр? Нет, мы ничего не вычисляем ежеминутно, и это, пожалуй, основное в нашей интеллектуальной жизни, даже в науке и бизнесе. Механизмы ощущений, 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, 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 искусственного интеллекта. 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 яснее понимаем, что моделирование процессов мозговой деятельности, воспроизводящих методы ассоциативного мышления, открывает самые широкие возможности, присущие живым организмам. Это и высокий универсализм, и высокое быстроедействие.

Природа реализует самые простые принципы, утверждая, что «гениальное просто». Сложное обусловлено количеством и структурированием простого. Механизмы логического вывода, реализованные мозгом человека, основаны на простых логических элементах типа «если — то», «посылка — следствие», приемлемых даже для простейших. Огромное количество таких элементов в столь же простом взаимодействии, соединяясь и выстраиваясь в логические цепочки, воспроизводя иерархию и рекурсию, образуют сложные выводы, тотчас же обращенные для новых выводов.

Но, пожалуй, основное достоинство мозга — способность параллельного выполнения сложных логических предикатов и последовательного вычисления длинных логических цепочек. Такие действия достигаются, во-первых, при одновременной обработке всех сигналов, поступивших на рецепторы, и, во-вторых, при реализации каждым нейроном передаточной функции для поступивших сигналов. Передаточная функция является суррогатом логических функций булевых переменных для обработки действительных переменных.

Важно подчеркнуть, что информация перерабатывается в нейросети не непосредственно, а опосредствовано — на уровне оценок, на уровне ее воздействия на величину возбуждения рецепторов или нейронов входного слоя подструктур.

А как же наше формальное мышление, вторая сигнальная система по Павлову, а как конструирование алгоритмов и правил вычислений? Полагаем, что это — все более и более высокие уровни того же логического мышления, наслоение 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.

Сформулируем теперь достаточный, сегодняшний принцип построения нейросети, как элемента ИИ:

1. Следует признать, что основа имитации нейроструктуры мозга — это метод табличной интерполяции.

2. Таблицы заполняются или по известным алгоритмам вычислений, или экспериментально, или экспертами.

3. Нейросеть обеспечивает 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 в преобразовании не самой исходной информации, а оценок этой информации, в подмене информации величинами возбуждения рецепторов, искусно распределенных между видами, типами, параметрами, диапазонами их изменения или отдельными значениями.

6. Нейроны выходного слоя каждой подструктуры своим возбуждением указывают на соответствующие решения. В то же время эти сигналы возбуждения на правах исходной опосредованной информации могут использоваться в следующем звене логической цепочки без внешнего вмешательства в рабочем режиме.

## 8.2 Мониторинг банковской системы

В [23] приводится пример блистательного применения самоорганизующихся карт Кохонена (SOM — Self-Organizing Map) [24] для исследования банковской системы России в 1999 — 2000 гг. В основе мониторинга лежит рейтинговая оценка на основе автоматического выполнения одной процедуры: по многомерному вектору параметров банков<sup>1</sup> на экране компьютера высвечивает-

1) Обращается внимание на то, что нейросетевые технологии позволяют строить наглядные функции многих переменных, как бы преобразуя многомерное пространство в одно-, двух- или трехмерное.

ся результат кластеризации — отнесение банка к различным категориям преуспевания. Для каждого отдельно взятого исследования различных факторов необходимо строить свои SOM. Прогноз возможен лишь на основе анализа временного ряда оценок SOM. Новые SOM необходимы и для продления цепочки выводов, с подключением данных извне, например 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 анализ с прогнозом, оперативно учитывать складывающуюся ситуацию и вводить в рассмотрение новые факторы и опыт экспертов. Следует согласиться с тем, что мозгу все это подвластно, и мы вновь обращаемся к его структурам, предлагая проект программных средств системы мониторинга.

**Структура нейросети и способы обучения.** Логические функции, лежащие в основе мониторинга, преимущественно основаны на конъюнкции логических значений переменных, отображающих диапазоны изменения параметров или показателей банков. В общем виде такая логическая функция имеет вид:

$$\bigwedge_{i=1}^n \left\{ \begin{array}{l} \text{значение } i\text{-го показателя банка принадлежит диапазону} \\ [a, b] \end{array} \right\} \rightarrow \{\text{указатель значения рейтинга}\}$$

В [23] представлены следующие показатели:

- собственный капитал;
- сальдированные активы;
- ликвидные активы;
- обязательства до востребования;
- вклады населения;
- коэффициент ликвидности;
- бюджетные средства.

Можно расширить систему показателей:

- объем инвестиций в эпоху бурно развивающейся экономики;
- 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 фонд;
- участие в программах ЮНЕСКО и т.д.

Такой простой вид логической функции при переходе в область действительных переменных говорит о достаточности однослойной нейросети (рис. 8.1), содержащей входной слой рецепторов и выходной слой, на котором формируются результаты мониторинга.

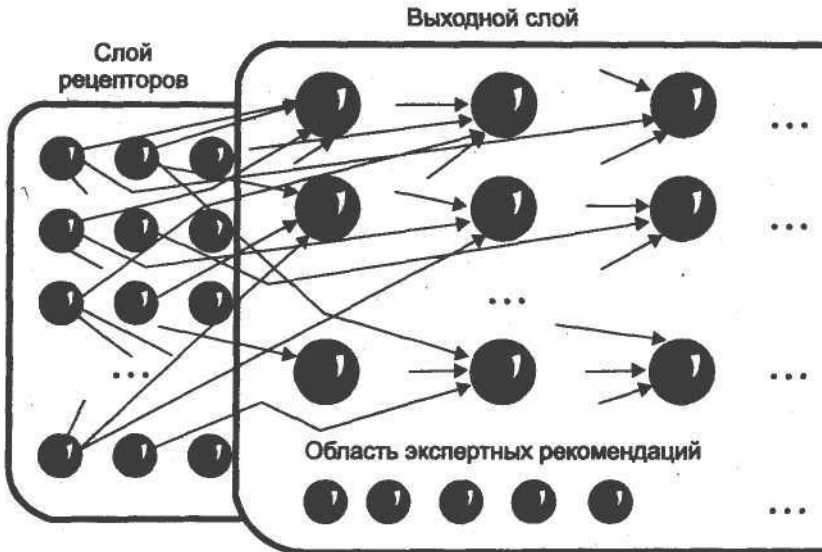


Рис. 8.1. Структура нейросети для банковского мониторинга

При построении входного слоя необходимо учитывать не только текущие показатели, но и динамику изменения 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 — построение базы знаний, который соответствует концепции создания нейросети «под задачу»: непосредственное введение связей оператором-ис-

следователем «вручную» — от рецепторов к нейронам выходного слоя в соответствии с причинно-следственными связями. Тем самым сеть создается уже обученной.

Тогда передаточная функция тоже будет простейшей и основанной на суммировании величин возбуждения на входе нейрона, умноженных на вес связи:

$$V := k \sum_j \omega_j V_j;$$
$$V_j = \begin{cases} V, & \text{если } V > h, \\ 0 & \text{— в противном случае.} \end{cases}$$

Задание веса связей га по сравнению с грубым заданием всех весов, равных единице, целесообразнее в связи с возможным желанием оператора или эксперта в разной степени учитывать влияние различных показателей.

Порог  $h$  отсекает заведомо неприемлемые выводы, упрощая дальнейшую обработку (например, нахождение среднего). Коэффициент приведения  $k$  обусловлен следующими соображениями. Максимальное значение  $V$  может достигать  $n$ . Для того чтобы значение рейтинга находилось в некотором приемлемом диапазоне, например в  $[0, 1]$ , значения возбуждения надо преобразовать, положив  $k = 1/n$ .

Принятые выше допущения позволяют оперативно вводить изменения и уточнения оператором — экспертом — пользователем, развивать сеть, вводя новые факторы и учитывая опыт. Для этого оператору достаточно, щелкнув мышью, выделить рецептор, а затем нейрон выходного слоя и связь установлена! Осталось только приблизительно назначить вес введенной связи из диапазона  $(0, 1]$ .

Здесь следует сделать **Очень Важное Замечание (ОВЗ)**, касательно всего материала книги и предназначенное очень внимательному читателю.

Ранее, рассматривая обучение, мы четко классифицировали исходные эталонные ситуации, принимая достоверность каждого компонента, равной единице. Проводя затем трассировку и прокладывая динамические пути возбуждения, мы также полагали веса связей, равными единице (или некоторому максимальному постоянному значению). Но ведь учитель сразу может полу-

чить дополнительную степень свободы, принимая во внимание факторы в той степени и с теми весами, которые он задаст! Сделаем допущение, что разные факторы в разной степени влияют на результат, и такое влияние заложим на этапе обучения принудительно.

Например, известно, что накануне войны население в огромном количестве закупает мыло, спички и соль. Значит, наблюдая за этим фактором, можно прогнозировать скорое начало войны. Создавая нейросеть для анализа 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, соответствующего заявлению «Скоро война!».

Тем не менее интенсивная закупка мыла, спичек и соли необходимо, но *не такое уж* достаточное условие наступления войны. Оно может свидетельствовать, например, о бурном возрождении туризма в район Главного хребта Кавказа. В словах «не такое уж» заключается смысл нечеткой логики [5], позволяющей учитывать не непреложность события, не булеву переменную «да — нет», а некоторое промежуточное, неопределенное, взвешенное состояние типа "*влияет, но не так уж, прямо, что обязательно...*". Поэтому связи (все или некоторые), исходящие из данного (данных) рецептора, положим равными некоторой предполагаемой величине, меньшей единицы и корректируемой впоследствии, которая отражает влияние возбуждения рецептора на вывод.

Таким образом, одновременная закупка мыла, соли и спичек учитывается дважды: уровень закупки будет отображен в степени возбуждения соответствующих рецепторов, а характер влияния закупки на вывод «Скоро война!» - с помощью весов синаптических связей.

Согласитесь, что при построении одноуровневых сетей такой подход напрашивается сам собой и реализуется предельно просто.

**Структура экрана рецепторов.** Экран рецепторов показан на рис. 8.2. Основную часть его составляет «окно прокрутки», в котором можно просматривать и задавать состояние рецепторного слоя, несомненно, не способного поместиться на статическом экране.



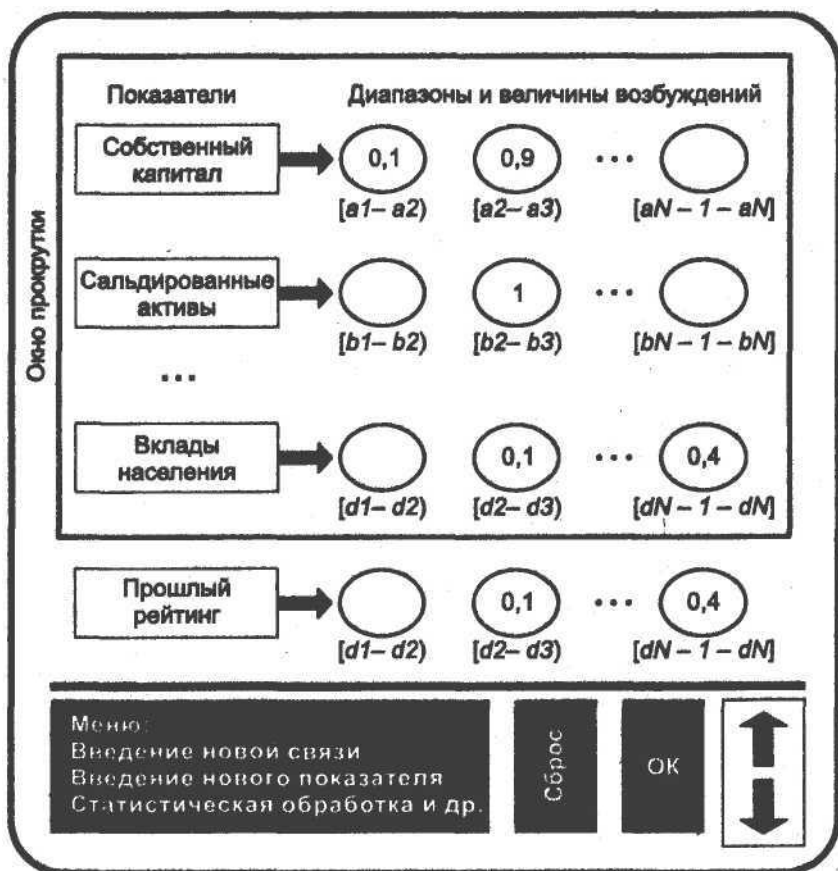


Рис. 8.2. Экран рецепторов

В «окне прокрутки» указаны показатели и их оценочные значения в диапазоне  $[0, 1]$  для соответствующих рецепторов. Это вероятностные значения, основанные на достоверности, интуиции, экспертных оценках. Оценки предполагают 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 0,6, 0,2 и 0,2 рецепторов, соответствующих диапазонам  $[20 - 25]$ ,  $[30 - 35]$ ,  $[40 - 45]$ .

На экране отображены статически задаваемые показатели, такие, как рейтинг в результате прошлых измерений, выборочные ранее найденные показатели, а также показатели политической, социальной и экономической конъюнктуры. (Их обилие и развитие могут все-таки потребовать прокрутки.)

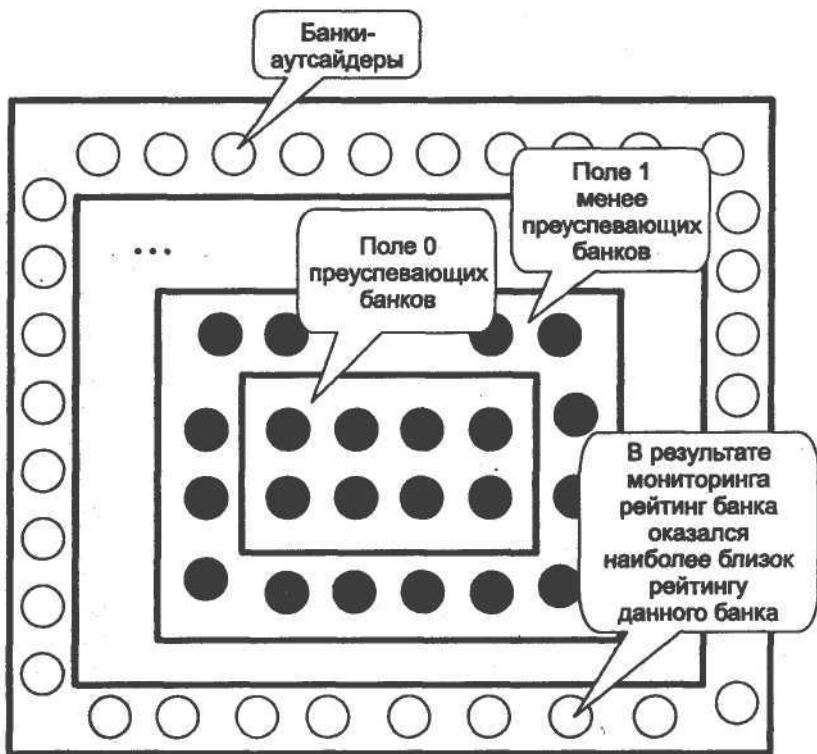
Следует также отобразить управление прокруткой и меню основных действий:

- 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.

**Структура экрана выходного слоя.** Экран выходного слоя (рис. 8.3) отображает систему концентрических (вложенных) прямоугольников или других плоских фигур, отражающих распространение рейтинга по убыванию. В центре экрана яркими точками отмечены самые преуспевающие банки или предполагаемые идеальные образы. Каждому элементу экрана жестко соответствует нейрон выходного слоя. В результате мониторинга может максимально возбудиться нейрон, соответствующий эталону, однако скорее всего высветится точка экрана, не совпадающая ни с каким эталоном, являющаяся промежуточной или усредненной.

Несомненно, следует предусмотреть меню для операции усредненной оценки рейтинга, демонстрации категории преуспевания, выдачи сигналов предупреждения, текстов заключений, рекомендуемых стратегий развития, сохранения данных для дальнейшего развития и т.д.

**Обучение нейросети.** Для обучения нейросети на основе экспертных оценок следует задать диапазоны допустимых параметров, позволяющие 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, можно получить несколько идеальных представителей. Соответствующие им нейроны, т.е. элементы экрана выходного слоя, выделяют произвольно, рассредоточивая по области экрана. Желательно, чтобы эталоны с большим рейтингом располагались ближе к центру.



Ряс. 8.3. Эcran выходного слоя

Далее переходят к подобному же заполнению охватывающего прямоугольника, на основе следующей рейтинговой категории и т.д. до банков-аутсайдеров.

Для проведения подобной работы экспертами предварительно формируется таблица (табл. 8.1).

Нейронам, отображающим банки, на экране соответствуют величины их возбуждения — рейтинги.

**Методика мониторинга.** Обученная система, которая поступает в распоряжение пользователя после высококвалифицированной экспертизы экономистов и политиков, готова к использованию в рамках **CASE-технологии**<sup>1</sup>.

<sup>1</sup> CASE — Computer Aided Software Engineering.

Таблица 8.1

## Экспертные оценки для обучения нейросети

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	
...	...	...	...	...	
$a_{n1}$	$a_{n2}$	...	$a_{nn}$	0,7	Банки-аутсайдеры
...	...	...	...	...	
$a_{n+r,1}$	$a_{n+r,2}$	...	$a_{n+r,n}$	0,1	

При этом 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 т.д.

Предположим, пользователь исследует ситуацию, сложившуюся вокруг банка "Инвест-Туда-и-Обратно". Естественно, он не располагает сколько-нибудь удовлетворительной информацией о целесообразности собственных вложений и поэтому приступает к скрупулезному сбору данных, в результате чего получает приблизительные, вероятные, разноречивые характеристики для моделирования.

С помощью экрана 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. Такие показатели, как рейтинг в прошлом и миграция, пока неизвестны, но полученный результат предполагается использовать в дальнейшем.

После ввода данных на экране выходного слоя яркая точка вблизи области аутсайдеров красноречиво свидетельствует о за-

щите гражданского права ненасильственного выбора решения о целесообразности вложения праведно накопленного капитала.

Координаты этой точки на экране определяются по известной формуле нахождения среднего по координатам высветившихся нейронов тех банков, которым близок контролируемый банк, и по величинам их возбуждения. Но по этим же формулам на основе рейтингов высветившихся банков находится рейтинг исследуемого банка!

Пользователь может принять решение о дополнении базы знаний и, следовательно, нейросети информацией о новом банке, что целесообразно, если совет экспертов подверг существенной критике получившийся результат и указывает тем самым на ошибку нейросети. Достаточно только воспользоваться опцией «Дополнить», в результате выполнения которой инициируется диалог компьютера с пользователем:

- Вы хотите изменить рейтинг? — Да.
- Новое значение рейтинга? — ...
- Сохранить!

Тогда нейрон выходного слоя с найденными координатами ставится в соответствие новому банку. Формируются 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 таким же образом, как список пристрелянных установок артиллерийской батареи после поражения очередной цели.

Однако значительное принудительное изменение рейтинга может потребовать перемещения высветившейся точки в область банков с соответствующим уровнем рейтинга, т.е. необходимо за данным банком закрепить другой нейрон выходного слоя, в другой области экрана. Это также устанавливается в результате диалога компьютера с пользователем.

**Корректировка и развитие.** Выше мы уже упоминали о необходимости и возможности постоянного уточнения и развития нейросети. Можно изменять представление о «продвинутости» банка-эталона (реального или идеального) и дополнять базу знаний, т.е. данную нейросеть. Можно корректировать веса связей как меры влияния отдельных показателей на выходной результат. Можно вводить новые показатели с их весами, рассматривать новые решения и устанавливать степень влияния на них тех же или

новых показателей. Можно приспособить нейросеть для решения смежных задач с учетом влияния отдельных показателей на миграцию банков (переход с одного рейтингового уровня на другой) и т.д.

Наконец, можно, приобретя данный программный продукт с дружественным интерфейсом и прекрасным сервисом, с развитым набором функций преобразования нейросети, переделать ее для совершенно другой задачи, например для увлекательной игры в железнодорожную рулетку, на которой мы намерены остановиться ниже.

В заключение отметим, что в экономике и бизнесе, а также в управлении сложными объектами преобладают системы принятия решений, где каждая ситуация образуется на основе неизменного числа факторов. Каждый фактор представлен вариантом или значением из исчерпывающего множества, т.е. каждая ситуация представляется конъюнкцией, в которой обязательно участвуют высказывания относительно всех факторов, по которым формируется нейросеть. Тогда все конъюнкции (ситуации) имеют одинаковое число высказываний. Если в этом случае две отличные друг от друга ситуации приводят к разным решениям, соответствующая нейросеть является *совершенной*. Привлекательность таких нейросетей заключается в их сводимости к однослойным. Если провести «размножение» решений (см. подразд. 5.2), то получим совершенную нейросеть (без обратных связей).

К построению совершенной нейросети можно свести задачу настоящего раздела, подразд. 6.2, а также, например, задачу оценки странового риска и др.

# 9 ПРИМЕРЫ ПРИМЕНЕНИЯ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

## 9.1 Проектирование игровой системы

**Построение обученной нейросети.** Рассмотрим увлекательную детскую игру «железнодорожная рулетка», основанную на известной задаче о встрече. Помните: «Из пунктов *A* и *B* двигаются навстречу друг другу...»?

Начальник станции Кукуевка (ведущий) 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. Проблема 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 с приветственным флажком, либо линейного с подстилочной соломкой. Ситуация усложняется помехами при передаче данных, в связи с чем достоверность сообщения о скорости паровозов меньше единицы. Кроме того, необходимо учесть ожидаемые денежные затраты на единовременную добавку к пенсии линейных.

Начальник станции Кукуевка хочет добиться определенности по принципу «если — то», а в случае недостоверных данных — «на что это похоже, и что делать?».

Ясно, что без элементов искусственного интеллекта не обойтись. Необходимо что-то похожее на табличный метод, но с автоматической интерполяцией, что-то связанное с ассоциативным мышлением...

Представим себе все варианты будущего поведения нашей системы принятия решений для ее обучения. Графическая структу-

ра логического функционирования создаваемой системы приведена на рис. 9.2.

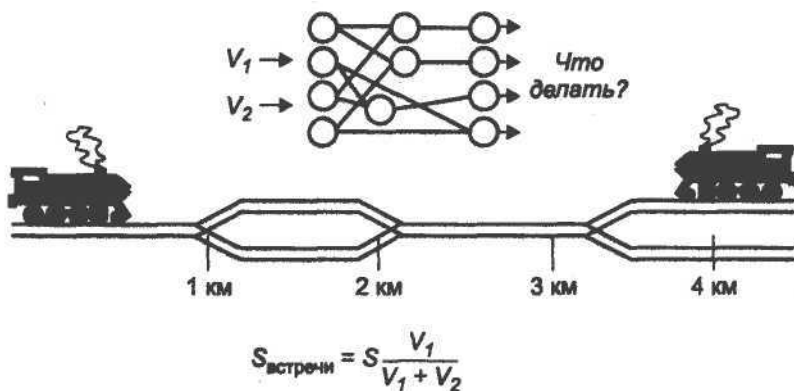


Рис. 9.1. Схема игры

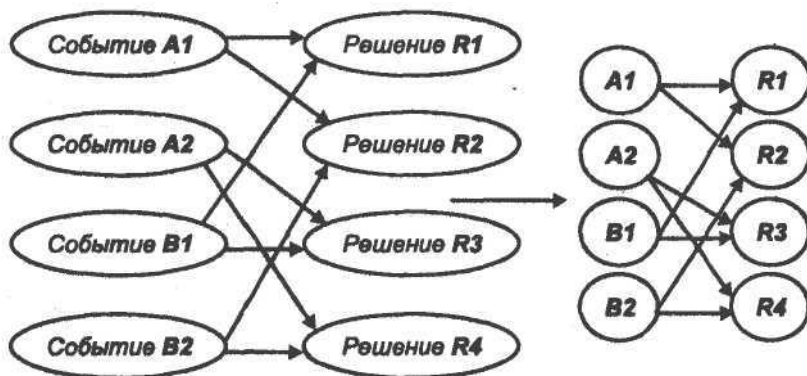


Рис. 9.2. Система принятия решений

1. Скорость кукуевского паровоза — 60 км/ч (событие  $A \cup$ ). Скорость пырловского паровоза — 60 км/ч (событие  $B \setminus$ ). Одновременное выполнение этих событий обозначим  $A1 \& B1$ . Тогда точка встречи находится как раз посередине перегона, что скорее всего требует помощи линейного с соложкой. Но возможен и другой исход за счет неточного определения скоростей. Тогда на вся-



кий случай потребуется линейный с флажком. Решение, согласно которому оба линейных отправляются на границу второго и третьего километров, обозначим  $R1$ . С ним связаны расходы на единовременное пособие  $M1$ .

2. Скорость кукуевского паровоза — 60 км/ч (событие  $Ax$ ), скорость пырловского — 80 км/ч (событие  $B2$ ). Выполняется условие  $Ax \& B2$ . Тогда точка встречи находится на втором километре пути, и, следовательно, требуется решение  $R2$ : «Отправить линейного с флажком на второй километр!» В активе указанного линейного появляется сумма  $M2$  условных единиц.

3. Скорость кукуевского паровоза - 80 км/ч (событие  $A2$ ), скорость пырловского — 60 км/ч (событие  $B1$ ). Выполняется условие  $A2 \& B1$ . Тогда точка встречи находится на третьем километре пути, что требует сочувственного вмешательства линейного с соломкой (решение  $R3$ ), с оплатой труда в  $M3$  условных единиц.

4. Скорость кукуевского и пырловского паровозов — 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 так же, как, по-видимому, на логическом уровне действуют структуры нашего мозга.

Представим себе, что на месте каждого овала (далее - кружочка, см. рис. 9.2) действует нейроподобный элемент или просто нейрон. Нейроны-рецепторы приходят в возбужденное состояние (подобно сетчатке глаза) в соответствии с той ситуацией, которая задана на входе системы. Рассмотрим, например, ситуацию  $A1 \& B2$ . Положим величины возбуждения нейронов  $A1$  и  $B2$  равными единице и запишем  $V_{A1} = V_{B2} \equiv 1$ . При этом величины возбуждения нейронов  $A2$  и  $B1$  остались равными нулю.

Подчеркнем тот факт, что возбуждение нейронов-рецепторов осуществляется в результате ввода информации.

Для «принимающих» возбуждение нейронов (направление распространения сигналов возбуждения указано стрелками) в результате выполнения *передаточной функции* формируется величина возбуждения  $V$ . Пусть в нашем случае эта функция имеет вид

$$V = \xi(\sum_i V_i - h),$$

где  $i$  - индекс нейрона, передающего свое возбуждение,  
 $h$  — порог.

Следует учесть, что функция  $\xi(x)$  заменяет отрицательную величину нулем, т.е.

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

Положим  $A = 1$  и рассчитаем величины возбуждения нейронов *выходного слоя*  $R \setminus - R4$  для ситуации  $A \setminus \& B2$ :

$$\begin{aligned} V_{R1}: & 1 + 0 - 1 = 0, & V_{R1} &= 0; \\ V_{R2}: & 1 + 1 - 1 = 1, & V_{R2} &= 1; \\ V_{R3}: & 0 + 0 - 1 = -1, & V_{R3} &= 0; \\ V_{R4}: & 0 + 1 - 1 = 0, & V_{R4} &= 0. \end{aligned}$$

Таким образом, при точном знании ситуации получаем предусмотренное решение. Проверим, что наша сеть так же правильно работает и по остальным *эталонам*, по которым мы ее *обучили*, наметив направления распространения возбуждений 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 обобщить тип исходных данных, допустив рассмотрение значений, интерпретируемых как достоверность? Или каких-то других взаимных оценок, используемых людьми, не сведущими в теории вероятности и не знакомыми с понятием «полное множество событий»?

Например, в результате искажения информации начальник станции Кукуевка пребывает в раздумье: скорость пырловского паровоза равна не то 60 км/ч, не то 80 км/ч, но скорее всего 60 км/ч. Он по наитию набирает:  $A1 = 1$ ,  $B1 = 0,7$ ,  $B2 = 0,4$ . На какую ситуацию это похоже и какое решение принять? Считаем:

$$\begin{aligned} V_{R1}: & 1 + 0,7 - 1 = 0,7, & V_{R1} &= 0,7; \\ V_{R2}: & 1 + 0,4 - 1 = 0,4, & V_{R2} &= 0,4; \\ V_{R3}: & 0 + 0,7 - 1 = -0,3, & V_{R3} &= 0; \\ V_{R4}: & 0 + 0,4 - 1 = -0,6, & V_{R4} &= 0. \end{aligned}$$

Мы видим, что максимальной величины возбуждения достиг нейрон  $R1$ , определивший главное решение. Но мы вправе учесть и решение  $R2$  с меньшим приоритетом, дав линейному с флажком дополнительные указания. По известной формуле математи-

ческого ожидания оценим, на сколько облегчится карман начальника станции Кукуевка:

$$M = \frac{M1 \cdot 0,7 + M2 \cdot 0,4 + M3 \cdot 0 + M4 \cdot 0}{0,7 + 0,4 + 0 + 0}$$

Таким образом, сформирована уже *обученная* нейросеть.

А что если объединить решения  $R1$  и  $R4$ , отличающиеся только скоростью передвижения линейных? Следуя рассмотренному выше принципу, мы получили бы сеть на рис. 9.3.

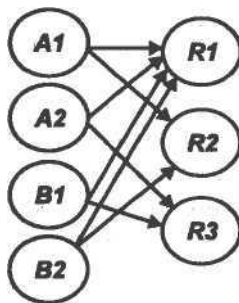


Рис. 9.3. Однослойная нейросеть, непригодная для обучения

Легко видеть, что максимальное возбуждение, соответствующее решению /И, достигается всегда, когда задана ситуация, требующая максимального возбуждения только лишь нейронов  $R2$  или  $R3$ . Сеть дает побочный эффект, что требует корректировки решения. Она заключается в том, что если максимально и одинаково возбуждись два нейрона выходного слоя и один из них  $R1$ , то верное решение соответствует второму. Если максимально возбуждился только нейрон  $R1$ , то он представляет правильное решение.

В данном случае происходит коллизия при замене операций конъюнкции  $\wedge$  и дизъюнкции  $\vee$  одной операцией  $\&$ . Ведь логический предикат возбуждения решения  $R1$  имеет вид

$$(A1 \wedge B1) \vee (A2 \wedge B2) \rightarrow R1.$$

Благодаря этому справедлива гипотеза о минимальной длине статической цепочки в разд. 2.9.

Тогда легко провести коррекцию, построив обученную сеть так, как показано на рис. 9.4, введя так называемые скрытые нейроны 1 и 2.

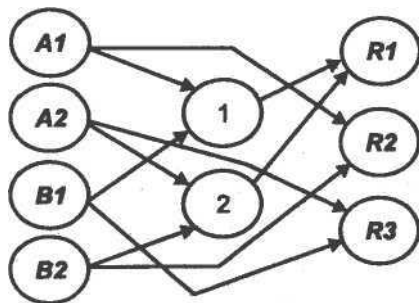


Рис.9.4. Обученная нейросеть

**Выбор нейросети, обучение, трассировка.** Мы построили нейросеть, пользуясь приемами, известными специалистам-схемотехникам, разрабатывающим устройства компьютера. Мы соединили элементы связями — «проводочками», произведя трассировку для правильного распространения сигнала в соответствии с желаемой функцией. Как неоднократно указывалось выше, все это относится к уже обученной сети. Причем из рассмотрения полностью исключен тот параметр, настройка которого позволяет обучить сеть, в частности, проложить нужные "проводочки" и перекусить ненужные.

Нейрон в данном случае только выполняет передаточную функцию, один из видов которой и был рассмотрен. 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, связанных с ним нейронов. Дендрит воспринимает сигнал, пропорциональный своему весу. Таким образом и формируется 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;

$i$  - индекс использованного дендрита;

$\omega_i$  — вес синаптической связи.

В результате весь смысл построения и обучения нейросети заключается в том, что синаптические *веса регулируются*, обуславливая пути прохождения возбуждений в сети.

В частности, представив некоторую абстрактную сеть и построив на ее основе сеть для игры в рулетку, положим некоторые веса связей, равными единице (наличие "проводочков"), а некоторые (или все оставшиеся) — равными нулю (отсутствие «проводочков»). Но ведь допустимы и некоторые промежуточные значения весов, хотя для практических целей наш подход оправдан.

Подойдем к построению нейросети для игры в железнодорожную рулетку иначе. Ранее сеть создавалась по известным условиям игры, теперь пусть нам задана нейросеть, а мы обучим ее для игры в рулетку.

Изобразим некоторый ориентированный ациклический граф (рис.9.5) и придадим ему смысл нейросети, поставив в соответствие его вершинам — нейронам (кроме предполагаемых рецепторов) все ту же передаточную функцию. Вот только как заставить сеть так реагировать на очевидные эталоны, чтобы максимального возбуждения достигали нейроны выходного слоя, соответствующие решениям? Для этого необходимо, полагая первоначально все веса нулевыми (или минимальными), увеличить некоторые из них до максимального значения или до единицы. Проще всего именно так и действовать: сначала все веса нулевые ("проводочки" есть, но их сопротивление чрезвычайно высоко), затем выбранные нами веса положим равными единице. Выше-сказанное равносильно тому, что какие-то "проводочки" оставлены, а какие-то перекушены. Описанное действие по обучению нейросети мы называем *трассировкой*.

Продемонстрируем алгоритм трассировки, введя упрощения по сравнению с рассмотренным в разд. 3.

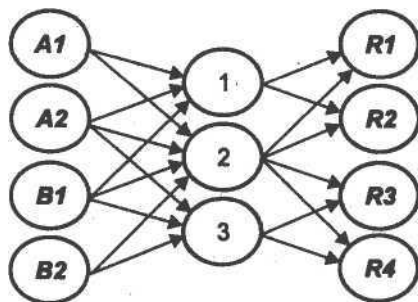


Рис. 9.5. Нейросеть, предложенная для обучения

Компьютерная обработка нейросети значительно облегчается, если сеть представлена матрицей следования  $S$  (рис. 9.6, *a*), где наличие связи обозначается ее весом.

1. Произведем трассировку возбуждений нейронов  $[A]$ ,

Исключим из матрицы  $S$  строки и столбцы, соответствующие не интересующим нас нейронам входного и выходного слоев. Матрица примет вид  $S_1$ , представленный на рис. 9.6, *б*. Про моделируем прохождение возбуждения следующим образом.

Присвоим нейронам, соответствующим нулевым строкам — входам матрицы  $S_1$ , признак «возбужден». Выделим столбцы, соответствующие этим же 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 на пересечении столбца и строки, соответствующей нейрону 1, записываем  $m_1 = 2$ . Таким образом обеспечивается возможность повторного использования нейронов при получении других решений. Исключаем из матрицы  $S_1$  строки и столбцы тех нейронов, возбуждение которых использовано. Нейрону 1 присваиваем признак «возбужден». Матрица  $S_1$  принимает вид, приведенный на рис. 9.6, *в*.

<b>A1</b>	■													
<b>A2</b>		■												
<b>B1</b>			■											
<b>B2</b>				■										
<b>1</b>	0	0	0		■									
<b>2</b>	0	0	0	0		■								
<b>3</b>		0	0	0			■							
<b>R1</b>					0	0		■						
<b>R2</b>					0	0			■					
<b>R3</b>							0	0		■				
<b>R4</b>							0	0			■			

**a**

<b>A1</b>	■													
<b>B1</b>		■												
<b>1</b>	0	0	■											
<b>2</b>	0	0		■										
<b>3</b>		0			■									
<b>R1</b>			0	0		■								

**Возб.**  
**Возб.**

**b**

<b>1</b>	■													
<b>2</b>		■												
<b>3</b>			■											
<b>R1</b>	0	0		■										

**Возб.**

**в**

<b>1</b>	■													
<b>R1</b>	0													

**Возб.**

**г**

<b>A1</b>	■													
<b>B2</b>		■												
<b>1</b>	1		■											
<b>2</b>	0	0		■										
<b>3</b>		0			■									
<b>R2</b>			0	0		■								

$m_1 = 2$

**д**

<b>2</b>	■													
<b>R2</b>	0													

**ж**

<b>A1</b>	■													
<b>B2</b>		■												
<b>2</b>	0	0	■											
<b>3</b>		0		■										
<b>R2</b>			0		■									

**Возб.**  
**Возб.**

**е**

<b>A2</b>	■													
<b>B2</b>		■												
<b>R4</b>			■											

**з**

**Рис. 9.6.** Матрица следования для игры:

- a** — общий вид; **b** - для трассировки первого решения;
- в** — первый шаг преобразования; **г** — второй шаг преобразования;
- д** - для трассировки второго решения; **е** - первый шаг преобразования матрицы следования для обучения второму эталону;
- ж** — второй шаг преобразования матрицы следования для обучения второму эталону; **з** — для трассировки четвертого решения

Исключаем из матрицы  $S_1$  строки и столбцы, соответствующие входам без признака «возбужден». Эти строки соответствуют нейронам 2 и 3. Вид полученной матрицы  $S_1$  представлен на рис. 9.6, г.

Повторяем описанные выше действия, что приводит к замене единицей единственного нуля.

Не приводя промежуточного рисунка, отметим, что мы подтвердили высоким (единичным) весом 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  $A2, B\setminus, R\setminus, R3, R4$  (рис. 9.6, д).

Строка, соответствующая нейрону 1, содержит одну единицу, при том, что  $m_1=2$ . Исключаем из рассмотрения и этот нейрон, как негодный для повторного использования. Вид матрицы  $S_2$  приведен на рис. 9.6, е.

Присваиваем строкам — входам матрицы  $S_2$  признак «возбужден». В совокупности соответствующих им столбцов находим строку, содержащую максимальное число нулей. Такая строка соответствует нейрону 2. Заменяем в ней нули единицами, что отмечаем в матрице  $S$ . Полагаем  $m_2 = 2$ . Присваиваем нейрону 2 признак «возбужден», а нейроны  $A1$  и  $B2$  исключаем из рассмотрения. Среди оставшихся строк исключаем «пустую», соответствующую нейрону 3 и не обладающую признаком «возбужден». Исключаем и соответствующий столбец. Матрица  $S_2$  показана на рис. 9.6, ж.

Повторение описанных 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 алгоритм построений, легко найдем связи с единичными весами  $A2 \rightarrow 3$ ,  $B1 \rightarrow 3$ ,  $3 \rightarrow R3$  в результате трассировки  $[A2, B1] \rightarrow R3$ .

4. Трассировка последнего пути возбуждения  $[A2, B2] \rightarrow R4$  обладает некоторыми особенностями, обусловленными отсутствием связей «через слой», которые видны на рис. 9.4.

Сформируем матрицу  $S_4$ , в которой не отражены те нейроны, в строках которых число единиц меньше соответствующего значения  $m$  (рис. 9.6, з). Но ведь в ней вообще нет никаких связей!



Придется их ввести, да еще с единичными весами согласно условию задачи. Очевидно, что больше трех эталонов эта сеть воспринять не способна. Таким образом, мы вводим дополнительные связи с единичными весами  $A1 \rightarrow R4$ ,  $B2 \rightarrow R4$ .

Обученная сеть, где выделены связи с единичными весами, окончательно представлена на рис. 9.7. Лишние "проводочки" можно перекусить.

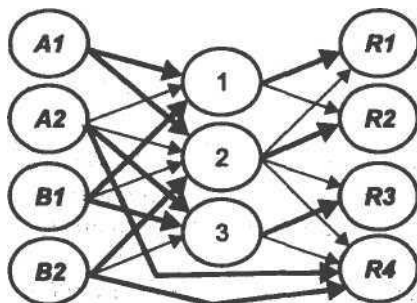


Рис. 9.7. Полностью обученная нейросеть

Вычислим с помощью нашей передаточной функции величины возбуждения нейронов для 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  $A2 \& B1$ . Для случая  $A2 \& B2$  имеем:  $V_1 = 0$ ,  $V_2 = 0$ ,  $V_3 = 0$ ,  $V_{R1} = 0$ ,  $V_{R2} = 0$ ,  $V_{R3} = 0$ ,  $V_{R4} = 1$ . Таким образом, построенная нейросеть распознает единственную ситуацию.

Отметим, что в процессе распространения по сети возбуждение «гаснет», не доходя до выходного слоя. Значит, следует уменьшить пороги, но тогда величины возбуждений нейронов выходного слоя становятся различными для разных ситуаций, ибо различна длина путей возбуждения. Это затрудняет корректное участие нейронов в тех случаях, когда данный выходной слой является входным для следующего логического уровня сети. Следовательно, наша передаточная функция не годится для выбранной структуры нейросети.

Приведем еще одну 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  $A = 1$  сеть правильно распознает все эталонные ситуации, обеспечивая равную величину возбуждения нейронов выходного слоя. Так, в 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 правильно реагирует на недостоверные ситуации и позволяет находить среднее.

А главное в том, что игра имеет массу перспектив развития. 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 агрессивен, а другой желает уклониться от столкновения? И т.д.

## 9.2 Служба безопасности

Охрана покоя семьи, противодействие взломщикам-грабителям актуальная задача нашего бурного времени. Мы справедливо предполагаем развитие рынка средств автоматического распознавания «свой — чужой» по мере объявленного роста благосостояния народных масс, находящего все большее число покупателей. Как же устроено такое средство охраны — предупреждения (рис. 9.8)?

Основной логической деятельностью устройства, конечно же, является нейросеть.

Представим себе многоуровневую нейросеть, первый логический уровень которой подбирает «словесный портрет» на основе признаков изображения, поступившего на его экран. В результате анализа, возможно, с двух точек — анфас и в профиль, установлено:



Рис. 9.8. Система распознавания «свой - чужой»

- волосы черные, шевелюра пышная, на 5 баллов;
- лоб высокий, залысины, поперечные морщины;
- брови густые, взлет, сросшиеся на переносице, № 8;
- нос орлиный, № 4 и т.д.

Конечно, баллы и номера не выставляются, они означают только, какой нейрон промежуточного выходного слоя возбудится максимально. Да и это нам знать ни к чему. Просто произойдет распределение возбуждений между нейронами промежуточного выходного слоя той части нейросети, что обучена распознаванию признаков. Возбуждение этих нейронов как нейронов входного слоя второй части нейросети, т.е. второго логического уровня, должно в результате обучения привести к существенному отличию «папы» от «слесаря дяди Юры», приходившего днем проверить исправный кран. Иными словами, привести к идентификации кто это: «свой» или «чужой»? Можно предусмотреть и особые приметы: чрезмерно красный нос, подозрительные красные пятна на лице и др. При формировании голосового ответа (задействован третий логический уровень) учитывается и эта дополнительная информация.

### 9.3 Парк фантасмагорий

Представим себе парк развлечений, в который заходит наша группа под руководством гида и, желая отдохнуть, устремляется в тенисто-прохладный уголок с причудливым ландшафтом. И вдруг из-за трепещущих кустов выползают монстры — крокодилы, вздымаются головы ящеров. Лохматый неандерталец ладит каменное орудие, подымается на свои куры ножки избушка. Гигантской головой оживает скала, покрытая нежными незабудками, деревья и кусты качают ветками. Наше замешательство и паника вызывают ответную реакцию: окружающие смеются, сердятся, радуются, угрожают.

Рассмотрим возможный принцип действия отдельного объекта (рис. 9.9), который контролирует сектор обзора. Электронно-оптическое «зрение» может быть расположено не обязательно в глазах, а упрятано в любой другой части тела, например на груди.



Рис. 9.9. Обзор экрана

Весь сектор обзора разбит на элементарные сегменты, образованные телесными углами в  $1 - 2^\circ$ . Первоначально должна быть предусмотрена реакция только на преобладающий в сегменте цвет. Например, красный цвет должен запускать комплекс программ ярости и повторный обзор строки для усиления ярости при последующем раздражении.

Желтый цвет может вызывать кокетство и эротическое возбуждение, зеленый — умиротворение, коричневый — ритмичный шаг и покачивание головой, голубой — благостное умиление с возведенными к небу глазами и т.д.

Обзор сегментов совершается по строкам, с частотой, согласующейся со скоростью обработки компьютером. Последняя, в свою очередь, обусловлена прохождением входного сигнала через нейросеть и различной степенью возбуждения нейронов выходного слоя, запуском программ движения на базе этих нейронов с параметрами, определяемыми величинами возбуждения и инерционностью такой сложной системы.

Итак, из каких элементарных движений складывается поведение объекта? Расширение — сужение глаз, открывание — закрывание рта, улыбка, покачивание головой, поднятие — опускание рук, угрожающий шаг (если вес позволяет), вертикальное — горизонтальное помахивание хвостом, призывное движение таза и др.

Как упоминалось ранее, в одном такте могут запускаться несколько программ, инициирующих движения с текущего состояния. В результате имитируется суммарная реакция на ситуацию в секторе обзора и ее изменение, обеспечивается непредсказуемость, неожиданность и восторг публики. А если представить себе, что такой реакцией наделены деревья, скалы, вода и ветры, то какой фантазмагорический получится фейерверк!..

## 9.4 Компьютерный человек КОМПИ

Известно, какое значение компьютерным играм придает самый богатый человек планеты Б. Гейтс. Вволю наигравшись в холодную войну, мы тоже можем себе позволить подумать о радости и развлечениях, а также о бизнесе!..

Пора объединить все разнообразие направлений *творческого* применения компьютера как средства, инструмента и объекта *искусства*. Сказанное может быть материализовано виртуальной структурой под названием *Computer Art Studio* и отражать направления: трехмерная и плоская анимация, графика и живопись; туристические объекты и исторические реконструкции; *нейро-*

компьютерные игры и анимация; литературный и музыкальный дизайн; фантастика и др.

Вот так благодаря логической цепочке «интеллект → моделирование → искусство» нейросетевые технологии должны занять подобающее им место в рядах Великого Искусства...

Представим себе школьника, который не успев протереть заспанные глаза, включает компьютер. На экране появляется милая, мордашка КОМПИ, популярного (нам пока неизвестного) героя (рис. 9.10). Наш школьник не очень твердо, но вполне узнаваемо, рисует ананас. КОМПИ приходит в восторг. Все мимические мышцы компьютерной мордашки приходят в движение.



Рис. 9.10. Компьютерная игра КОМПИ

Тогда подросток стирает рисунок, и КОМПИ приходит в состояние недоуменного ожидания. И вот после некоторых размышлений художник изображает... велосипед.

«Что это такое?» — спрашивает КОМПИ.

«Это — *велосипед*, на нем *катаются*», — следует ответ.

Дальнейшая пауза свидетельствует о том, что происходит обучение нейросети. Ситуация, отразившаяся на нейронах входного слоя, образующих экран для рисования, передается нейрону выходного слоя (или коры). Этот нейрон будет теперь соответствовать «эталону» — велосипеду. Нейрон, в свою очередь, должен определить возбуждение нейрона, ведающего действием «кататься». Запускаются программы, имитирующие движения 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 производящей компании) возможностей игры дополнением специальных программ имитации красивого велосипеда (распознанного по корявому эталону) и всех действий КОМПИ при езде на нем.

Средствами трехмерной графики, например 3D Studio MAX, производя деформаций 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. Еще не установлено точно, может ли это MAX, однако ясно, что результат достигается средствами двумерной и трехмерной компьютерной графики, на основе управления деформацией объектов. Не исключено, что здесь потребуются разработка новых систем компьютерной графики, основанных на «мышечной» деформации.

Например, создав объект в плоской или трехмерной памяти, мы можем, дважды щелкнув мышью, задать точку начала и точку конца мышцы, зафиксировав ее в списке в качестве нового невидимого объекта. Сокращение и растягивание мышцы сопровождается вычислением адресов такой пересылки информации, при которой, видимо, отображается улыбка и другие движения.

Таким образом можно осуществлять программируемую анимацию. Ряд программ различных действий, их запуск в зависимости от решений нейросети формируют комплекс интересных игр и даже сценических постановок.

• Итак, развитие игры, выпуск новых модификаций, расширяющих возможности, может в конце концов привести к весьма

сложному и многообразному взаимодействию юноши, влюбленного в компьютер, и компьютерного человечка, живущего в нем.

Сетевые технологий, т.е. совместное использование многих компьютеров, обеспечивают широкие перспективы развития подобных игр. И это не только весело. Это серьезный и важный принцип управления, который может быть использован во многих производственных и зрелищных приложениях.

## 9.5 Графический диспетчер движения поездов

Составление расписания — оптимизационная задача (оптимизируются известные технико-экономические показатели) значительной трудоемкости, поэтому до сих пор диспетчер фактически работает вручную, хотя существуют некоторые эвристические алгоритмы.

Традиционно расписание движения поездов через перегон, контролируемый одним диспетчером, составляется графически, в виде схемы с подразумеваемой системой координат: горизонтально — время, вертикально — расстояние от одного конечного пункта до другого с указанием промежуточных расстояний. Тогда график движения каждого поезда будет представлен в виде наклонной линии с переменным углом наклона, с учетом остановок. Общий угол наклона линии соответствует направлению движения поезда. Несколько поездов, следующих через перегон, порождают сложную «картинку» их обслуживания согласно приоритетам, возможного схождения путей в один, длины поезда (с учетом длины путей станций), необходимой скорости следования, запретом остановок.

Диспетчер должен умело управлять поездами так, чтобы за время ожиданий одних поездов на пунктах перегона провести другие, выдержав приоритет: скорый, пассажирский, сквозной, сборный и т.д., по возможности избежав столкновений и остановок вне пунктов следования.

На рис. 9.11 показан график движения поездов на однопутном перегоне. Здесь кривая 1 соответствует скорому поезду, кри-



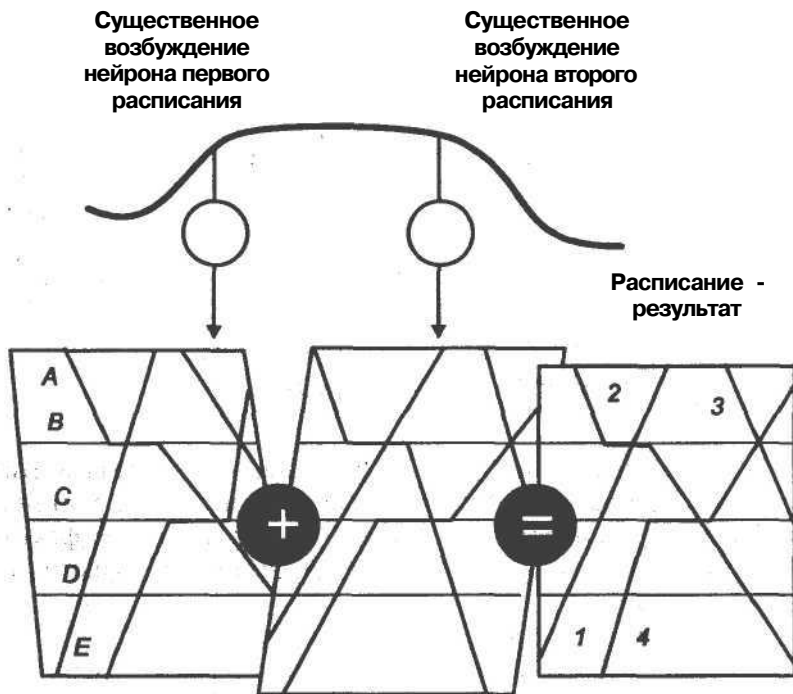


Рис. 9.11. Синтез расписания движения поездов

вые 2 и 3 — двум сквозным поездам, следующим в одном направлении из пункта А. Кривая 4 соответствует сборному поезду, следующему, как и поезд, в противоположном направлении из пункта В.

Можно представить себе мастерство, интуицию и опыт диспетчера, действительно демонстрирующего искусство!

Мы имеем дело с трудно формализуемой задачей и с необходимостью расширения интеллектуальных возможностей диспетчера. Конструируя нейросеть, необходимо выделить основные события, составляющие исходную ситуацию с четким ответом «да — нет», «1-0».

Как автоматизировать решение задачи для каждого конкретного перегона на основе богатого опыта диспетчера-эксперта?

Прежде всего надо определить понятие «ситуация», выделив и систематизировав все показатели и характеристики: макси-

мальное 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). Необходимо разработать способ отображения ситуации на входном слое нейросети.

Далее следует выделить эталонные ситуации и сопоставить им картинку, аналогично приведенным на рис. 9.11, т.е. каждой картинке-эталону нужно поставить в соответствие нейрон выходного слоя сети и проложить трассу - опорный путь возбуждения от эталона к соответствующему нейрону.

Нейросеть позволит нам перейти от статики экспертной системы (фактически таблицы) к динамике, позволяющей по любой случайной ситуации получать ответ на вопрос: «какая картинка в большей степени соответствует этой ситуации?».

Однако таким образом нельзя исключить аварии, т.е. на этом анализ ситуации не заканчивается. Мы не использовали всю информацию нейросети.

Во-первых, сравнение величины возбуждения нейрона выходного слоя (пусть и максимальной) по предъявленной ситуации с величиной возбуждения этого нейрона по эталону может свидетельствовать о достоверности полученного ответа. Если разница лежит в пределах экспериментально установленной точности, то ответ найден. Если же точность недостаточна, то можно построить (автоматически, конечно) приоритетный ряд, выделив нейроны выходного слоя с близкими (ближайшими) величинами возбуждения. Эти величины возбуждения нейронов образуют веса, которые необходимо учесть при использовании картинок, соответствующих этим нейронам.

Во-вторых, должна быть решена проблема формирования усредненной картинки по нескольким картинкам с их весами, где основные элементы сместятся в некоторое промежуточное положение. Только тщательное экспериментальное исследование даст ответ на вопрос о возможности и конкретной практической применимости такого решения проблемы.

Например, в исследуемом отрезке времени (см. рис. 9.11) поезд появляется на контролируемом перегоне. Возникают следующие вопросы:

1. Сколько поездов на обслуживании:

$$n = 1?$$

$$n = 2?$$

$$n = n_{\max} ?$$

2. Каков момент появления поезда:

$$i = 1?$$

$$i = 2?$$

$$t = t_{\max} ?$$

3. Каково направление движения:

*от пункта А ?*

*от пункта В ?*

4. Каков тип поезда:

*скорый?*

*товарный? и т.д.*

И так по всем поездам в планируемом отрезке времени. Таким образом, в зависимости от обстановки, которая ожидается на период планирования, составляется сложная усредненная картинка — график движения.

Однако мы говорим об ассоциативном мышлении, реализуемом нейросетью, о возможности нейросети интерполировать принимаемое решение. Что означает ее ответ: «На 60 из 100 это «картинка» А, но на 30 из 100 «картинка» В, а на 8 из 100 «картинка» С»? Непонятно, но наводит на мысль о возможности автоматического выполнения операций над «картинками». Однако получение «усредненной картинки» должно быть строго обосновано 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.

«Усредненная картинка» получается на основе зависимости

$$\begin{aligned} & \text{"Усредненная картинка"} = \\ & = \frac{0,6 \cdot \text{"картинкаА"} + 0,3 \cdot \text{"картинкаВ"} + 0,08 \cdot \text{"картинкаС"}}{0,6 + 0,3 + 0,08}, \end{aligned}$$

где «+» — условная операция наложения «картинок», а операция «х» использует вес 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» *A*, *B*, *C* для получения результата.

Кстати, на основе такой операции преобразования "картинок" получают среднестатистические образы населения стран и регионов, производят этническое прогнозирование и т.д.

В любом случае работа нейросетевого диспетчера должна сопровождаться работой анимационной модели, наглядно имитирующей движение поездов по сформированному графику и эффективно, в деталях воспроизводящей возможную катастрофу. Модель, в свою очередь, может служить средством обучения нейросети. А потому всегда должно быть предусмотрено, что окончательное принятие решения — за человеком.

## 9.6 Печать рукописи

Ясно, что восстановить рукописный текст способно только ассоциативное мышление, поэтому автоматизация воспроизведения написанного является актуальной задачей. В основе распознавания по признакам лежит все та же методика чистописания<sup>1</sup>.

В разд. 1 рассматривалось «признаковое» определение букв: *a* — кружочек с палочкой справа, заканчивающейся хвостиком вправо, *b* — кружочек с хвостиком сверху, *ц* — палочка с хвостиком вправо, к нему примыкает палочка, а к ней справа внизу странный специфический крючок и т.д.

Представим себе сканирование текста через «окно просмотра» (рис. 9.12) на входном слое нейросети. Рецепторы связаны с

<sup>1</sup> Применялось в России с 1740 по 1965 гг.

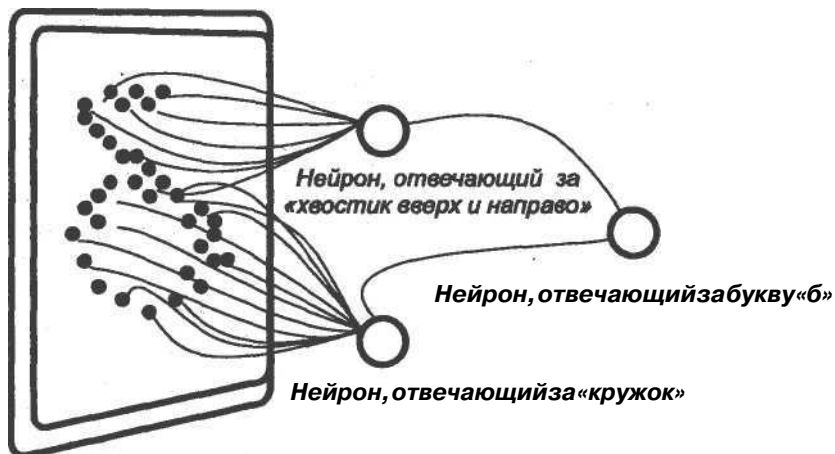


Рис. 9.12. Распознавание буквы по признакам

нейронами промежуточного выходного слоя, закрепленными за признаками. Соединение выполнено так, чтобы «захватить» букву, т.е. предусмотреть случайные, обусловленные почерком отклонения буквы от эталона и выявить ее отдельные признаки. Комбинации выявленных признаков, в виде сигналов возбуждения нейронов промежуточного выходного слоя, рассматриваемого в качестве входного слоя второго логического уровня, на выходном слое нейросети определяют распознаваемую букву.

Возможно, что при появлении очередной буквы, письма понадобится процедура разглядывания: увеличение — уменьшение, наклон влево — вправо, до тех пор, пока при некоторой фиксации образа не начнется 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, приобретает наклон влево или вправо так, чтобы получить максимальное соответствие по размеру и углу наклона с эталонами, подававшимися в процессе обучения. Впрочем, вышеуказанные манипуляции могут осуществляться динамически на основе проб в процессе распознавания буквы.

Для выявления всех признаков и их относительного положения «окно просмотра» необходимо разбить на сектора, определяющие понятия слева, справа, сверху, внизу, внизу справа и т.д.

Определенную трудность представляют пропуски и пробелы. Например, как отличить букву *c* от буквы *o*? Ведь может возбудиться «полный кружок» последней, только с меньшим значением величины возбуждения.

Необходимо наряду с позитивным изображением элементов букв анализировать их негативное изображение, тогда пустота в соответствии с эталоном будет возбуждать рецепторы, обретет активность. Наличие пустоты так же определяется посредством возбуждения нейронов, как и видимые линии.

Очевидно, что для понимания текста одного лишь распознавания букв недостаточно. Необходимо произвести распознавание слов, включив в работу контекст. Только таким комплексным методом восстанавливается смысл написанного, и это сулит неограниченные возможности развития и совершенствования нейросетевой технологии прочтения.

## 9.7 Защита информации

Мы знаем, насколько важна сегодня информация. Защита информации от несанкционированного доступа — идет ли речь о перевозке грузов или о финансовых коммерческих системах — первоочередная задача, актуальность которой все возрастает с развитием сетевых компьютерных и телекоммуникационных технологий. Высокие затраты на ее решение окупаются с лихвой.

Однако разнообразятся и средства «взломщика»: сегодня ему нет нужды ломать систему — он просто врывается в офис, и под дулом автомата Калашникова вы становитесь покорным исполнителем его желаний.

Впрочем, моральное состояние клиента выдает его чуткому серверу (рис. 9.13), на расстоянии выполняющему заявки в режиме «клиент — сервер». Наряду со сложной системой паролей, ограничения функций и доступа вашего терминала, успешно преодоленных насильником, предпринимателем и террористом, существует нечто непреодолимое, воплощенное в средствах идентификации пользователя и аутентификации. Это ваш образ, который характеризуется «почерком», психологическими особенностями, общей стратегией работы, т.е. индивидуальными чертами, протестированными системой защиты. Она хранит ваш образ и периодически сверяет с ним характер работы вашего терминала.

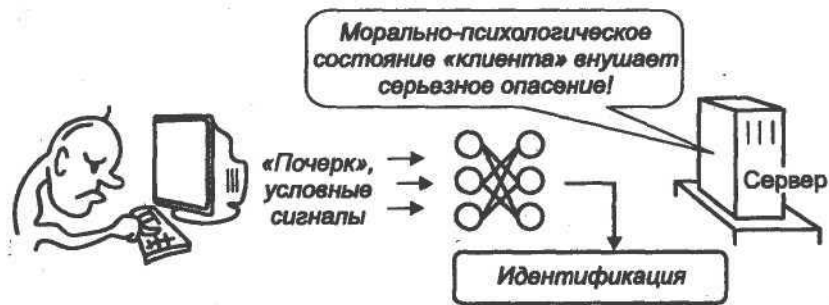


Рис. 9.13. Схема защиты информации

Сигнал тревоги может возникнуть не только при смене пользователя, но и в случае, когда его моральное состояние порождает серьезные отклонения от запечатленного образа. При этом могут использоваться и тайные специфические сигналы, но ряд основных характеристик обязателен. *Это* частота ввода символов с клавиатуры, продолжительность пауз между словами, частота использования клавиши "delete" и некоторых других и т.д. Возможно периодическое случайное анкетирование на основе единственного вопроса, требующее ответа «да — нет» и способствующее утверждению сложившегося образа.

Возникает вопрос: а причем здесь нейросеть? Ведь все эти проверки можно делать традиционными способами. Дело в том, что нейросеть — это средство *параллельного* выполнения сложных логических предикатов. Параллельного — значит, быстрого! А современный сервер — многопроцессорная вычислительная система, успешно выполняет роль нейрокомпьютера. Кроме того, при ассоциативном мышлении по принципу «на что это похоже» используются (конечно, в каких-то пределах) неполные и неточные данные, заставляющие по крайней мере встревожиться и послать администратора или даже охрану с проверкой.

## 9.8 Сивилла-прорицательница

В век всеобщей информатизации мы склонны ассоциативно прогнозировать: что будет с нами завтра, каковы общие тенденции и что предпринять?

Формализация обработки огромных потоков оперативной информации, в основном в средствах массовой информации, все более привлекает специалистов по прогнозу. В то же время здесь открываются огромные возможности для шарлатанов, и кажется удивительным, как остаются незамеченными несбывшиеся предсказания «авторитетных», не сходящих с телеэкрана, ясно-видящих.

Поставить прогноз на научную базу — вот задача современных честных ревнителей перспективы.

Основой прогнозирования являются 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 впредь. Комбинации и частота следования слов (рис. 9.14) образовывали критическую массу для неотвратимой цепной реакции социальных катаклизмов...

Итак, появление каждого «судьбоносного» слова добавляет определенное настроение нейросети. Результирующее настро-



Рис. 9.14. Прогнозирование



ние определяется распределением возбуждений нейронов выходного слоя — его максимумом, значением среднего, предпочтительным рядом. Частота их появления меняется во времени, обязательно предопределяя знаменательные события: развал Советского Союза, путч, вехи экономического краха, социальные и региональные беспорядки...

... 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.

В таком случае практический аппарат прогноза реализуется нейросетью, с помощью которой запоминаются отклонения вида "если —то", фиксирующие степень влияния часто употребляемых слов и выражений на прогноз социального, политического и экономического развития общества. Например, «безусловная победа, одной политической партии» — «увеличение веса тенденции тоталитаризма», «арест «олигарха» — «рост объема вывоза капитала», «эмиссия ценных бумаг компании» — «опасность банкротства». Указанные отношения не однозначны, так как эмиссия ценных бумаг может, наоборот, свидетельствовать об успешной деятельности предприятия и экономики в целом.

Очевидно, что указанная задача прогнозирования относится к классу трудно формализуемых. Это говорит об универсальности нейросетевых технологий, применимости средств искусственного интеллекта там, где неприменим никакой математический метод расчета.

## ЗАКЛЮЧЕНИЕ

Все чаще появляются научные работы, посвященных нейросетевым моделям, которые порождают все больше интересных, эффективных технологий для решения задач управления, планирования, распознавания образов и принятия решений. В этом разнообразии мы выделили лишь то направление, которое непосредственно связано с принципами человеческого интеллекта, с ассоциативным мышлением. Фактически это — построение таблиц, фиксирующих опыт и реализующих принцип «если — то», «посылка — следствие». Важным достоинством этих таблиц является то, что они позволяют наглядно воспроизводить функции многих переменных, не совместимых по типам и по физической природе. Эти функции не имеют формального математического описания и потому отображают трудно формализуемые задачи. Автоматическая интерполяция опыта является неотъемлемым свойством нейронной сети. Она позволяет делать вывод «на что больше всего *это* похоже и что следует предпринять», а также производить необходимое усреднение.

Важнейшее значение нейронные сети приобретают при построении систем управления и принятия решений. Технологии, основанные на применении «вычислительных» алгоритмов, приводят к тупиковой ситуации, когда традиционно развиваемая элементно-конструкторская база не обеспечивает требуемой производительности компьютеров. Нейросетевые технологии, использующие нетрудоемкий принцип ассоциативного мышления и высокий параллелизм, порождают прорыв в достижении необходимой реальной производительности вычислительных средств.

Конечно, статистические исследования накопленного опыта, моделирование и обработка нужны для обучения или построения обученной нейрости. Однако эти исследования проводятся вне рабочего режима, могут быть совместимы с ним, а также допускать корректирование, модификацию и развитие нейросети в процессе и эксплуатации.

Благодаря логическому методу построения математический аппарат нейронных сетей прост и доходчив, разработка нейросети не дорога, она не требует специальных знаний пользователя. В

то же время нейросеть легко допускает включение в рассмотрение новых факторов, изменение решений и т.д.

Воспроизведение принципов работы мозга позволяет создавать действительно универсальные, помехозащищенные, адаптивные, несложные и развиваемые информационные технологии в экономике, в бизнесе, в управлении сложными системами, в том числе и социальными, в искусстве, туризме и т.д.

Однако мы осознаем, что Природа действует в соответствии с законами своего развития. Для того чтобы реализовать примитивную логическую передаточную функцию, в нейроне происходит до 240 химических реакций. Создавая интеллект искусственный на принципиальном, логическом уровне, бессмысленно слепо и многотрудно воспроизводить существующие в природе методы, хотя такое моделирование может быть полезно для биологии и медицины. На пути логического создания разума мы не связаны с проблемами энергетики и надежности. Искусственная нейросеть не требует кровоснабжения и не подвергается огню на поражение. И потому мы смело идем по пути абстракции, сублимации и условной имитации, пользуясь богатыми возможностями современного компьютера и логического мышления.

Мы убеждены в необходимости создания студий *Computer Art Studio*, клубов по интересам и артелей, объединяющих все слои населения, всех возрастов и профессий, для реализации принципов искусственного интеллекта: это и искусство, и наука, и бизнес.

# ЛИТЕРАТУРА

## Использованная литература\*

1. *Галушкин А.И.* Сфера применения нейрокомпьютеров расширяется. Приложение к журналу «Информационные технологии». - 2001. - № 10.
2. *Галушкин А.И.* Теория нейронных сетей. Сер. «Нейрокомпьютеры и их применение». Кн. 1. - М.: ИПРЖР, 2000.
3. *Сигеру О., Марзуки Х., Рубин Ю.* Нейроуправление и его приложения / Под ред. А.И. Галушкина. Сер. «Нейрокомпьютеры и их применение». Кн. 2. - М.: ИПРЖР, 2000.
4. *Комашинский В.И., Смирнов ДА.* Введение в нейроинформационные технологии. — СПб.: Тема, 1999.
5. *Круглое В.В., Борисов В.В.* Искусственные нейронные сети. Теория и практика. — М.: Горячая линия — Телеком, 2001.
6. *Уоссерман Ф.* Нейрокомпьютерная техника. — М.: Мир, 1992.
7. ЭВМ пятого поколения. Концепции, проблемы, перспективы. — М.: Финансы и статистика, 1984.
8. *Игнатущенко В.В.* Организация структур управляющих многопроцессорных вычислительных систем. — М.: Энергоатомиздат, 1984.
9. *Барский А.Б.* Параллельные процессы в вычислительных системах: Планирование и организация. — М.: Радио и связь, 1990.
10. Нейрокомпьютеры и интеллектуальные роботы / Н.М. Амосов, Т.Н. Байдык, А.Д. Гольцов и др. — Киев: Наукова думка, 1991.
11. *Поспелов ДА.* Моделирование рассуждений: Опыт анализа мыслительных актов. - М.: Радио и связь, 1989.
12. *Hopfield J., Tank D.* Neurel computation of decision in optimization problems. Biol. Cybernet., 1985, vol. 52.
13. *Kohonen T.* Self-Organizing Maps. Springer-Verlag, 2<sup>nd</sup> ed., 1997.
14. *Шеннон Р.* Имитационное моделирование систем. Искусство и наука. — М.: Мир, 1978.

\* Источники расположены по мере упоминания их в тексте.

15. *Поспелов Д.А.* Большие системы. Ситуационное управление. - М.: Знание, 1975.
16. *Поспелов Д.А.* Ситуационное управление. Теория и практика. - М.: Наука, 1986.
17. *Головкин Б.А.* Параллельные вычислительные системы. — М.: Наука, 1980.
18. *Барский А.Б., Шилов В.В.* SPMD-архитектура и параллельная обработка структур данных. — Приложение к журналу «Информационные технологии». — 1999. — № 6.
19. *Барский А.Б., Шилов В.В.* SPMD-архитектура и параллельный логический вывод. — Приложение к журналу «Информационные технологии». - 1999. - № 12.
20. *Барский А. Б.* Параллельные технологии решения оптимизационных задач. — Приложение к журналу «Информационные технологии». - 2001. - № 2.
21. *Барский А.Б.* Обучение нейросети методом трассировки. Труды VIII Всеросс. конф. «Нейрокомпьютеры и их применение», 2002.
22. *Фролов Ю.В.* Интеллектуальные системы и управленческие решения. - М.: МГПУ, 2000.
23. *Фролов Ю.В., Пастухов Е.С.* Мониторинг изменений в банковском сообществе России с применением самоорганизующихся карт Кохонена // Банковские технологии. — 2000. — № 11.
24. *Deboeck G., Kohonen T.* Visual Exploration in Finance with Self-Organizing Maps. Springer-Verlag, 1998.

## **Дополнительная литература**

Нейронные сети: История развития теории / Под ред. А.И. Галушкина, Я.З. Цыпкина. Науч. сер. «Нейрокомпьютеры и их применение». Кн. 5. - М.: ИПРЖР, 2001.

*Головкин ВЛ.* Нейронные сети: обучение, организация и применение / Под ред. А.И. Галушкина. Науч. сер. «Нейрокомпьютеры и их применение». Кн. 4. - М.: ИПРЖР, 2001.

*Галушкин А.И.* Нейрокомпьютеры. Науч. сер. «Нейрокомпьютеры и их применение». Кн. 3. - М.: ИПРЖР, 2000.

*Горбань А.Н.* Обучение нейронных сетей. — М.: СП «ParaGraf», 1990.

*Терехов В.А., Ефимов Д.В., Тюкин И.Ю.* и др. Нейросетевые системы управления. - СПб.: СПбГУ, 1999.

Искусственный интеллект. Кн.1 / Под ред. Э.В. Попова; Кн. 2 / Под ред. Д.А. Поспелова. — М.: Радио и связь, 1990.

*Барский А.Б.* Нейронные сети и искусственный интеллект. — Приложение к журналу «Информационные технологии». — 2003. - № 1 .

Труды VIII Всеросс. конф. «Нейрокомпьютеры и их применение». Тезисы докладов. — М.: Век книги, 2002.

*Аляутдинов М.А., Балухто А.Н., Галушкин АИ.* и др. Разработка и исследование нейросетевых методов и алгоритмов обработки аэрокосмических изображений.

*Гаврилей Ю.К., Самарин АИ, Шевченко МА.* и др. Нейросетевые подходы к реализации алгоритмов активного зрения.

*Ланкин Ю.П., Кофман Г.Б.* Моделирование формообразования древесных стволов с помощью нейронных сетей.

*Томашевич Н.С.* Нейросетевые алгоритмы выделения признаков и распознавания объектов на 2D изображениях, инвариантные к аффинным преобразованиям.

*Тумоян Е.П.* Цифровая обработка изображений в задаче идентификации возбудителей туберкулеза.

*Елкин Д.В.* Применение нейронных сетей для решения задачи маршрутизации в вычислительных сетях с пакетной обработкой данных.

*Васильев В.И., Жернаков СВ.* Контроль и диагностика силовых установок летательных аппаратов на базе нейронных сетей.

*Друзяка И.С., Ковалевский С.В.* Моделирование функционально-двигательных характеристик спортсмена с применением нейросетевых технологий.

*Павловский В.Е., Серов А.Ю.* Нейросетевая система счисления пути и определения ориентации шагающего робота.

*Галуев ГА., Габриелян БА.* Нейросетевая система биометрической аутентификации личности для разграничения доступа к персональным компьютерам.

*Харламов А.А., Кушир ДА., Самаев Е.С.* Использование нейросетевых технологий в задаче распознавания речи.

*Кулик С.Д.* Программы для решения задач криминалистики.

*Орловский С.Н., Баскакова Т.Ф., Комиссаров СВ.* и др. Использование нейросетей для оптимальных технологий тушения лесных пожаров.

*Шевелев О.Г., Тюттерев В.В.* Нейронные сети в задаче определения авторства текста.

*Воловикова Е.М., Ковалевский СВ.* Персонал-технологии и нейросети.

*Зозуля Ю.И.* Интеллектуальные нейросистемы: модели и методы решения типовых задач автоматизации технологических процессов.

**Булдакова Т.Н., Кузнецов А.В. Нейросетевой** подход в задачах идентификации медицинских сигналов.

**Федяев О.И., Семисалов СМ., Гладунов С.А.** Нейросетевое прогнозирование динамики развития черепно-мозгового травматизма.

**Букаев Г.И., Романов А.Н., Бублик Н.Д. и др.** Опыт разработки новой технологии камеральных проверок, отбора налогоплательщиков для выездных проверок и оценки сумм ожидаемых доначислений на основе нейросетевых информационных моделей.

**Жучков АА.** Применение нейрорешений для снижения объемов финансирования и рисков инвестиций в модернизацию систем управления предприятиями.

**Ибраимова Т.Б.** Прогнозирование тенденций финансовых рынков с помощью нейронных сетей.

**Иванов М.Н.** Анализ роста курса акций с применением нейронных сетей.

**Захарченко В.М., Шука А.А.** Оптический нейрокомпьютер. Отечественный приоритет и перспективы развития.

**Литовченко Ц.Г., Митюков В.Г., Сафонов И.А.** Построение больших нейросетей на одном ПЛИС.

**Горлов В.Н.** К вопросу о механизме реализации ассоциативной памяти на принципах многослойных нейронных сетей.

**Лацев А.Я., Глушич Д.В.** Синтез алгоритмов обучения нейронных сетей.

**Стасевич В.П., Шумков ЕЛ.** Новый принцип построения самообучаемых нейросетей.

**Бутенко А.А., Галушкин А.И. Пантюхин Д.В.** Нейросетевые методы решения систем линейных алгебраических уравнений.

**Лоскутов А.И.** Синтез оптимальных программ испытания бортовой аппаратуры космических аппаратов с решением транспортной задачи на основе динамической нейронной сети Хопфилда.

**Владимирский Б.М.** Что вычисляет мозг? (**К**нейробиологическим основаниям нейрокомпьютинга.)

**Ланкин Ю.П.** Адаптивные модели: Стихионика, нейроинформатика, поисковая оптимизация.

**Савельев А.В.** Концептуальные системы нейробиологии и нейрокомпьютинга.

# СОДЕРЖАНИЕ

<b>К читателю</b> .....	<b>3</b>
<b>Предисловие</b> .....	<b>5</b>
<b>Введение</b> .....	<b>7</b>
<b>1. Основы нейросетевых технологий</b> .....	<b>13</b>
1.1. Модель мозга .....	13
1.2. Устойчивость и помехозащищенность .....	18
1.3. Ввод и «разглядывание» эталонов и образов .....	20
1.4. Пространство признаков .....	22
1.5. Кора .....	24
1.6. Локализация максимального возбуждения на выходном слое .....	24
<b>2. Построение современной нейросетевой технологии</b> .....	<b>30</b>
2.1. Построение примитивной нейросети .....	30
2.2. Возбуждение входного слоя .....	33
2.3. «Схемотехнический» подход к построению нейросети «под задачу» .....	35
2.4. Построение нейросети «под задачу» .....	42
2.5. Формализация нейросети .....	46
2.6. Модель механизма запоминания .....	50
2.7. Применение типовых нейросетей .....	52
2.8. Энергетика нейросети .....	58
2.9. Рекомендации .....	59
2.10. Нейросетевые технологии и нейрокомпьютеры .....	63
<b>3. Трассировка нейросети</b> .....	<b>67</b>
3.1. Подход: опыт — предпосылки обобщения .....	67
3.2. Алгоритм трассировки нейросети .....	86
3.3. Приведение нейросети после трассировки .....	88
3.4. Трассировка двухслойной нейросети .....	90



<b>4. Стратегии обучения и самообучения</b> . . . . .	<b>93</b>
4.1. Динамизм обучения. . . . .	93
4.2. Не задавайте глупых вопросов! . . . . .	99
4.3. Познание нового — основа самообучения. . . . .	100
<b>5. Нейронные сети с обратными связями</b> . . . . .	<b>103</b>
5.1. К вопросу о происхождении человека. . . . .	103
5.2. Как же вводить обратные связи? . . . . .	105
<b>6. Нейросетевые самообучающиеся системы управления</b> . . . . .	<b>109</b>
6.1. Самообучение на основе ситуационного управления . . . . .	109
6.2. Нейросетевое воплощение. . . . .	114
<b>7. Логическое программирование нейросети</b> . . . . .	<b>119</b>
7.1. ПРОЛОГ-программа_____	119
7.2. Нейросеть для задачи логического вывода . . . . .	123
<b>8. Нейросети и управление финансами</b> . . . . .	<b>128</b>
8.1. Табличный метод - основа искусственного интеллекта . . . . .	128
8.2. Мониторинг банковской системы_____	130
<b>9. Примеры применения нейросетевых технологий</b> ..	<b>141</b>
9.1. Проектирование игровой системы . . . . .	141
9.2. Служба безопасности . . . . .	152
9.3. Парк фантазмагорий . . . . .	153
9.4. Компьютерный человечек КОМПИ . . . . .	154
9.5. Графический диспетчер движения поездов_____	158
9.6. Печать рукописи . . . . .	162
9.7. Защита информации . . . . .	164
9.8. Сивилла-прорицательница . . . . .	165
<b>Заключение</b> . . . . .	<b>168</b>
<b>Литература</b> . . . . .	<b>170</b>

Производственное издание

**Барский Аркадий Бенционович**

**НЕЙРОННЫЕ СЕТИ: РАСПОЗНАВАНИЕ,  
УПРАВЛЕНИЕ, ПРИНЯТИЕ РЕШЕНИЙ**

Заведующая редакцией *Л.А. Табакова*  
Ведущий редактор *Н.А. Кузнецова*  
Редактор *Л.Г. Гагарина*  
Младший редактор *Н.А. Федорова*  
Художественный редактор *Ю.И. Артюхов*  
Технический редактор *И.В. Завгородняя*  
Корректоры *Т.М. Васильева, Г.В. Хлопцева*  
Компьютерная верстка *И.В. Зык*  
Оформление художника *Н.М. Биксентеева*

ИБ № 4654

Подписано в печать 17.05.2004. Формат 60x88/16  
Гарнитура «Таймс». Печать офсетная  
Усл.пл. 10,78. Уч.-изд. л. 10,44  
Тираж 3000 экз. Заказ 489. «С» 120

Издательство «Финансы и статистика»  
101000, Москва, ул. Покровка, 7  
Телефон (095) 925-35-02, факс (095) 925-09-57  
E-mail: mail@finstat.ru <http://www.finstat.ru>

ГП Псковской области «Великолукская городская типография»  
Комитета по средствам массовой информации  
182100, Великие Луки, ул. Полиграфистов, 78/12  
Тел./факс: (811-53) 3-62-95  
E-mail: VTL@MART.RU

В серии "Прикладные информационные технологии"  
вышла книга

**А.Н.Калашян, Г.Н.Калянов**  
**Структурные модели бизнеса:**  
**DFD-технологии, 256 с.**

Изложена технология современного структурного анализа и проектирования бизнес-процессов, основанная на потоковых диаграммах. Приведены примеры моделей основных и вспомогательных бизнес-процессов из различных отраслей и направлений производственно-хозяйственной деятельности современных предприятий и учреждений. Представленная технология и модели могут быть рассмотрены в качестве рабочего инструмента и соответственно рабочих прототипов для практической деятельности специалистов.

Для студентов, аспирантов, преподавателей экономических вузов, специалистов-менеджеров всех уровней, а также для получающих второе высшее образование в области менеджмента.

Книгу можно приобрести в киоске издательства по адресу:  
101000, Москва, ул.Покровка, 7  
(метро "Китай-город", выход на ул.Маросейка)  
Тел.: (095)925-35-02, 923-18-68, 923-80-42. Факс (095)925-09-57  
E-mail: mail@finstat.ru <http://www.finstat.ru>

ISBN 5-279-02757-X



9 785279 027576