

Ф. Уоссермен

## **Нейрокомпьютерная техника: Теория и практика**

В книге американского автора в общедоступной форме излагаются основы построения нейрокомпьютеров. Описаны структура нейронных сетей и различные алгоритмы их настройки. Отдельные главы посвящены вопросам реализации нейронных сетей.

Для специалистов в области вычислительной техники, а также студентов соответствующих специальностей вузов.

Перевод на русский язык, Ю. А. Зуев, В. А. Точенов, 1992.

## ОГЛАВЛЕНИЕ

<b>ПРЕДИСЛОВИЕ .....</b>	<b>4</b>
БЛАГОДАРНОСТИ.....	4
<b>ВВЕДЕНИЕ .....</b>	<b>4</b>
ПОЧЕМУ ИМЕННО ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ? .....	4
СВОЙСТВА ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ .....	4
ИСТОРИЧЕСКИЙ АСПЕКТ .....	5
ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ СЕГОДНЯ .....	7
ПЕРСПЕКТИВЫ НА БУДУЩЕЕ.....	7
ВЫВОДЫ .....	8
<b>ГЛАВА 1. ОСНОВЫ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ.....</b>	<b>9</b>
БИОЛОГИЧЕСКИЙ ПРОТОТИП .....	9
ИСКУССТВЕННЫЙ НЕЙРОН .....	10
ОДНОСЛОЙНЫЕ ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ .....	13
МНОГОСЛОЙНЫЕ ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ.....	13
ТЕРМИНОЛОГИЯ, ОБОЗНАЧЕНИЯ И СХЕМАТИЧЕСКОЕ ИЗОБРАЖЕНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ .....	15
ПРОЛОГ .....	16
<b>ГЛАВА 2. ПЕРСЕПТРОНЫ.....</b>	<b>17</b>
ПЕРСЕПТРОНЫ И ЗАРОЖДЕНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ.....	17
ПЕРСЕПТРОННАЯ ПРЕДСТАВЛЯЕМОСТЬ.....	18
ОБУЧЕНИЕ ПЕРСЕПТРОНА .....	24
АЛГОРИТМ ОБУЧЕНИЯ ПЕРСЕПТРОНА .....	24
<b>ГЛАВА 3. ПРОЦЕДУРА ОБРАТНОГО РАСПРОСТРАНЕНИЯ.....</b>	<b>27</b>
ВВЕДЕНИЕ В ПРОЦЕДУРУ ОБРАТНОГО РАСПРОСТРАНЕНИЯ .....	27
ОБУЧАЮЩИЙ АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ .....	27
ОБЗОР ОБУЧЕНИЯ.....	29
ДАЛЬНЕЙШИЕ АЛГОРИТМИЧЕСКИЕ РАЗРАБОТКИ.....	34
ПРИМЕНЕНИЯ .....	34
ПРЕДОСТЕРЕЖЕНИЕ .....	35
<b>ГЛАВА 4. СЕТИ ВСТРЕЧНОГО РАСПРОСТРАНЕНИЯ.....</b>	<b>36</b>
ВВЕДЕНИЕ В СЕТИ ВСТРЕЧНОГО РАСПРОСТРАНЕНИЯ.....	36
СТРУКТУРА СЕТИ.....	36
НОРМАЛЬНОЕ ФУНКЦИОНИРОВАНИЕ .....	37
ОБУЧЕНИЕ СЛОЯ КОХОНЕНА .....	38
ОБУЧЕНИЕ СЛОЯ ГРОССБЕРГА.....	42
СЕТЬ ВСТРЕЧНОГО РАСПРОСТРАНЕНИЯ ПОЛНОСТЬЮ .....	42
ПРИЛОЖЕНИЕ: СЖАТИЕ ДАННЫХ.....	43
ОБСУЖДЕНИЕ .....	44
<b>ГЛАВА 5. СТОХАСТИЧЕСКИЕ МЕТОДЫ .....</b>	<b>45</b>
ИСПОЛЬЗОВАНИЕ ОБУЧЕНИЯ .....	45
ПРИЛОЖЕНИЯ К ОБЩИМ НЕЛИНЕЙНЫМ ЗАДАЧАМ ОПТИМИЗАЦИИ .....	49
ОБРАТНОЕ РАСПРОСТРАНЕНИЕ И ОБУЧЕНИЕ КОШИ .....	50

<b>ГЛАВА 6. СЕТИ ХОПФИЛДА.....</b>	<b>52</b>
КОНФИГУРАЦИИ СЕТЕЙ С ОБРАТНЫМИ СВЯЗЯМИ.....	53
ПРИЛОЖЕНИЯ.....	59
ОБСУЖДЕНИЕ.....	62
ВЫВОДЫ.....	63
<b>ГЛАВА 7. ДВУНАПРАВЛЕННАЯ АССОЦИАТИВНАЯ ПАМЯТЬ .....</b>	<b>64</b>
СТРУКТУРА ДАП.....	64
ВОССТАНОВЛЕНИЕ ЗАПОМНЕННЫХ АССОЦИАЦИЙ.....	65
КОДИРОВАНИЕ АССОЦИАЦИЙ.....	66
ЕМКОСТЬ ПАМЯТИ.....	68
НЕПРЕРЫВНАЯ ДАП.....	68
АДАПТИВНАЯ ДАП.....	69
КОНКУРИРУЮЩАЯ ДАП.....	69
ЗАКЛЮЧЕНИЕ.....	69
<b>ГЛАВА 8. АДАПТИВНАЯ РЕЗОНАНСНАЯ ТЕОРИЯ .....</b>	<b>70</b>
АРХИТЕКТУРА АРТ.....	70
РЕАЛИЗАЦИЯ АРТ.....	76
ПРИМЕР ОБУЧЕНИЯ СЕТИ АРТ.....	78
ХАРАКТЕРИСТИКИ АРТ.....	80
ЗАКЛЮЧЕНИЕ.....	82
<b>ГЛАВА 9. ОПТИЧЕСКИЕ НЕЙРОННЫЕ СЕТИ .....</b>	<b>83</b>
ВЕКТОРНО-МАТРИЧНЫЕ УМНОЖИТЕЛИ.....	83
ГОЛОГРАФИЧЕСКИЕ КОРРЕЛЯТОРЫ.....	88
ЗАКЛЮЧЕНИЕ.....	91
<b>ГЛАВА 10. КОГНИТРОН И НЕОКОГНИТРОН.....</b>	<b>92</b>
КОГНИТРОН.....	93
НЕОКОГНИТРОН.....	100
ЗАКЛЮЧЕНИЕ.....	104
<b>ПРИЛОЖЕНИЕ А. БИОЛОГИЧЕСКИЕ НЕЙРОННЫЕ СЕТИ.....</b>	<b>106</b>
ЧЕЛОВЕЧЕСКИЙ МОЗГ: БИОЛОГИЧЕСКАЯ МОДЕЛЬ ДЛЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ.....	106
ОРГАНИЗАЦИЯ ЧЕЛОВЕЧЕСКОГО МОЗГА.....	106
КОМПЬЮТЕРЫ И ЧЕЛОВЕЧЕСКИЙ МОЗГ.....	111
<b>ПРИЛОЖЕНИЕ Б. АЛГОРИТМЫ ОБУЧЕНИЯ.....</b>	<b>113</b>
ОБУЧЕНИЕ С УЧИТЕЛЕМ И БЕЗ УЧИТЕЛЯ.....	113
МЕТОД ОБУЧЕНИЯ ХЭББА.....	113
ВХОДНЫЕ И ВЫХОДНЫЕ ЗВЕЗДЫ.....	114
ОБУЧЕНИЕ ПЕРСЕПТРОНА.....	116
МЕТОД ОБУЧЕНИЯ УИДРОУ-ХОФФА.....	117
МЕТОДЫ СТАТИСТИЧЕСКОГО ОБУЧЕНИЯ.....	117
САМООРГАНИЗАЦИЯ.....	117

## Предисловие

Что такое искусственные нейронные сети? Что они могут делать? Как они работают? Как их можно использовать? Эти и множество подобных вопросов задают специалисты из разных областей. Найти вразумительный ответ нелегко. Университетских курсов мало, семинары слишком дороги, а соответствующая литература слишком обширна и специализированна. Готовящиеся к печати превосходные книги могут обескуражить начинающих. Часто написанные на техническом жаргоне, многие из них предполагают свободное владение разделами высшей математики, редко используемыми в других областях.

Эта книга является систематизированным вводным курсом для профессионалов, не специализирующихся в математике. Все важные понятия формулируются сначала обычным языком. Математические выкладки используются, если они делают изложение более ясным. В конце глав помещены сложные выводы и доказательства, а также приводятся ссылки на другие работы. Эти ссылки составляют обширную библиографию важнейших работ в областях, связанных с искусственными нейронными сетями. Такой многоуровневый подход не только предоставляет читателю обзор по искусственным нейронным сетям, но также позволяет заинтересованным лицам серьезнее и глубже изучить предмет.

Значительные усилия были приложены, чтобы сделать книгу понятной и без чрезмерного упрощения материала. Читателям, пожелавшим продолжить более углубленное теоретическое изучение, не придется переучиваться. При упрощенном изложении даются ссылки на более подробные работы.

Книгу не обязательно читать от начала до конца. Каждая глава предполагается замкнутой, поэтому для понимания достаточно лишь знакомства с содержанием гл. 1 и 2. Хотя некоторое повторение материала неизбежно, большинству читателей это не будет обременительно.

Книга имеет практическую направленность. Если главы внимательно изучены, то большую часть сетей оказывается возможным реализовать на обычном компьютере общего назначения. Читателю настоятельно рекомендуется так и поступать. Никакой другой метод не позволит добиться столь же глубокого понимания.

## БЛАГОДАРНОСТИ

Прежде всего самую глубокую признательность я хотел бы выразить своей жене Саре за то, что она воодушевляла меня, а также за ее терпение в течение тех месяцев, которые я провел за пишущей машинкой.

Я хотел бы поблагодарить моих друзей и коллег, которые так великодушно дарили мне свое время и знания, исправляли мои ошибки и создавали атмосферу, способствующую развитию идей. Я хотел бы выразить мою особую признательность Dr. Surapol Dasananda, Santa Clara University; Dr. Elizabeth Center, College of Notre Dame; Dr. Peter Rowe, College of Notre Dame; Caries Rockwell, Microlog Corp.; Tom Schwartz, The Schwartz Associates; Dennis Reinhardt, Dair Corp.; Coe Miles-Schlichting; and Douglas Marquardt. Благодарю также Kyla Carlson и Nang Cao за помощь в подготовке иллюстраций.

На мне лежит, разумеется, ответственность за все оставшиеся неисправленными ошибки, так как мои друзья и коллеги не могли опекать меня ежеминутно.

## Введение

### ПОЧЕМУ ИМЕННО ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ?

После двух десятилетий почти полного забвения интерес к искусственным нейронным сетям быстро вырос за последние несколько лет. Специалисты из таких далеких областей, как техническое конструирование, философия, физиология и психология, заинтригованы возможностями, предоставляемыми этой технологией, и ищут приложения им внутри своих дисциплин.

Это возрождение интереса было вызвано как теоретическими, так и прикладными достижениями. Неожиданно открылись возможности использования вычислений в сферах, до этого относящихся лишь к области человеческого интеллекта, возможности создания машин, способность которых учиться и запоминать удивительным образом напоминает мыслительные процессы человека, и наполнения новым значительным содержанием критикованного термина «искусственный интеллект».

### СВОЙСТВА ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Искусственные нейронные сети индуцированы биологией, так как они состоят из элементов, функциональные возможности которых аналогичны большинству элементарных функций биологического нейрона. Эти элементы затем организуются по способу, который может соответствовать (или не соответствовать) анатомии мозга. Несмотря на такое поверхностное сходство, искусственные нейронные сети демонстрируют удивительное число свойств присущих мозгу. Например,

они обучаются на основе опыта, обобщают предыдущие прецеденты на новые случаи и извлекают существенные свойства из поступающей информации, содержащей излишние данные.

Несмотря на такое функциональное сходство, даже самый оптимистичный их защитник не предположит, что в скором будущем искусственные нейронные сети будут дублировать функции человеческого мозга. Реальный «интеллект», демонстрируемый самыми сложными нейронными сетями, находится ниже уровня дождевого червя, и энтузиазм должен быть умерен в соответствии с современными реалиями. Однако равным образом было бы неверным игнорировать удивительное сходство в функционировании некоторых нейронных сетей с человеческим мозгом. Эти возможности, как бы они ни были ограничены сегодня, наводят на мысль, что глубокое проникновение в человеческий интеллект, а также множество революционных приложений, могут быть не за горами.

### **Обучение**

Искусственные нейронные сети могут менять свое поведение в зависимости от внешней среды. Этот фактор в большей степени, чем любой другой, ответствен за тот интерес, который они вызывают. После предъявления входных сигналов (возможно, вместе с требуемыми выходами) они самонастраиваются, чтобы обеспечивать требуемую реакцию. Было разработано множество обучающих алгоритмов, каждый со своими сильными и слабыми сторонами. Как будет указано в этой книге позднее, все еще существуют проблемы относительно того, чему сеть может обучиться и как обучение должно проводиться.

### **Обобщение**

Отклик сети после обучения может быть до некоторой степени нечувствителен к небольшим изменениям входных сигналов. Эта внутренне присущая способность видеть образ сквозь шум и искажения жизненно важна для распознавания образов в реальном мире. Она позволяет преодолеть требование строгой точности, предъявляемое обычным компьютером, и открывает путь к системе, которая может иметь дело с тем несовершенным миром, в котором мы живем. Важно отметить, что искусственная нейронная сеть делает обобщения автоматически благодаря своей структуре, а не с помощью использования «человеческого интеллекта» в форме специально написанных компьютерных программ.

### **Абстрагирование**

Некоторые из искусственных нейронных сетей обладают способностью извлекать сущность из входных сигналов. Например, сеть может быть обучена на последовательность искаженных версий буквы «А». После соответствующего обучения предъявление такого искаженного примера приведет к тому, что сеть породит букву совершенной формы. В некотором смысле она научится порождать то, что никогда не видела.

Эта способность извлекать идеальное из несовершенных входов ставит интересные философские вопросы. Она напоминает концепцию идеалов, выдвинутую Платоном в его «Республике». Во всяком случае способность извлекать идеальные прототипы является у людей весьма ценным качеством.

### **Применимость**

Искусственные нейронные сети не являются панацеей. Они, очевидно, не годятся для выполнения таких задач, как начисление заработной платы. Похоже, однако, что им будет отдаваться предпочтение в большом классе задач распознавания образов, с которыми плохо или вообще не справляются обычные компьютеры.

## **ИСТОРИЧЕСКИЙ АСПЕКТ**

Людей всегда интересовало их собственное мышление. Это самовопрошение, думание мозга о себе самом является, возможно, отличительной чертой человека. Имеется множество размышлений о природе мышления, простирающихся от духовных до анатомических. Обсуждение этого вопроса, протекавшее в горячих спорах философов и теологов с физиологами и анатомами, принесло мало пользы, так как сам предмет весьма труден для изучения. Те, кто опирался на самоанализ и размышление, пришли к выводам, не отвечающим уровню строгости физических наук. Экспериментаторы же нашли, что мозг труден для наблюдения и ставит в тупик своей организацией. Короче говоря, мощные методы научного исследования, изменившие наш взгляд на физическую реальность, оказались бессильными в понимании самого человека.

Нейробиологи и нейроанатомы достигли значительного прогресса. Усердно изучая структуру и функции нервной системы человека, они многое поняли в «электропроводке» мозга, но мало узнали о его функционировании. В процессе накопления ими знаний выяснилось, что мозг имеет ошеломляющую сложность. Сотни миллиардов нейронов, каждый из которых соединен с сотнями или тысячами других, образуют систему, далеко превосходящую наши самые смелые мечты о

суперкомпьютерах. Тем не менее мозг постепенно выдает свои секреты в процессе одного из самых напряженных и честолюбивых исследований в истории человечества.

Лучшее понимание функционирования нейрона и картины его связей позволило исследователям создать математические модели для проверки своих теорий. Эксперименты теперь могут проводиться на цифровых компьютерах без привлечения человека или животных, что решает многие практические и морально-этические проблемы. В первых же работах выяснилось, что эти модели не только повторяют функции мозга, но и способны выполнять функции, имеющие свою собственную ценность. Поэтому возникли и остаются в настоящее время две взаимно обогащающие друг-друга цели нейронного моделирования: первая – понять функционирование нервной системы человека на уровне физиологии и психологии и вторая – создать вычислительные системы (искусственные нейронные сети), выполняющие функции, сходные с функциями мозга. Именно эта последняя цель и находится в центре внимания этой книги.

Параллельно с прогрессом в нейроанатомии и нейрофизиологии психологами были созданы модели человеческого обучения. Одной из таких моделей, оказавшейся наиболее плодотворной, была модель Д. Хэбба, который в 1949г. предложил закон обучения, явившийся стартовой точкой для алгоритмов обучения искусственных нейронных сетей. Дополненный сегодня множеством других методов он продемонстрировал ученым того времени, как сеть нейронов может обучаться.

В пятидесятые и шестидесятые годы группа исследователей, объединив эти биологические и физиологические подходы, создала первые искусственные нейронные сети. Выполненные первоначально как электронные сети, они были позднее перенесены в более гибкую среду компьютерного моделирования, сохранившуюся и в настоящее время. Первые успехи вызвали взрыв активности и оптимизма. Минский, Розенблатт, Уидроу и другие разработали сети, состоящие из одного слоя искусственных нейронов. Часто называемые перцептронами, они были использованы для такого широкого класса задач, как предсказание погоды, анализ электрокардиограмм и искусственное зрение. В течение некоторого времени казалось, что ключ к интеллекту найден и воспроизведение человеческого мозга является лишь вопросом конструирования достаточно большой сети.

Но эта иллюзия скоро рассеялась. Сети не могли решать задачи, внешне весьма сходные с теми, которые они успешно решали. С этих необъяснимых неудач начался период интенсивного анализа. Минский, используя точные математические методы, строго доказал ряд теорем, относящихся к функционированию сетей.

Его исследования привели к написанию книги [4], в которой он вместе с Пайпертом доказал, что используемые в то время однослойные сети теоретически неспособны решить многие простые задачи, в том числе реализовать функцию «Исключающее ИЛИ». Минский также не был оптимистичен относительно потенциально возможного здесь прогресса:

Перцептрон показал себя заслуживающим изучения, несмотря на жесткие ограничения (и даже благодаря им). У него много привлекательных свойств: линейность, занимательная теорема об обучении, простота модели параллельных вычислений. Нет оснований полагать, что эти достоинства сохраняться при переходе к многослойным системам. Тем не менее мы считаем важной задачей для исследования подкрепление (или опровержение) нашего интуитивного убеждения, что такой переход бесплоден.

Возможно, будет открыта какая-то мощная теорема о сходимости или найдена глубокая причина неудач дать интересную «теорему обучения» для многослойных машин ([4], с.231-232).

Блеск и строгость аргументации Минского, а также его престиж породили огромное доверие к книге – ее выводы были неувязимы. Разочарованные исследователи оставили поле исследований ради более обещающих областей, а правительства перераспределили свои субсидии, и искусственные нейронные сети были забыты почти на два десятилетия.

Тем не менее несколько наиболее настойчивых ученых, таких как Кохонен, Гроссберг, Андерсон продолжили исследования. Наряду с плохим финансированием и недостаточной оценкой ряд исследователей испытывал затруднения с публикациями. Поэтому исследования, опубликованные в семидесятые и начале восьмидесятых годов, разбросаны в массу различных журналов, некоторые из которых малоизвестны. Постепенно появился теоретический фундамент, на основе которого сегодня конструируются наиболее мощные многослойные сети. Оценка Минского оказалась излишне пессимистичной, многие из поставленных в его книге задач решаются сейчас сетями с помощью стандартных процедур.

За последние несколько лет теория стала применяться в прикладных областях и появились новые корпорации, занимающиеся коммерческим использованием этой технологии. Нарастание научной активности носило взрывной характер. В 1987 г. было проведено четыре крупных совещания по искусственным нейронным сетям и опубликовано свыше 500 научных сообщений – феноменальная скорость роста.

Урок, который можно извлечь из этой истории, выражается законом Кларка, выдвинутым писателем и ученым Артуром Кларком. В нем утверждается, что, если крупный уважаемый ученый говорит, что нечто может быть выполнено, то он (или она) почти всегда прав. Если же ученый говорит, что это не может быть выполнено, то он (или она) почти всегда не прав. История науки является летописью ошибок и частичных истин. То, что сегодня не подвергается сомнениям, завтра отвергается. Некритическое восприятие «фактов» независимо от их источника может парализовать научный поиск. С одной стороны, блестящая научная работа Минского задержала развитие искусственных нейронных сетей. Нет сомнений, однако, в том, что область

пострадала вследствие необоснованного оптимизма и отсутствия достаточной теоретической базы. И возможно, что шок, вызванный книгой «Перцептроны», обеспечил необходимый для созревания этой научной области период.

## **ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ СЕГОДНЯ**

Имеется много впечатляющих демонстраций возможностей искусственных нейронных сетей: сеть научили превращать текст в фонетическое представление, которое затем с помощью уже иных методов превращалось в речь [7]; другая сеть может распознавать рукописные буквы [1]; сконструирована система сжатия изображений, основанная на нейронной сети [2]. Все они используют сеть обратного распространения – наиболее успешный, по-видимому, из современных алгоритмов. Обратное распространение, независимо предложенное в трех различных работах [8, 5, 6,], является систематическим методом для обучения многослойных сетей, и тем самым преодолевает ограничения, указанные Минским.

Как подчеркивается в следующих главах, обратное распространение не свободно от проблем. Прежде всего нет гарантии, что сеть может быть обучена за конечное время. Много усилий, израсходованных на обучение, пропадает напрасно после затрат большого количества машинного времени. Когда это происходит, попытка обучения повторяется – без всякой уверенности, что результат окажется лучше. Нет также уверенности, что сеть обучится наилучшим возможным образом. Алгоритм обучения может попасть в «ловушку» так называемого локального минимума и будет получено худшее решение.

Разработано много других сетевых алгоритмов обучения, имеющих свои специфические преимущества. Некоторые из них обсуждаются в последующих главах. Следует подчеркнуть, что никакая из сегодняшних сетей не является панацеей, все они страдают от ограничений в своих возможностях обучаться и вспоминать.

Мы имеем дело с областью, продемонстрировавшей свою работоспособность, имеющей уникальные потенциальные возможности, много ограничений и множество открытых вопросов. Такая ситуация настраивает на умеренный оптимизм. Авторы склонны публиковать свои успехи, но не неудачи, создавая тем самым впечатление, которое может оказаться нереалистичным. Те, кто ищет капитал, чтобы рискнуть и основать новые фирмы, должны представить убедительный проект последующего осуществления и прибыли. Существует, следовательно, опасность, что искусственные нейронные сети начнут продавать раньше, чем придет их время, обещая функциональные возможности, которых пока невозможно достигнуть. Если это произойдет, то область в целом может пострадать от потери кредита доверия и вернуться к застою периода семидесятых годов. Для улучшения существующих сетей требуется много основательной работы. Должны быть развиты новые технологии, улучшены существующие методы и расширены теоретические основы, прежде чем данная область сможет полностью реализовать свои потенциальные возможности.

## **ПЕРСПЕКТИВЫ НА БУДУЩЕЕ**

Искусственные нейронные сети предложены для задач, простирающихся от управления боем до присмотра за ребенком. Потенциальными приложениями являются те, где человеческий интеллект малоэффективен, а обычные вычисления трудоемки или неадекватны. Этот класс приложений во всяком случае не меньше класса, обслуживаемого обычными вычислениями, и можно предполагать, что искусственные нейронные сети займут свое место наряду с обычными вычислениями в качестве дополнения такого же объема и важности.

### **Искусственные нейронные сети и экспертные системы**

В последние годы над искусственными нейронными сетями доминировали логические и символично-операционные дисциплины. Например, широко пропагандировались экспертные системы, у которых имеется много заметных успехов, так же, как и неудач. Кое-кто говорит, что искусственные нейронные сети заменят собой современный искусственный интеллект, но многое свидетельствует о том, что они будут существовать, объединяясь в системах, где каждый подход используется для решения тех задач, с которыми он лучше справляется.

Эта точка зрения подкрепляется тем, как люди функционируют в нашем мире. Распознавание образов отвечает за активность, требующую быстрой реакции. Так как действия совершаются быстро и бессознательно, то этот способ функционирования важен для выживания во враждебном окружении. Вообразите только, что было бы, если бы наши предки вынуждены были обдумывать свою реакцию на прыгнувшего хищника?

Когда наша система распознавания образов не в состоянии дать адекватную интерпретацию, вопрос передается в высшие отделы мозга. Они могут запросить добавочную информацию и займут больше времени, но качество полученных в результате решений может быть выше.

Можно представить себе искусственную систему, подражающую такому разделению труда. Искусственная нейронная сеть реагировала бы в большинстве случаев подходящим образом на внешнюю среду. Так как такие сети способны указывать доверительный уровень каждого решения, то сеть «знает, что она не знает» и передает данный случай для разрешения экспертной

системе. Решения, принимаемые на этом более высоком уровне, были бы конкретными и логичными, но они могут нуждаться в сборе дополнительных фактов для получения окончательного заключения. Комбинация двух систем была бы более мощной, чем каждая из систем в отдельности, следуя при этом высокоэффективной модели, даваемой биологической эволюцией.

### Соображения надежности

Прежде чем искусственные нейронные сети можно будет использовать там, где поставлены на карту человеческая жизнь или ценное имущество, должны быть решены вопросы, относящиеся к их надежности.

Подобно людям, структуру мозга которых они копируют, искусственные нейронные сети сохраняют в определенной мере непредсказуемость. Единственный способ точно знать выход состоит в испытании всех возможных входных сигналов. В большой сети такая полная проверка практически неосуществима и должны использоваться статистические методы для оценки функционирования. В некоторых случаях это недопустимо. Например, что является допустимым уровнем ошибок для сети, управляющей системой космической обороны? Большинство людей скажет, любая ошибка недопустима, так как ведет к огромному числу жертв и разрушений. Это отношение не меняется от того обстоятельства, что человек в подобной ситуации также может допускать ошибки.

Проблема возникает из-за допущения полной безошибочности компьютеров. Так как искусственные нейронные сети иногда будут совершать ошибки даже при правильном функционировании, то, как ощущается многими, это ведет к ненадежности – качеству, которое мы считаем недопустимым для наших машин.

Сходная трудность заключается в неспособности традиционных искусственных нейронных сетей "объяснить", как они решают задачу. Внутреннее представление, получающееся в результате обучения, часто настолько сложно, что его невозможно проанализировать, за исключением самых простых случаев. Это напоминает нашу неспособность объяснить, как мы узнаем человека, несмотря на различие в расстоянии, угле, освещении и на прошедшие годы. Экспертная система может проследить процесс своих рассуждений в обратном порядке, так что человек может проверить ее на разумность. Сообщалось о встраивании этой способности в искусственные нейронные сети [3], что может существенно повлиять на приемлемость этих систем.

### ВЫВОДЫ

Искусственные нейронные сети являются важным расширением понятия вычисления. Они обещают создание автоматов, выполняющих функции, бывшие ранее исключительной прерогативой человека. Машины могут выполнять скучные, монотонные и опасные задания, и с развитием технологии возникнут совершенно новые приложения.

Теория искусственных нейронных сетей развивается стремительно, но в настоящее время она недостаточна, чтобы быть опорой для наиболее оптимистических проектов. В ретроспективе видно, что теория развивалась быстрее, чем предсказывали пессимисты, но медленнее, чем надеялись оптимисты, – типичная ситуация. Сегодняшний взрыв интереса привлек к нейронным сетям тысячи исследователей. Резонно ожидать быстрого роста нашего понимания искусственных нейронных сетей, ведущего к более совершенным сетевым парадигмам и множеству прикладных возможностей.

### Литература

1. Burr, D. J. 1987. Experiments with a connectionist text reader. In Proceedings of the First International on Neural Networks, eds. M. Caudill and C. Butler, vol. 4, pp. 717–24. San Diego, CA: SOS Printing.
2. Cottrell, G. W., Munro P., and Zipser D., 1987. Image compressions by backpropagation: An example of extensional programming. Advances in cognitive science (vol.3). Norwood, NJ: Ablex.
3. Gallant S. I., 1988. Connectionist expert system. Communications of the ACM 31:152–69.
4. Minsky M., and Papert S., 1969. Perceptrons. Cambridge, MA: MIT Press. (Русский перевод: Минский М. Л., Пейперт С. Перцептроны. –М. Мир. – 1971.
5. Parker, D. B. 1982. Learning-logic. Invention Report, s. 81–64, File 1. Office of Technology Licensing, Stanford University.
6. Rumelhart D. E., Hinton G. E., and Williams R. J. 1986. Learning internal representations by error propagation. In Parallel distributed processing, vol. 1, pp. 318–62. Cambridge, MA: MIT Press.
7. Sejnowski T. J., and Rosenberg C. R. 1987. Parallel Networks that learn to pronounce English text. Complex Systems 3:145–68.
8. Werbos P. J. 1974. Beyond regression: New tools for prediction and analysis in the behavioral sciences. Masters thesis. Harvard University.



## Глава 1.

### Основы искусственных нейронных сетей

Искусственные нейронные сети чрезвычайно разнообразны по своим конфигурациям. Несмотря на такое разнообразие, сетевые парадигмы имеют много общего. В этой главе подобные вопросы затрагиваются для того, чтобы читатель был знаком с ними к тому моменту, когда позднее они снова встретятся в книге.

Используемые здесь обозначения и графические представления были выбраны как наиболее широко используемые в настоящее время (опубликованных стандартов не имеется), они сохраняются на протяжении всей книги.

#### БИОЛОГИЧЕСКИЙ ПРОТОТИП

Развитие искусственных нейронных сетей вдохновляется биологией. То есть рассматривая сетевые конфигурации и алгоритмы, исследователи мыслят их в терминах организации мозговой деятельности. Но на этом аналогия может и закончиться. Наши знания о работе мозга столь ограничены, что мало бы нашлось руководящих ориентиров для тех, кто стал бы ему подражать. Поэтому разработчикам сетей приходится выходить за пределы современных биологических знаний в поисках структур, способных выполнять полезные функции. Во многих случаях это приводит к необходимости отказа от биологического правдоподобия, мозг становится просто метафорой, и создаются сети, невозможные в живой материи или требующие неправдоподобно больших допущений об анатомии и функционировании мозга.

Несмотря на то что связь с биологией слаба и зачастую несущественна, искусственные нейронные сети продолжают сравниваться с мозгом. Их функционирование часто напоминает человеческое познание, поэтому трудно избежать этой аналогии. К сожалению, такие сравнения неплодотворны и создают неоправданные ожидания, неизбежно ведущие к разочарованию. Исследовательский энтузиазм, основанный на ложных надеждах, может испариться, столкнувшись с суровой действительностью, как это уже однажды было в шестидесятые годы, и многообещающая область снова придет в упадок, если не будет соблюдаться необходимая сдержанность.

Несмотря на сделанные предупреждения, полезно все же знать кое-что о нервной системе млекопитающих, так как она успешно решает задачи, к выполнению которых лишь стремятся искусственные системы. Последующее обсуждение весьма кратко. Приложение А содержит более обширное (но ни в коем случае не полное) рассмотрение нервной системы млекопитающих для тех, кто хочет узнать больше об этом восхитительном предмете.

Нервная система человека, построенная из элементов, называемых нейронами, имеет ошеломляющую сложность. Около  $10^{11}$  нейронов участвуют в примерно  $10^{15}$  передающих связях, имеющих длину метр и более. Каждый нейрон обладает многими качествами, общими с другими элементами тела, но его уникальной способностью является прием, обработка и передача электрохимических сигналов по нервным путям, которые образуют коммуникационную систему мозга.

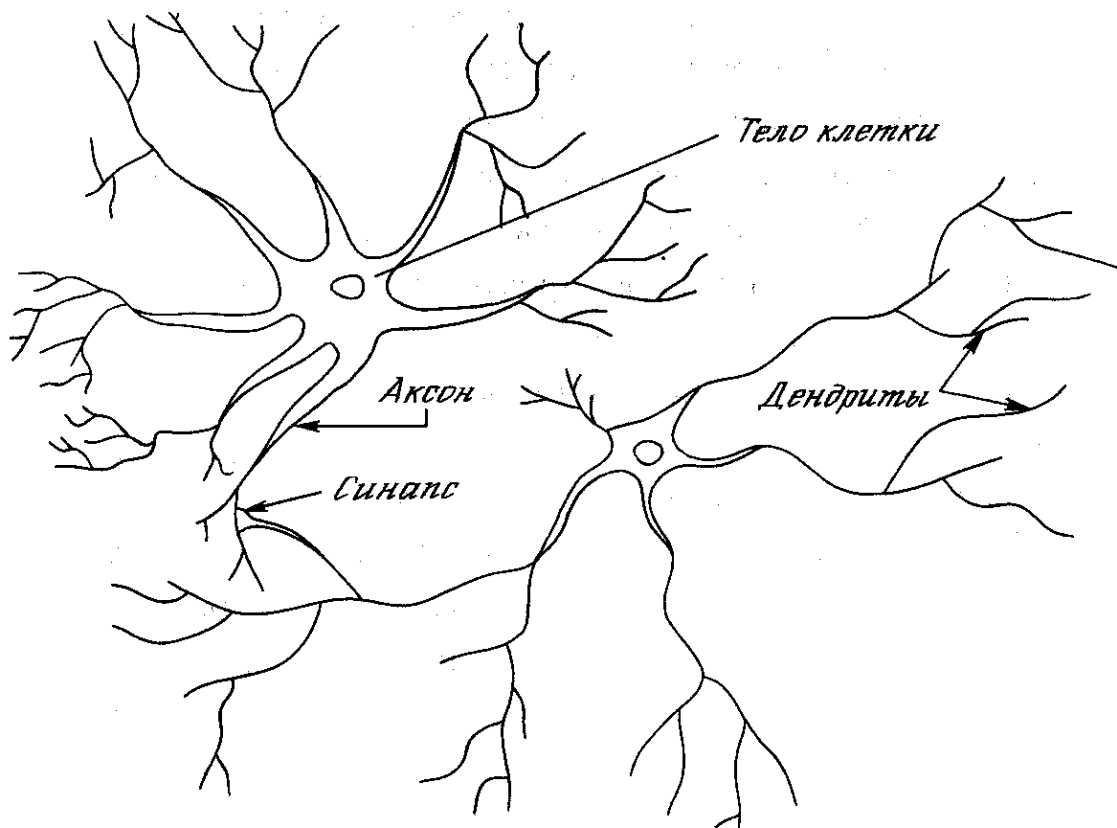


Рис. 1.1. Биологический нейрон

На рис. 1.1 показана структура пары типичных биологических нейронов. Дендриты идут от тела нервной клетки к другим нейронам, где они принимают сигналы в точках соединения, называемых синапсами. Принятые синапсом входные сигналы подводятся к телу нейрона. Здесь они суммируются, причем одни входы стремятся возбудить нейрон, другие – воспрепятствовать его возбуждению. Когда суммарное возбуждение в теле нейрона превышает некоторый порог, нейрон возбуждается, посылая по аксону сигнал другим нейронам. У этой основной функциональной схемы много усложнений и исключений, тем не менее большинство искусственных нейронных сетей моделируют лишь эти простые свойства.

## ИСКУССТВЕННЫЙ НЕЙРОН

Искусственный нейрон имитирует в первом приближении свойства биологического нейрона. На вход искусственного нейрона поступает некоторое множество сигналов, каждый из которых является выходом другого нейрона. Каждый вход умножается на соответствующий вес, аналогичный синаптической силе, и все произведения суммируются, определяя уровень активации нейрона. На рис. 1.2 представлена модель, реализующая эту идею. Хотя сетевые парадигмы весьма разнообразны, в основе почти всех их лежит эта конфигурация. Здесь множество входных сигналов, обозначенных  $x_1, x_2, \dots, x_n$ , поступает на искусственный нейрон. Эти входные сигналы, в совокупности обозначаемые вектором  $\mathbf{X}$ , соответствуют сигналам, приходящим в синапсы биологического нейрона. Каждый сигнал умножается на соответствующий вес  $w_1, w_2, \dots, w_n$  и поступает на суммирующий блок, обозначенный  $\Sigma$ . Каждый вес соответствует «силе» одной биологической синаптической связи. (Множество весов в совокупности обозначается вектором  $\mathbf{W}$ .) Суммирующий блок, соответствующий телу биологического элемента, складывает взвешенные входы алгебраически, создавая выход, который мы будем называть NET. В векторных обозначениях это может быть компактно записано следующим образом:

$$NET = \mathbf{XW}.$$

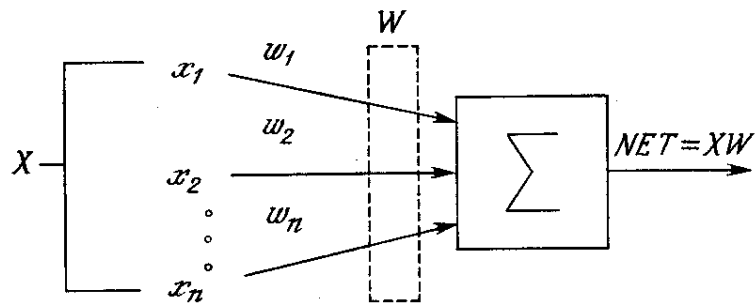


Рис. 1.2. Искусственный нейрон

### Активационные функции

Сигнал NET далее, как правило, преобразуется активационной функцией  $F$  и дает выходной нейронный сигнал OUT. Активационная функция может быть обычной линейной функцией

$$OUT = K(NET),$$

где  $K$  – постоянная, пороговой функции

$$\begin{aligned} OUT &= 1, \text{ если } NET > T, \\ OUT &= 0 \text{ в остальных случаях,} \end{aligned}$$

где  $T$  – некоторая постоянная пороговая величина, или же функцией, более точно моделирующей нелинейную передаточную характеристику биологического нейрона и представляющей нейронной сети большие возможности.

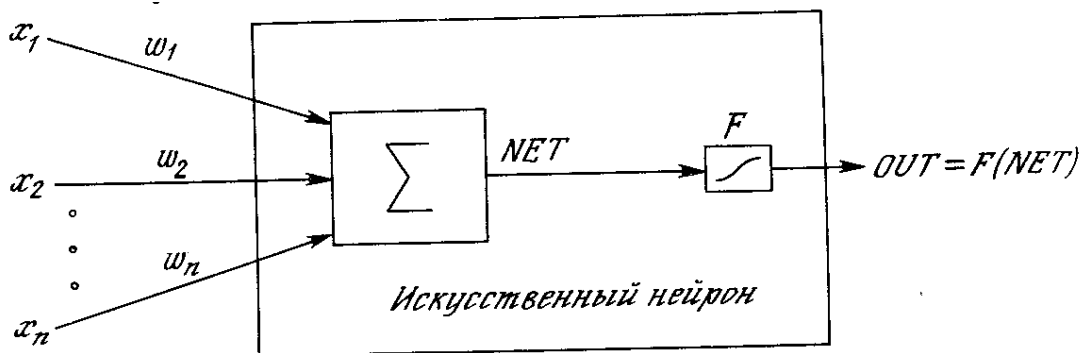


Рис. 1.3. Искусственный нейрон с активационной функцией

На рис. 1.3 блок, обозначенный  $F$ , принимает сигнал NET и выдает сигнал OUT. Если блок  $F$  сужает диапазон изменения величины NET так, что при любых значениях NET значения OUT принадлежат некоторому конечному интервалу, то  $F$  называется «сжимающей» функцией. В качестве «сжимающей» функции часто используется логистическая или «сигмоидальная» (S-образная) функция, показанная на рис. 1.4а. Эта функция математически выражается как  $F(x) = 1/(1 + e^{-x})$ . Таким образом,

$$OUT = \frac{1}{1 + e^{-NET}}.$$

По аналогии с электронными системами активационную функцию можно считать нелинейной усилительной характеристикой искусственного нейрона. Коэффициент усиления вычисляется как отношение приращения величины OUT к вызвавшему его небольшому приращению величины NET. Он выражается наклоном кривой при определенном уровне возбуждения и изменяется от малых значений при больших отрицательных возбуждениях (кривая почти горизонтальна) до максимального значения при нулевом возбуждении и снова уменьшается, когда возбуждение становится большим положительным. Гроссберг (1973) обнаружил, что подобная нелинейная характеристика решает поставленную им дилемму шумового насыщения. Каким образом одна и та же сеть может обрабатывать как слабые, так и сильные сигналы? Слабые сигналы нуждаются в большом сетевом усилении, чтобы дать пригодный к использованию выходной сигнал. Однако усилительные каскады с большими коэффициентами усиления могут привести к насыщению выхода шумами усилителей (случайными флуктуациями), которые присутствуют в любой физически реализованной сети. Сильные входные сигналы в свою очередь также будут приводить к

насыщению усилительных каскадов, исключая возможность полезного использования выхода. Центральная область логистической функции, имеющая большой коэффициент усиления, решает проблему обработки слабых сигналов, в то время как области с падающим усилением на положительном и отрицательном концах подходят для больших возбуждений. Таким образом, нейрон функционирует с большим усилением в широком диапазоне уровня входного сигнала.

$$OUT = \frac{1}{1 + e^{-NET}} = F(NET).$$

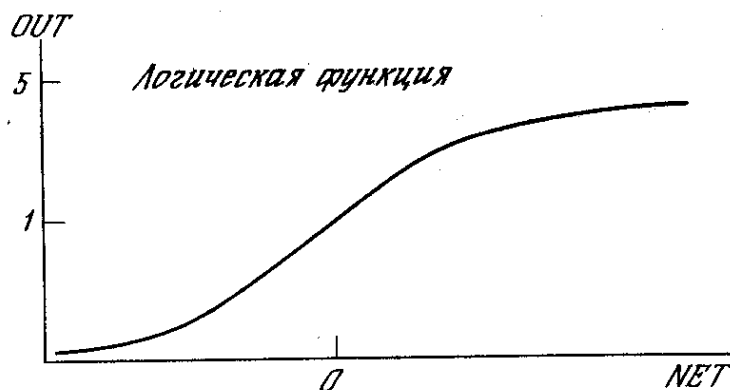


Рис. 1.4а. Сигмоидальная логистическая функция

Другой широко используемой активационной функцией является гиперболический тангенс. По форме она сходна с логистической функцией и часто используется биологами в качестве математической модели активации нервной клетки. В качестве активационной функции искусственной нейронной сети она записывается следующим образом:

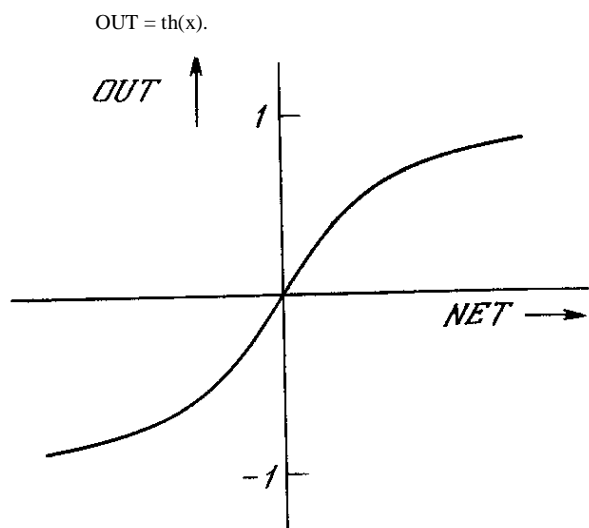


Рис. 1.4б. Функция гиперболического тангенса

Подобно логистической функции гиперболический тангенс является S-образной функцией, но он симметричен относительно начала координат, и в точке NET = 0 значение выходного сигнала OUT равно нулю (см. рис. 1.4б). В отличие от логистической функции гиперболический тангенс принимает значения различных знаков, что оказывается выгодным для ряда сетей (см. гл. 3).

Рассмотренная простая модель искусственного нейрона игнорирует многие свойства своего биологического двойника. Например, она не принимает во внимание задержки во времени, которые воздействуют на динамику системы. Входные сигналы сразу же порождают выходной сигнал. И, что более важно, она не учитывает воздействия функции частотной модуляции или синхронизирующей функции биологического нейрона, которые ряд исследователей считают решающими.

Несмотря на эти ограничения, сети, построенные из этих нейронов, обнаруживают свойства, сильно напоминающие биологическую систему. Только время и исследования смогут ответить на вопрос, являются ли подобные совпадения случайными или следствием того, что в модели верно схвачены важнейшие черты биологического нейрона.

## ОДНОСЛОЙНЫЕ ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ

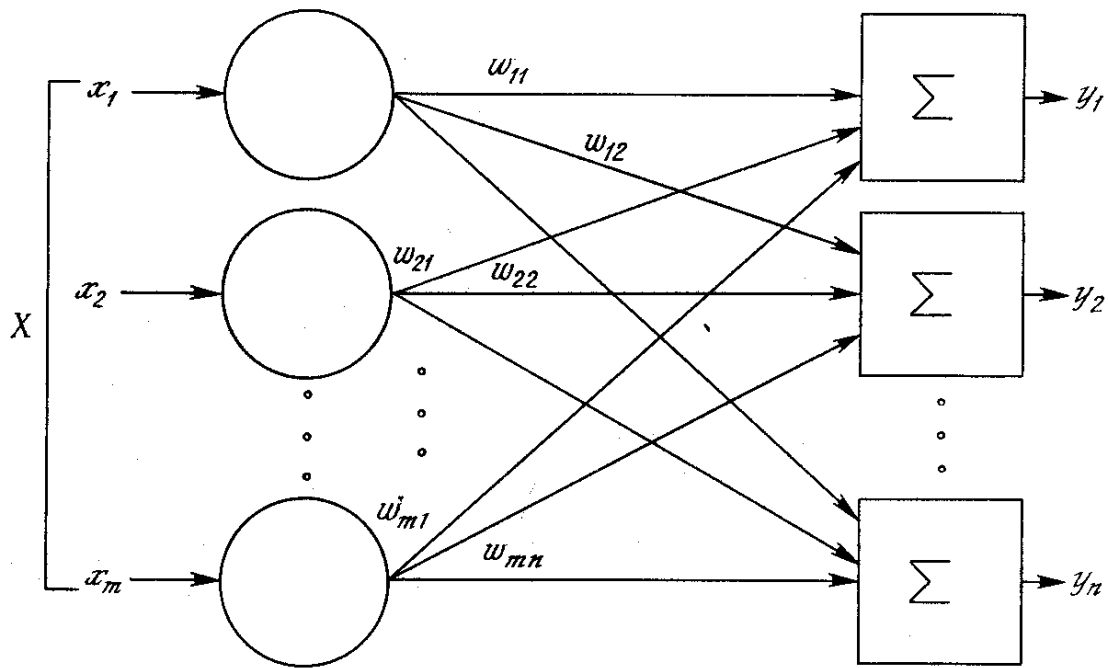


Рис. 1.5. Однослойная нейронная сеть

Хотя один нейрон и способен выполнять простейшие процедуры распознавания, сила нейронных вычислений проистекает от соединений нейронов в сетях. Простейшая сеть состоит из группы нейронов, образующих слой, как показано в правой части рис. 1.5. Отметим, что вершины-круги слева служат лишь для распределения входных сигналов. Они не выполняют каких-либо вычислений, и поэтому не будут считаться слоем. По этой причине они обозначены кругами, чтобы отличать их от вычисляющих нейронов, обозначенных квадратами. Каждый элемент из множества входов  $X$  отдельным весом соединен с каждым искусственным нейроном. А каждый нейрон выдает взвешенную сумму входов в сеть. В искусственных и биологических сетях многие соединения могут отсутствовать, все соединения показаны в целях общности. Могут иметь место также соединения между выходами и входами элементов в слое. Такие конфигурации рассматриваются в гл. 6.

Удобно считать веса элементами матрицы  $W$ . Матрица имеет  $m$  строк и  $n$  столбцов, где  $m$  – число входов, а  $n$  – число нейронов. Например,  $w_{2,3}$  – это вес, связывающий третий вход со вторым нейроном. Таким образом, вычисление выходного вектора  $N$ , компонентами которого являются выходы OUT нейронов, сводится к матричному умножению  $N = XW$ , где  $N$  и  $X$  – векторы-строки.

## МНОГОСЛОЙНЫЕ ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ

Более крупные и сложные нейронные сети обладают, как правило, и большими вычислительными возможностями. Хотя созданы сети всех конфигураций, какие только можно себе представить, послонная организация нейронов копирует слоистые структуры определенных отделов мозга. Оказалось, что такие многослойные сети обладают большими возможностями, чем однослойные (см. гл. 2), и в последние годы были разработаны алгоритмы для их обучения.

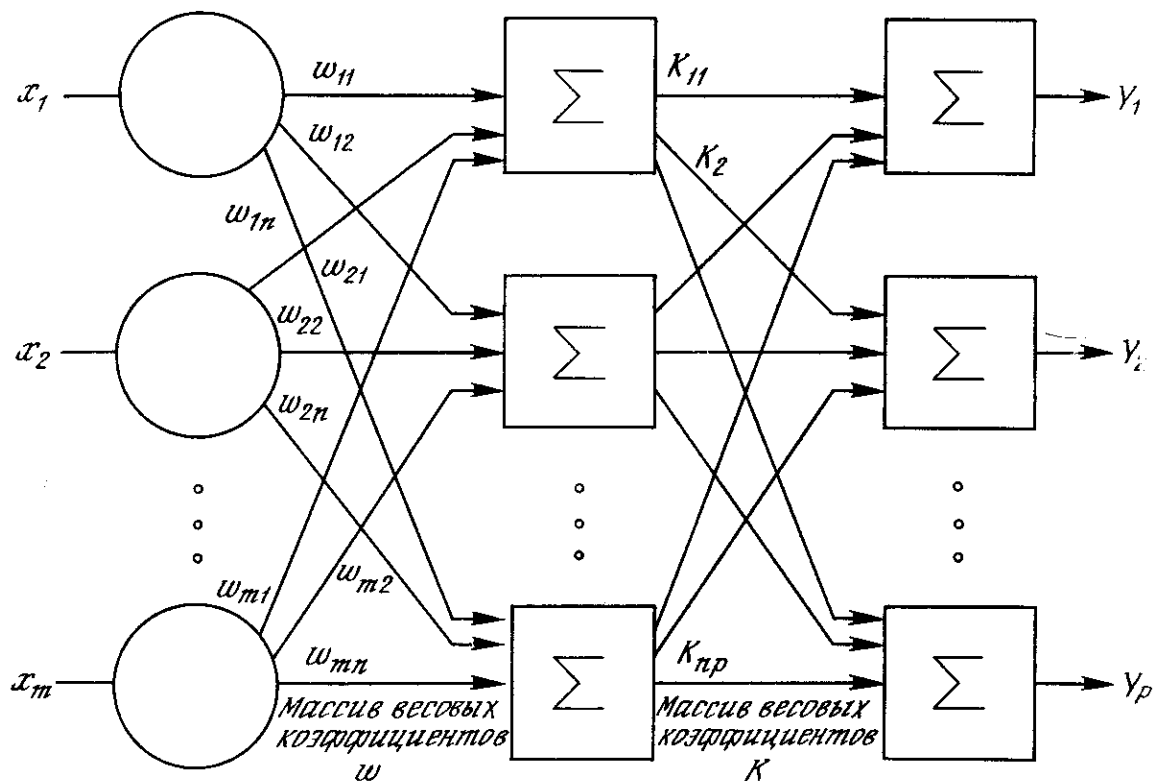


Рис. 1.6. Двухслойная нейронная сеть

Многослойные сети могут образовываться каскадами слоев. Выход одного слоя является входом для последующего слоя. Подобная сеть показана на рис. 1.6 и снова изображена со всеми соединениями.

### Нелинейная активационная функция

Многослойные сети не могут привести к увеличению вычислительной мощности по сравнению с однослойной сетью лишь в том случае, если активационная функция между слоями будет нелинейной. Вычисление выхода слоя заключается в умножении входного вектора на первую весовую матрицу с последующим умножением (если отсутствует нелинейная активационная функция) результирующего вектора на вторую весовую матрицу.

$$(XW_1)W_2$$

Так как умножение матриц ассоциативно, то

$$X(W_1W_2).$$

Это показывает, что двухслойная линейная сеть эквивалентна одному слою с весовой матрицей, равной произведению двух весовых матриц. Следовательно, любая многослойная линейная сеть может быть заменена эквивалентной однослойной сетью. В гл. 2 показано, что однослойные сети весьма ограничены по своим вычислительным возможностям. Таким образом, для расширения возможностей сетей по сравнению с однослойной сетью необходима нелинейная активационная функция.

### Сети с обратными связями

У сетей, рассмотренных до сих пор, не было обратных связей, т. е. соединений, идущих от выходов некоторого слоя к входам этого же слоя или предшествующих слоев. Этот специальный класс сетей, называемых сетями без обратных связей или сетями прямого распространения, представляет интерес и широко используется. Сети более общего вида, имеющие соединения от выходов к входам, называются сетями с обратными связями. У сетей без обратных связей нет памяти, их выход полностью определяется текущими входами и значениями весов. В некоторых конфигурациях сетей с обратными связями предыдущие значения выходов возвращаются на входы; выход, следовательно, определяется как текущим входом, так и предыдущими выходами. По этой причине сети с обратными связями могут обладать свойствами, сходными с кратковременной человеческой памятью, сетевые выходы частично зависят от предыдущих входов.

## **ТЕРМИНОЛОГИЯ, ОБОЗНАЧЕНИЯ И СХМАТИЧЕСКОЕ ИЗОБРАЖЕНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ**

К сожалению, для искусственных нейронных сетей еще нет опубликованных стандартов и устоявшихся терминов, обозначений и графических представлений. Порой идентичные сетевые парадигмы, представленные различными авторами, покажутся далекими друг от друга. В этой книге выбраны наиболее широко используемые термины.

### **Терминология**

Многие авторы избегают термина «нейрон» для обозначения искусственного нейрона, считая его слишком грубой моделью своего биологического прототипа. В этой книге термины «нейрон», «клетка», «элемент» используются взаимозаменяемо для обозначения «искусственного нейрона» как краткие и саморазъясняющие.

### **Дифференциальные уравнения или разностные уравнения**

Алгоритмы обучения, как и вообще искусственные нейронные сети, могут быть представлены как в дифференциальной, так и в конечно-разностной форме. При использовании дифференциальных уравнений предполагают, что процессы непрерывны и осуществляются подобно большой аналоговой сети. Для биологической системы, рассматриваемой на микроскопическом уровне, это не так. Активационный уровень биологического нейрона определяется средней скоростью, с которой он посылает дискретные потенциальные импульсы по своему аксону. Средняя скорость обычно рассматривается как аналоговая величина, но важно не забывать о действительном положении вещей.

Если моделировать искусственную нейронную сеть на аналоговом компьютере, то весьма желательно использовать представление с помощью дифференциальных уравнений. Однако сегодня большинство работ выполняется на цифровых компьютерах, что заставляет отдавать предпочтение конечно-разностной форме как наиболее легко программируемой. По этой причине на протяжении всей книги используется конечно-разностное представление.

### **Графическое представление**

Как видно из публикаций, нет общепринятого способа подсчета числа слоев в сети. Многослойная сеть состоит, как показано на рис. 1.6, из чередующихся множеств нейронов и весов. Ранее в связи с рис. 1.5 уже говорилось, что входной слой не выполняет суммирования. Эти нейроны служат лишь в качестве разветвлений для первого множества весов и не влияют на вычислительные возможности сети. По этой причине первый слой не принимается во внимание при подсчете слоев, и сеть, подобная изображенной на рис. 1.6, считается двухслойной, так как только два слоя выполняют вычисления. Далее, веса слоя считаются связанными со следующими за ними нейронами. Следовательно, слой состоит из множества весов со следующими за ними нейронами, суммирующими взвешенные сигналы.

### **Обучение искусственных нейронных сетей**

Среди всех интересных свойств искусственных нейронных сетей ни одно не захватывает так воображения, как их способность к обучению. Их обучение до такой степени напоминает процесс интеллектуального развития человеческой личности что может показаться, что достигнуто глубокое понимание этого процесса. Но проявляя осторожность, следует сдерживать эйфорию. Возможности обучения искусственных нейронных сетей ограничены, и нужно решить много сложных задач, чтобы определить, на правильном ли пути мы находимся. Тем не менее уже получены убедительные достижения, такие как «говорящая сеть» Сейновского (см. гл. 3), и возникает много других практических применений.

### **Цель обучения**

Сеть обучается, чтобы для некоторого множества входов давать желаемое (или, по крайней мере, сообразное с ним) множество выходов. Каждое такое входное (или выходное) множество рассматривается как вектор. Обучение осуществляется путем последовательного предъявления входных векторов с одновременной подстройкой весов в соответствии с определенной процедурой. В процессе обучения веса сети постепенно становятся такими, чтобы каждый входной вектор вырабатывал выходной вектор.

### **Обучение с учителем**

Различают алгоритмы обучения с учителем и без учителя. Обучение с учителем предполагает, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе они называются обучающей парой. Обычно сеть обучается на некотором числе таких обучающих пар. Предъявляется выходной вектор, вычисляется выход сети и сравнивается с соответствующим целевым вектором, разность (ошибка) с помощью обратной связи подается в сеть и веса изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Векторы обучающего множества предъявляются

последовательно, вычисляются ошибки и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемо низкого уровня.

### Обучение без учителя

Несмотря на многочисленные прикладные достижения, обучение с учителем критиковалось за свою биологическую неправдоподобность. Трудно вообразить обучающий механизм в мозге, который бы сравнивал желаемые и действительные значения выходов, выполняя коррекцию с помощью обратной связи. Если допустить подобный механизм в мозге, то откуда тогда возникают желаемые выходы? Обучение без учителя является намного более правдоподобной моделью обучения в биологической системе. Развита Кохоненом [3] и многими другими, она не нуждается в целевом векторе для выходов и, следовательно, не требует сравнения с предопределенными идеальными ответами. Обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы получались согласованные выходные векторы, т. е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения, следовательно, выделяет статистические свойства обучающего множества и группирует сходные векторы в классы. Предъявление на вход вектора из данного класса даст определенный выходной вектор, но до обучения невозможно предсказать, какой выход будет производиться данным классом входных векторов. Следовательно, выходы подобной сети должны трансформироваться в некоторую понятную форму, обусловленную процессом обучения. Это не является серьезной проблемой. Обычно не сложно идентифицировать связь между входом и выходом, установленную сетью.

### Алгоритмы обучения

Большинство современных алгоритмов обучения выросло из концепций Хэбба [2]. Им предложена модель обучения без учителя, в которой синаптическая сила (вес) возрастает, если активированы оба нейрона, источник и приемник. Таким образом, часто используемые пути в сети усиливаются и феномен привычки и обучения через повторение получает объяснение.

В искусственной нейронной сети, использующей обучение по Хэббу, наращивание весов определяется произведением уровней возбуждения передающего и принимающего нейронов. Это можно записать как

$$w_{ij}(n+1) = w(n) + \alpha \text{OUT}_i \text{OUT}_j,$$

где  $w_{ij}(n)$  – значение веса от нейрона  $i$  к нейрону  $j$  до подстройки,  $w_{ij}(n+1)$  – значение веса от нейрона  $i$  к нейрону  $j$  после подстройки,  $\alpha$  – коэффициент скорости обучения,  $\text{OUT}_i$  – выход нейрона  $i$  и вход нейрона  $j$ ,  $\text{OUT}_j$  – выход нейрона  $j$ .

Сети, использующие обучение по Хэббу, конструктивно развивались, однако за последние 20 лет были развиты более эффективные алгоритмы обучения. В частности, в работах [4 – 6] и многих других были развиты алгоритмы обучения с учителем, приводящие к сетям с более широким диапазоном характеристик обучающих входных образов и большими скоростями обучения, чем использующие простое обучение по Хэббу.

В настоящее время используется огромное разнообразие обучающих алгоритмов. Потребовалась бы значительно большая по объему книга, чем эта, для рассмотрения этого предмета полностью. Чтобы рассмотреть этот предмет систематически, если и не исчерпывающе, в каждой из последующих глав подробно описаны алгоритмы обучения для рассматриваемой в главе парадигмы. В дополнение в приложении Б представлен общий обзор, в определенной мере более обширный, хотя и не очень глубокий. В нем дан исторический контекст алгоритмов обучения, их общая таксономия, ряд преимуществ и ограничений. В силу необходимости это приведет к повторению части материала, оправданием ему служит расширение взгляда на предмет.

### ПРОЛОГ

В последующих главах представлены и проанализированы некоторые наиболее важные сетевые конфигурации и их алгоритмы обучения. Представленные парадигмы дают представление об искусстве конструирования сетей в целом, его прошлом и настоящем. Многие другие парадигмы при тщательном рассмотрении оказываются лишь их модификациями. Современное развитие нейронных сетей скорее эволюционно, чем революционно. Поэтому понимание представленных в данной книге парадигм позволит следить за прогрессом в этой быстро развивающейся области.

Упор сделан на интуитивные и алгоритмические, а не математические аспекты. Книга адресована скорее пользователю искусственных нейронных сетей, чем теоретику. Сообщается, следовательно, достаточно информации, чтобы дать читателю возможность понимать основные идеи. Те, кто знаком с программированием, смогут реализовать любую из этих сетей. Сложные математические выкладки опущены, если только они не имеют прямого отношения к реализации сети. Для заинтересованного читателя приводятся ссылки на более строгие и полные работы.



## Литература

1. Grossberg S. 1973. Contour enhancement, short-term memory, and consistencies in reverberating neural networks. *Studies in Applied Mathematics* 52:217,257.
2. Hebb D. O. 1961. *Organization of behavior*. New York: Science Edition.
3. Kohonen T. 1984. *Self-organization and associative memory*. Series in Information Sciences, vol. 8. Berlin: Springer Verlag.
4. Rosenblatt F. 1962. *Principles of neurodynamics*. New York: Spartan Books. (Русский перевод: Розенблатт Ф. Принципы нейродинамики. – М.: Мир., 1965.)
5. Widrow B. 1959. Adaptive sampled-data systems, a statistical theory of adaptation. 1959 IRE WESCON Convention Record, part 4, pp. 88-91. New York: Institute of Radio Engineers.
6. Widrow B., Hoff M. 1960. Adaptive switching circuits. 1960 IRE WESCON Convention Record, pp. 96-104. New York: Institute of Radio Engineers.

## Глава 2. Персептроны

### ПЕРСЕПТРОНЫ И ЗАРОЖДЕНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

В качестве научного предмета искусственные нейронные сети впервые заявили о себе в 40-е годы. Стремясь воспроизвести функции человеческого мозга, исследователи создали простые аппаратные (а позже программные) модели биологического нейрона и системы его соединений. Когда нейрофизиологи достигли более глубокого понимания нервной системы человека, эти ранние попытки стали восприниматься как весьма грубые аппроксимации. Тем не менее на этом пути были достигнуты впечатляющие результаты, стимулировавшие дальнейшие исследования, приведшие к созданию более изощренных сетей.

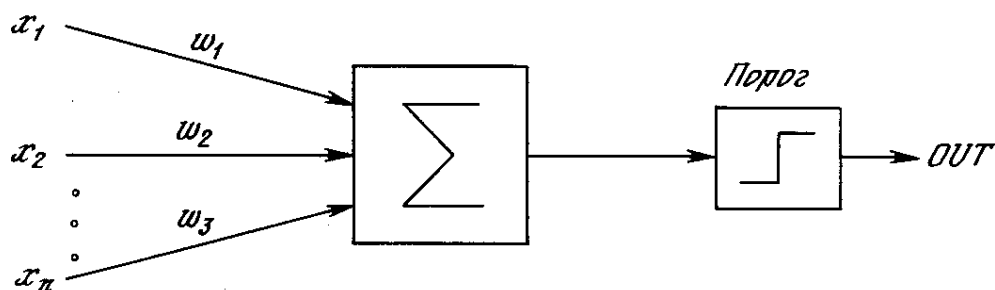


Рис. 2.1. Персептронный нейрон

Первое систематическое изучение искусственных нейронных сетей было предпринято Маккаллоком и Питтсом в 1943 г. [1]. Позднее в работе [3] они исследовали сетевые парадигмы для распознавания изображений, подвергаемых сдвигам и поворотам. Простая нейронная модель, показанная на рис. 2.1, использовалась в большей части их работы. Элемент  $\Sigma$  умножает каждый вход  $x$  на вес  $w$  и суммирует взвешенные входы. Если эта сумма больше заданного порогового значения, выход равен единице, в противном случае – нулю. Эти системы (и множество им подобных) получили название *персептронов*. Они состоят из одного слоя искусственных нейронов, соединенных с помощью весовых коэффициентов с множеством входов (см. рис. 2.2), хотя в принципе описываются и более сложные системы.

В 60-е годы персептроны вызвали большой интерес и оптимизм. Розенблатт [4] доказал замечательную теорему об обучении персептронов, объясняемую ниже. Уидроу [5-8] дал ряд убедительных демонстраций систем персептронного типа, и исследователи во всем мире стремились изучить возможности этих систем. Первоначальная эйфория сменилась разочарованием, когда оказалось, что персептроны не способны обучиться решению ряда простых задач. Минский [2] строго проанализировал эту проблему и показал, что имеются жесткие ограничения на то, что могут выполнять однослойные персептроны, и, следовательно, на то, чему они могут обучаться. Так как в то время методы обучения многослойных сетей не были известны, исследователи перешли в более многообещающие области, и исследования в области нейронных сетей пришли в упадок. Недавнее открытие методов обучения многослойных сетей в большей степени, чем какой-либо иной фактор, повлияло на возрождение интереса и исследовательских усилий.

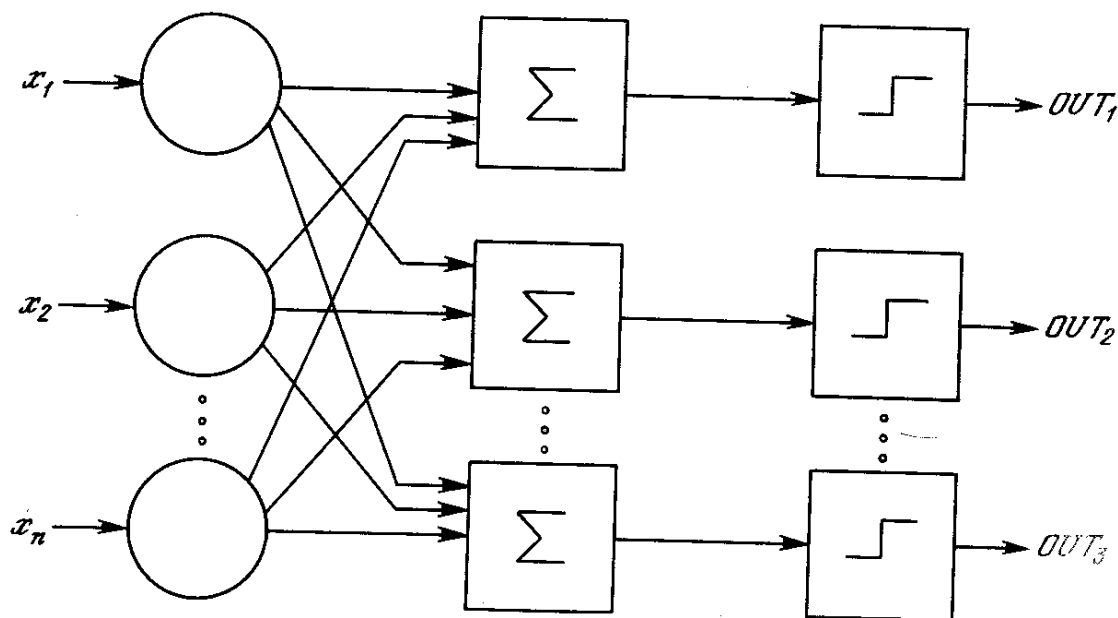


Рис. 2.2. Персептрон со многими выходами

Работа Минского, возможно, и охладила пыл энтузиастов персептрона, но обеспечила время для необходимой консолидации и развития лежащей в основе теории. Важно отметить, что анализ Минского не был опровергнут. Он остается важным исследованием и должен изучаться, чтобы ошибки 60-х годов не повторились.

Несмотря на свои ограничения персептроны широко изучались (хотя не слишком широко использовались). Теория персептронов является основой для многих других типов искусственных нейронных сетей, и персептроны иллюстрируют важные принципы. В силу этих причин они являются логической исходной точкой для изучения искусственных нейронных сетей.

### ПЕРСЕПТРОННАЯ ПРЕДСТАВЛЯЕМОСТЬ

Доказательство теоремы обучения персептрона [4] показало, что персептрон способен научиться всему, что он способен представлять. Важно при этом уметь различать представляемость и обучаемость. Понятие представляемости относится к способности персептрона (или другой сети) моделировать определенную функцию. Обучаемость же требует наличия систематической процедуры настройки весов сети для реализации этой функции.

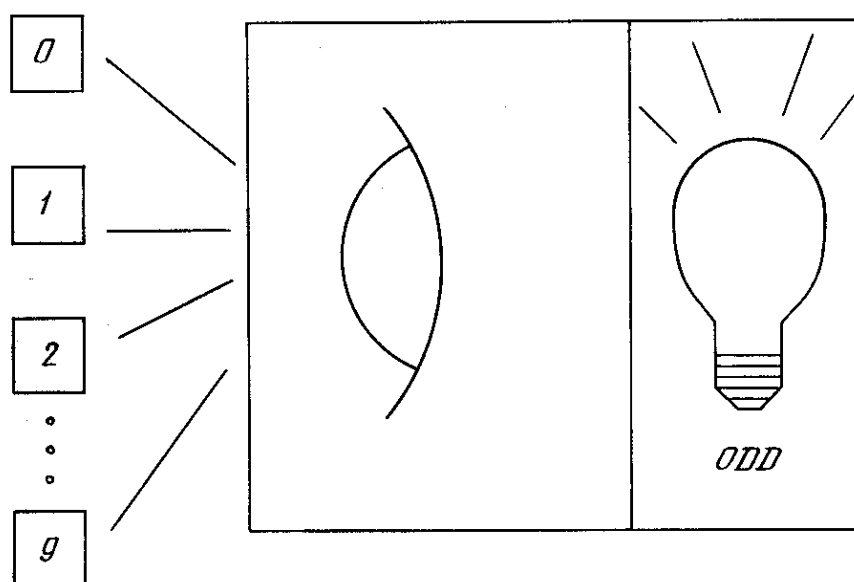


Рис. 2.3. Система распознавания изображений

Для иллюстрации проблемы представляемости допустим, что у нас есть множество карт, помеченных цифрами от 0 до 9. Допустим также, что мы обладаем гипотетической машиной, способной отличать карты с нечетным номером от карт с четным номером и зажигающей индикатор на своей панели при предъявлении карты с нечетным номером (см. рис. 2.3). Представима ли такая машина перцептроном? То есть может ли быть сконструирован перцептрон и настроены его веса (неважно каким образом) так, чтобы он обладал такой же разделяющей способностью? Если это так, то говорят, что перцептрон способен представлять желаемую машину. Мы увидим, что возможности представления однослойными перцептронами весьма ограничены. Имеется много простых машин, которые не могут быть представлены перцептроном независимо от того, как настраиваются его веса.

### Проблема функции ИСКЛЮЧАЮЩЕЕ ИЛИ

Один из самых пессимистических результатов Минского показывает, что однослойный перцептрон не может воспроизвести такую простую функцию, как ИСКЛЮЧАЮЩЕЕ ИЛИ. Это функция от двух аргументов, каждый из которых может быть нулем или единицей. Она принимает значение единицы, когда один из аргументов равен единице (но не оба). Проблему можно проиллюстрировать с помощью однослойной однонейронной системы с двумя входами, показанной на рис. 2.4. Обозначим один вход через  $x$ , а другой через  $y$ , тогда все их возможные комбинации будут состоять из четырех точек на плоскости  $x$ - $y$ , как показано на рис. 2.5. Например, точка  $x = 0$  и  $y = 0$  обозначена на рисунке как точка А Табл. 2.1 показывает требуемую связь между входами и выходом, где входные комбинации, которые должны давать нулевой выход, помечены  $A_0$  и  $A_1$ , единичный выход –  $B_0$  и  $B_1$ .

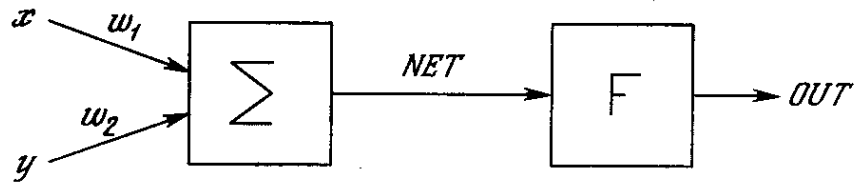


Рис. 2.4. Однонейронная система

В сети на рис. 2.4 функция  $F$  является обычным порогом, так что  $OUT$  принимает значение ноль, когда  $NET$  меньше 0,5, и единица в случае, когда  $NET$  больше или равно 0,5. Нейрон выполняет следующее вычисление:

$$NET = xw_1 + yw_2 \quad (2.1)$$

Никакая комбинация значений двух весов не может дать соотношения между входом и выходом, задаваемого табл. 2.1. Чтобы понять это ограничение, зафиксируем  $NET$  на величине порога 0,5. Сеть в этом случае описывается уравнением (2.2). Это уравнение линейно по  $x$  и  $y$ , т. е. все значения по  $x$  и  $y$ , удовлетворяющие этому уравнению, будут лежать на некоторой прямой в плоскости  $x$ - $y$ .

$$xw_1 + yw_2 = 0,5 \quad (2.2)$$

Таблица 2.1. Таблица истинности для функции ИСКЛЮЧАЮЩЕЕ ИЛИ

Точки	Значения $x$	Значения $y$	Требуемый выход
$A_0$	0	0	0
$B_0$	1	0	1
$B_1$	0	1	1
$A_1$	1	1	0

Любые входные значения для  $x$  и  $y$  на этой линии будут давать пороговое значение 0,5 для  $NET$ . Входные значения с одной стороны прямой обеспечат значения  $NET$  больше порога, следовательно,  $OUT=1$ . Входные значения по другую сторону прямой обеспечат значения  $NET$  меньше порогового значения, делая  $OUT$  равным 0. Изменения значений  $w_1$ ,  $w_2$  и порога будут менять наклон и положение прямой. Для того чтобы сеть реализовала функцию ИСКЛЮЧАЮЩЕЕ ИЛИ, заданную табл. 2.1, нужно расположить прямую так, чтобы точки А были с одной стороны прямой, а точки В – с другой. Попытавшись нарисовать такую прямую на рис. 2.5, убеждаемся, что это невозможно. Это означает, что какие бы значения ни приписывались весам и порогу, сеть неспособна воспроизвести соотношение между входом и выходом, требуемое для представления функции ИСКЛЮЧАЮЩЕЕ ИЛИ.

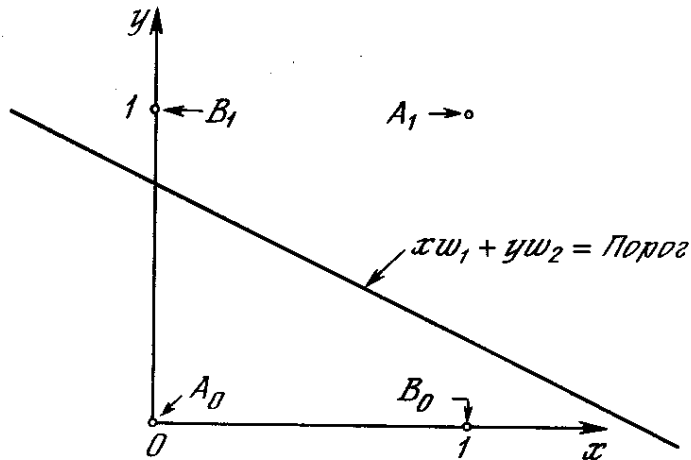


Рис. 2.5. Проблема ИСКЛЮЧАЮЩЕЕ ИЛИ

Взглянув на задачу с другой точки зрения, рассмотрим NET как поверхность над плоскостью  $x$ - $y$ . Каждая точка этой поверхности находится над соответствующей точкой плоскости  $x$ - $y$  на расстоянии, равном значению NET в этой точке. Можно показать, что наклон этой NET-поверхности одинаков для всей поверхности  $x$ - $y$ . Все точки, в которых значение NET равно величине порога, проектируются на линию уровня плоскости NET (см. рис. 2.6).

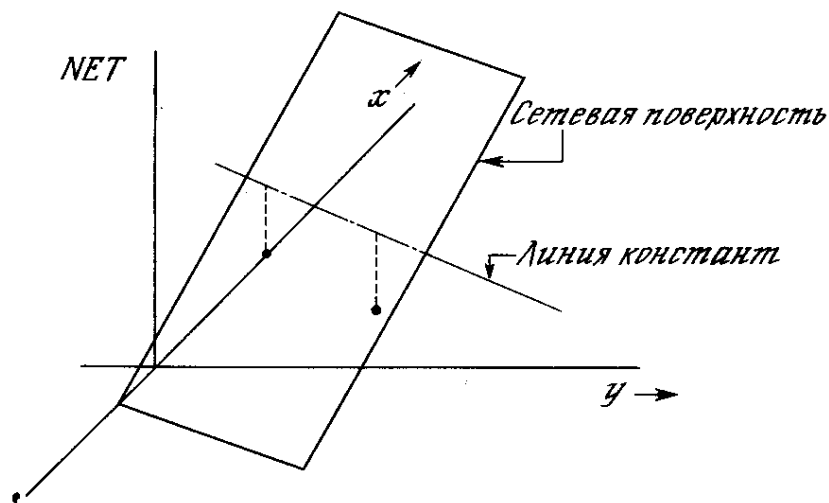


Рис. 2.6. Персептронная NET-плоскость

Ясно, что все точки по одну сторону пороговой прямой спроецируются в значения NET, большие порога, а точки по другую сторону дадут меньшие значения NET. Таким образом, пороговая прямая разбивает плоскость  $x$ - $y$  на две области. Во всех точках по одну сторону пороговой прямой значение OUT равно единице, по другую сторону – нулю.

### Линейная разделимость

Как мы видели, невозможно нарисовать прямую линию, разделяющую плоскость  $x$ - $y$  так, чтобы реализовывалась функция ИСКЛЮЧАЮЩЕЕ ИЛИ. К сожалению, этот пример не единственный. Имеется обширный класс функций, не реализуемых однослойной сетью. Об этих функциях говорят, что они являются линейно неразделимыми, и они накладывают определенные ограничения на возможности однослойных сетей.

Линейная разделимость ограничивает однослойные сети задачами классификации, в которых множества точек (соответствующих входным значениям) могут быть разделены геометрически. Для нашего случая с двумя входами разделитель является прямой линией. В случае трех входов разделение осуществляется плоскостью, рассекающей трехмерное пространство. Для четырех или более входов визуализация невозможна и необходимо мысленно представить  $n$ -мерное пространство, рассекаемое «гиперплоскостью» – геометрическим объектом, который рассекает пространство четырех или большего числа измерений.

Так как линейная разделимость ограничивает возможности перцептронного представления, то важно знать, является ли данная функция разделимой. К сожалению, не существует простого способа определить это, если число переменных велико.

Нейрон с  $n$  двоичными входами может иметь  $2^n$  различных входных образов, состоящих из нулей и единиц. Так как каждый входной образ может соответствовать двум различным бинарным выходам (единица и ноль), то всего имеется  $2^{2^n}$  функций от  $n$  переменных.

**Таблица 2.2. Линейно разделимые функции**

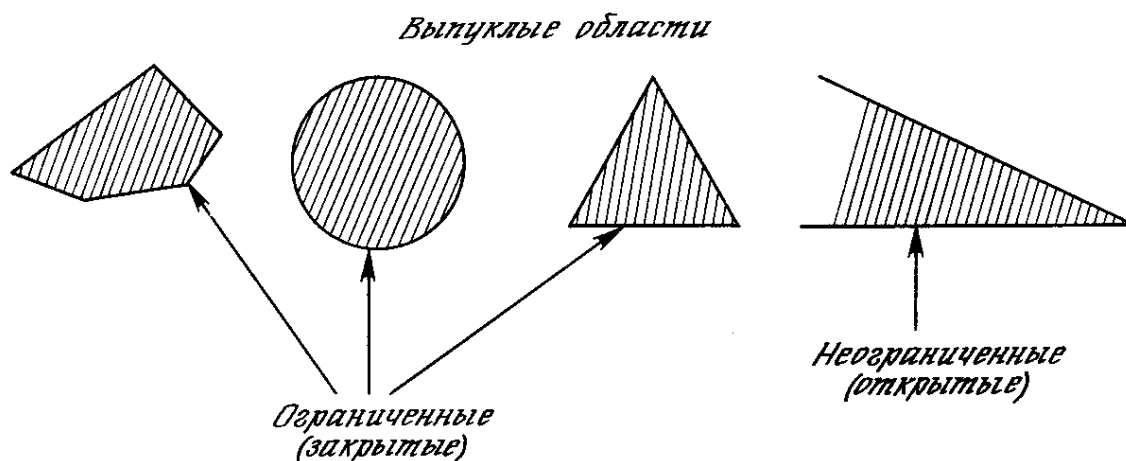
$n$	$2^{2^n}$	Число линейно разделимых функций
1	4	4
2	16	14
3	256	104
4	65536	1882
5	$4,3 \times 10^9$	94572
6	$1,8 \times 10^{19}$	15 028 134

(Взято из R. O. Winder, Single-stage logic. Paper presented at the AIEE Fall General Meeting, 1960.)

Как видно из табл. 2.2, вероятность того, что случайно выбранная функция окажется линейно разделимой, весьма мала даже для умеренного числа переменных. По этой причине однослойные перцептроны на практике ограничены простыми задачами.

### Преодоление ограничения линейной разделимости

К концу 60-х годов проблема линейной разделимости была хорошо понята. К тому же было известно, что это серьезное ограничение представляемости однослойными сетями можно преодолеть, добавив дополнительные слои. Например, двухслойные сети можно получить каскадным соединением двух однослойных сетей. Они способны выполнять более общие классификации, отделяя те точки, которые содержатся в выпуклых ограниченных или неограниченных областях. Область называется выпуклой, если для любых двух ее точек соединяющий их отрезок целиком лежит в области. Область называется ограниченной, если ее можно заключить в некоторый круг. Неограниченную область невозможно заключить внутри круга (например, область между двумя параллельными линиями). Примеры выпуклых ограниченных и неограниченных областей представлены на рис. 2.7.



**Рис. 1.7. Выпуклые ограниченные и неограниченные области**

Чтобы уточнить требование выпуклости, рассмотрим простую двухслойную сеть с двумя входами, подведенными к двум нейронам первого слоя, соединенными с единственным нейроном в слое 2 (см. рис. 2.8). Пусть порог выходного нейрона равен 0,75, а оба его веса равны 0,5. В этом случае для того, чтобы порог был превышен и на выходе появилась единица, требуется, чтобы оба нейрона первого уровня на выходе имели единицу. Таким образом, выходной нейрон реализует логическую функцию И. На рис. 2.8 каждый нейрон слоя 1 разбивает плоскость  $x$ - $y$  на две полуплоскости, один обеспечивает единичный выход для входов ниже верхней линии, другой – для входов выше нижней линии. На рис. 2.8 показан результат такого двойного разбиения, где выходной сигнал нейрона второго слоя равен единице только внутри  $V$ -образной области. Аналогично во втором слое может быть

использовано три нейрона с дальнейшим разбиением плоскости и созданием области треугольной формы. Включением достаточного числа нейронов во входной слой может быть образован выпуклый многоугольник любой желаемой формы. Так как они образованы с помощью операции И над областями, задаваемыми линиями, то все такие многогранники выпуклы, следовательно, только выпуклые области и возникают. Точки, не составляющие выпуклой области, не могут быть отделены от других точек плоскости двухслойной сетью.

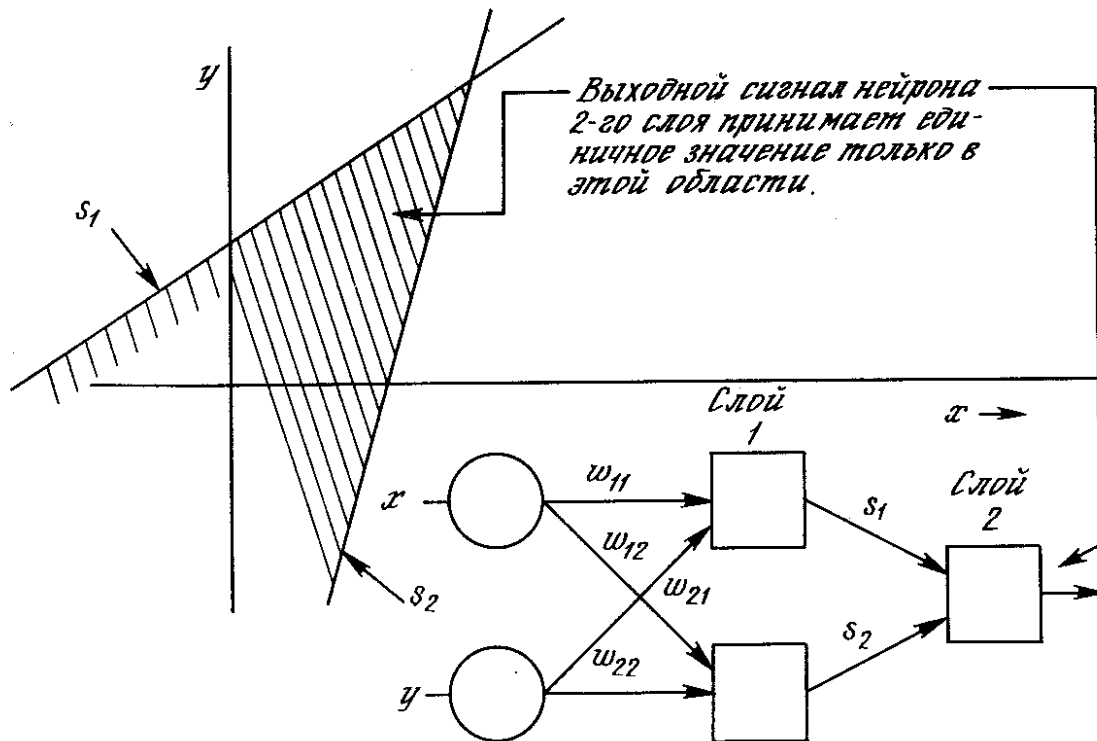


Рис. 1.8. Выпуклая область решений, задаваемая двухслойной сетью

Нейрон второго слоя не ограничен функцией И. Он может реализовывать многие другие функции при подходящем выборе весов и порога. Например, можно сделать так, чтобы единичный выход любого из нейронов первого слоя приводил к появлению единицы на выходе нейрона второго слоя, реализовав тем самым логическое ИЛИ. Имеется 16 двоичных функций от двух переменных. Если выбирать подходящим образом веса и порог, то можно воспроизвести 14 из них (все, кроме ИСКЛЮЧАЮЩЕЕ ИЛИ и ИСКЛЮЧАЮЩЕЕ НЕТ).

Входы не обязательно должны быть двоичными. Вектор непрерывных входов может представлять собой произвольную точку на плоскости  $x$ - $y$ . В этом случае мы имеем дело со способностью сети разбивать плоскость на непрерывные области, а не с разделением дискретных множеств точек. Для всех этих функций, однако, линейная разделимость показывает, что выход нейрона второго слоя равен единице только в части плоскости  $x$ - $y$ , ограниченной многоугольной областью. Поэтому для разделения плоскостей  $P$  и  $Q$  необходимо, чтобы все  $P$  лежали внутри выпуклой многоугольной области, не содержащей точек  $Q$  (или наоборот).

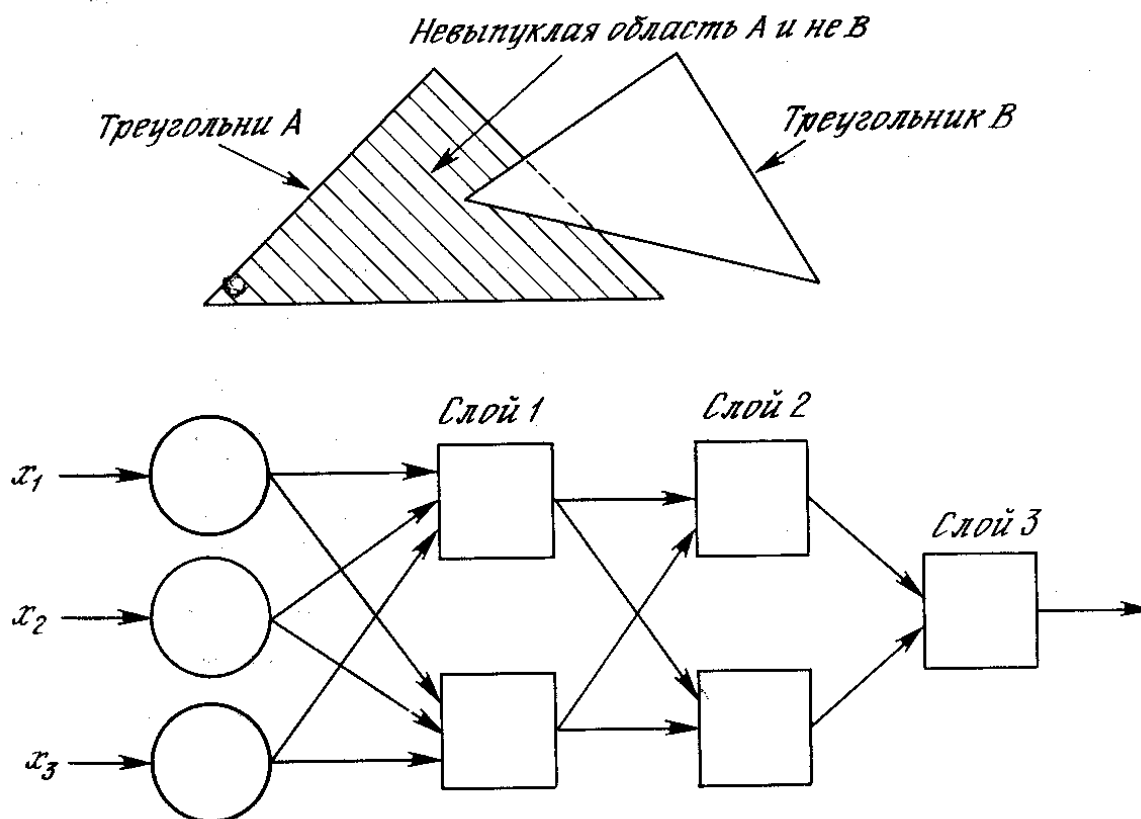


Рис. 2.9. «Вогнутая» область решений, задаваемая трехслойной сетью

Трехслойная сеть, однако, является более общей. Ее классифицирующие возможности ограничены лишь числом искусственных нейронов и весов. Ограничения на выпуклость отсутствуют. Теперь нейрон третьего слоя принимает в качестве входа набор выпуклых многоугольников, и их логическая комбинация может быть невыпуклой. На рис. 2.9 иллюстрируется случай, когда два треугольника А и В, скомбинированные с помощью функций «А и не В», задают невыпуклую область. При добавлении нейронов и весов число сторон многоугольников может неограниченно возрастать. Это позволяет аппроксимировать область любой формы с любой точностью. Вдобавок не все выходные области второго слоя должны пересекаться. Возможно, следовательно, объединять различные области, выпуклые и невыпуклые, выдавая на выходе единицу всякий раз, когда входной вектор принадлежит одной из них.

Несмотря на то что возможности многослойных сетей были известны давно, в течение многих лет не было теоретически обоснованного алгоритма для настройки их весов. В последующих главах мы детально изучим многослойные обучающие алгоритмы, но сейчас достаточно понимать проблему и знать, что исследования привели к определенным результатам.

### Эффективность запоминания

Серьезные вопросы имеются относительно эффективности запоминания информации в персептроне (или любых других нейронных сетях) по сравнению с обычной компьютерной памятью и методами поиска информации в ней. Например, в компьютерной памяти можно хранить все входные образы вместе с классифицирующими битами. Компьютер должен найти требуемый образ и дать его классификацию. Различные хорошо известные методы могли бы быть использованы для ускорения поиска. Если точное соответствие не найдено, то для ответа может быть использовано правило ближайшего соседа.

Число битов, необходимое для хранения этой же информации в весах персептрона, может быть значительно меньшим по сравнению с методом обычной компьютерной памяти, если образы допускают экономичную запись. Однако Минский [2] построил патологические примеры, в которых число битов, требуемых для представления весов, растет с размерностью задачи быстрее, чем экспоненциально. В этих случаях требования к памяти с ростом размерности задачи быстро становятся невыполнимыми. Если, как он предположил, эта ситуация не является исключением, то персептроны часто могут быть ограничены только малыми задачами. Насколько общими являются такие неподатливые множества образов? Это остается открытым вопросом, относящимся ко всем нейронным сетям. Поиски ответа чрезвычайно важны для исследований по нейронным сетям.

## **ОБУЧЕНИЕ ПЕРСЕПТРОНА**

Способность искусственных нейронных сетей обучаться является их наиболее интригующим свойством. Подобно биологическим системам, которые они моделируют, эти нейронные сети сами моделируют себя в результате попыток достичь лучшей модели поведения.

Используя критерий линейной разделимости, можно решить, способна ли однослойная нейронная сеть реализовывать требуемую функцию. Даже в том случае, когда ответ положительный, это принесет мало пользы, если у нас нет способа найти нужные значения для весов и порогов. Чтобы сеть представляла практическую ценность, нужен систематический метод (алгоритм) для вычисления этих значений. Розенблатт [4] сделал это в своем алгоритме обучения персептрона вместе с доказательством того, что персептрон может быть обучен всему, что он может реализовывать.

Обучение может быть с учителем или без него. Для обучения с учителем нужен «внешний» учитель, который оценивал бы поведение системы и управлял ее последующими модификациями. При обучении без учителя, рассматриваемого в последующих главах, сеть путем самоорганизации делает требуемые изменения. Обучение персептрона является обучением с учителем.

Алгоритм обучения персептрона может быть реализован на цифровом компьютере или другом электронном устройстве, и сеть становится в определенном смысле самоподстраивающейся. По этой причине процедуру подстройки весов обычно называют «обучением» и говорят, что сеть «обучается». Доказательство Розенблатта стало основной вехой и дало мощный импульс исследованиям в этой области. Сегодня в той или иной форме элементы алгоритма обучения персептрона встречаются во многих сетевых парадигмах.

## **АЛГОРИТМ ОБУЧЕНИЯ ПЕРСЕПТРОНА**

Персептрон обучают, подавая множество образов по одному на его вход и подстраивая веса до тех пор, пока для всех образов не будет достигнут требуемый выход. Допустим, что входные образы нанесены на демонстрационные карты. Каждая карта разбита на квадраты и от каждого квадрата на персептрон подается вход. Если в квадрате имеется линия, то от него подается единица, в противном случае – ноль. Множество квадратов на карте задает, таким образом, множество нулей и единиц, которое и подается на входы персептрона. Цель состоит в том, чтобы научить персептрон включать индикатор при подаче на него множества входов, задающих нечетное число, и не включать в случае четного.



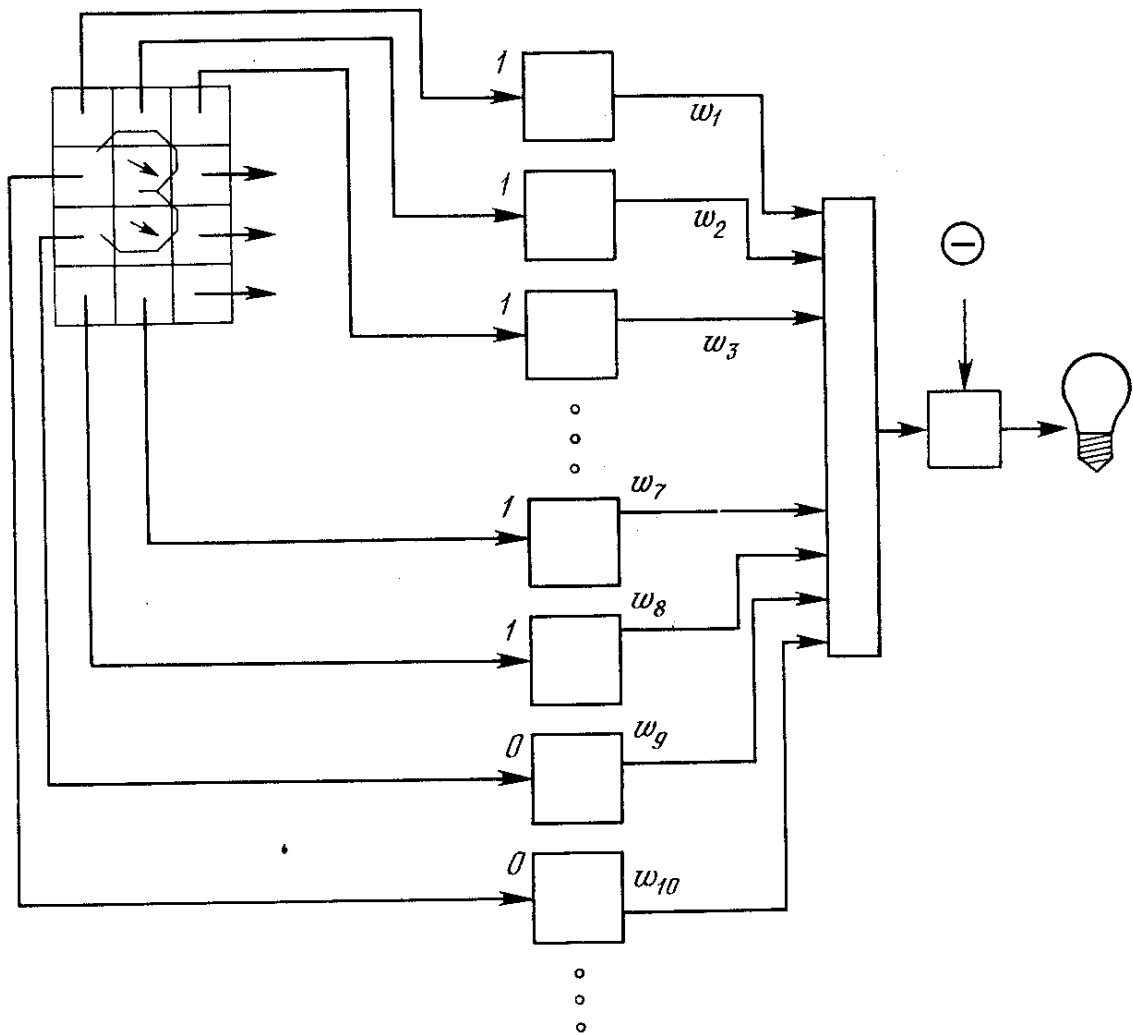


Рис. 1.10. Персептронная система распознавания изображений

На рис. 2.10 показана такая персептронная конфигурация. Допустим, что вектор  $X$  является образом распознаваемой демонстрационной карты. Каждая компонента (квадрат)  $X = (x_1, x_2, \dots, x_n)$  — умножается на соответствующую компоненту вектора весов  $W = (w_1, w_2, \dots, w_n)$ . Эти произведения суммируются. Если сумма превышает порог  $\Theta$ , то выход нейрона  $Y$  равен единице (индикатор загорается), в противном случае он — ноль. Как мы видели в гл. 1, эта операция компактно записывается в векторной форме как  $Y = XW$ , а после нее следует пороговая операция.

Для обучения сети образ  $X$  подается на вход и вычисляется выход  $Y$ . Если  $Y$  правилен, то ничего не меняется. Однако если выход неправилен, то веса, присоединенные к входам, усиливающим ошибочный результат, модифицируются, чтобы уменьшить ошибку.

Чтобы увидеть, как это осуществляется, допустим, что демонстрационная карта с цифрой 3 подана на вход и выход  $Y$  равен 1 (показывая нечетность). Так как это правильный ответ, то веса не изменяются. Если, однако, на вход подается карта с номером 4 и выход  $Y$  равен единице (нечетный), то веса, присоединенные к единичным входам, должны быть уменьшены, так как они стремятся дать неверный результат. Аналогично, если карта с номером 3 дает нулевой выход, то веса, присоединенные к единичным входам, должны быть увеличены, чтобы скорректировать ошибку.

Этот метод обучения может быть подытожен следующим образом:

1. Подать входной образ и вычислить  $Y$ .
- 2 а. Если выход правильный, то перейти на шаг 1;
- б. Если выход неправильный и равен нулю, то добавить все входы к соответствующим им весам; или
- в. Если выход неправильный и равен единице, то вычесть каждый вход из соответствующего ему веса.
3. Перейти на шаг 1.

За конечное число шагов сеть научится разделять карты на четные и нечетные при условии, что множество цифр линейно разделяемо. Это значит, что для всех нечетных карт выход будет больше порога, а для всех четных — меньше. Отметим, что

это обучение глобально, т. е. сеть обучается на всем множестве карт. Возникает вопрос о том, как это множество должно предъявляться, чтобы минимизировать время обучения. Должны ли элементы множества предъявляться- последовательно друг за другом или карты следует выбирать случайно? Несложная теория служит здесь путеводителем.

### Дельта-правило

Важное обобщение алгоритма обучения персептрона, называемое дельта-правилом, переносит этот метод на непрерывные входы и выходы. Чтобы понять, как оно было получено, шаг 2 алгоритма обучения персептрона может быть сформулирован в обобщенной форме с помощью введения величины  $\delta$ , которая равна разности между требуемым или целевым выходом  $T$  и реальным выходом  $Y$

$$\delta = (T - Y). \quad (2.3)$$

Случай, когда  $\delta=0$ , соответствует шагу 2а, когда выход правилен и в сети ничего не изменяется. Шаг 2б соответствует случаю  $\delta > 0$ , а шаг 2в случаю  $\delta < 0$ .

В любом из этих случаев персептронный алгоритм обучения сохраняется, если  $\delta$  умножается на величину каждого входа  $x_i$  и это произведение добавляется к соответствующему весу. С целью обобщения вводится коэффициент «скорости обучения»  $\eta$ , который умножается на  $\delta x_i$ , что позволяет управлять средней величиной изменения весов.

В алгебраической форме записи

$$\Delta_i = \eta \delta x_i, \quad (2.4)$$

$$w_i(n+1) = w_i(n) + \Delta_i, \quad (2.5)$$

где  $\Delta_i$  – коррекция, связанная с  $i$ -м входом  $x_i$ ;  $w_i(n+1)$  – значение веса  $i$  после коррекции;  $w_i\{n\}$  -значение веса  $i$  до коррекции.

Дельта-правило модифицирует веса в соответствии с требуемым и действительным значениями выхода каждой полярности как для непрерывных, так и для бинарных входов и выходов. Эти свойства открыли множество новых приложений.

### Трудности с алгоритмом обучения персептрона

Может оказаться затруднительным определить, выполнено ли условие разделимости для конкретного обучающего множества. Кроме того, во многих встречающихся на практике ситуациях входы часто меняются во времени и могут быть разделимы в один момент времени и неразделимы в другой. В доказательстве алгоритма обучения персептрона ничего не говорится также о том, сколько шагов требуется для обучения сети. Мало утешительного в знании того, что обучение закончится за конечное число шагов, если необходимое для этого время сравнимо с геологической эпохой. Кроме того, не доказано, что персептронный алгоритм обучения более быстр по сравнению с простым перебором всех возможных значений весов, и в некоторых случаях этот примитивный подход может оказаться лучше.

На эти вопросы никогда не находилось удовлетворительного ответа, они относятся к природе обучающего материала. В различной форме они возникают в последующих главах, где рассматриваются другие сетевые парадигмы. Ответы для современных сетей как правило не более удовлетворительны, чем для персептрона. Эти проблемы являются важной областью современных исследований.

### Литература

1. McCulloch W. W., Pitts W. 1943. A logical calculus of the ideas imminent in nervous activiti. Bulletin of Mathematical Biophysics 5:115-33. (Русский перевод: Маккаллоу У. С., Питтс У. Логическое исчисление идей, относящихся к нервной деятельности. Автоматы. – М.: ИЛ. – 1956.
2. Minsky M. L, Papert S. 1969. Perceptrons. Cambridge, MA: MIT Press. (Русский перевод: Минский М. Л., Пейперт С. Персептроны. – М: Мир. – 1971.)
3. Pitts W. Mcculloch W. W. 1947. How we know universals. Bulletin of Mathematical Biophysics 9:127-47.
4. Rosenblatt F. 1962. Principles of Neurodynamics. New York: Spartan Books. (Русский перевод: Розенблатт Ф. Принципы нейродинамики. – М: Мир. – 1965.)
5. Widrow B. 1961. The speed of adaptation in adaptive control system, paper \*1933-61. American Rocket Society Guidance Control and Navigation Conference.
6. Widrow B. 1963. A statistical theory of adaptation. Adaptive control systems. New York: Pergamon Press.
7. Widrow B., Angell J. B. 1962. Reliable, trainable networks for computing and control. Aerospace Engineering 21:78-123.

8. Widrow B., Hoff M. E. 1960. Adaptive switching circuits. 1960 IRE WESCON Convention Record, part 4, pp. 96-104. New York: Institute of Radio Engineers.

### Глава 3. Процедура обратного распространения

#### ВВЕДЕНИЕ В ПРОЦЕДУРУ ОБРАТНОГО РАСПРОСТРАНЕНИЯ

Долгое время не было теоретически обоснованного алгоритма для обучения многослойных искусственных нейронных сетей. А так как возможности представления с помощью однослойных нейронных сетей оказались весьма ограниченными, то и вся область в целом пришла в упадок.

Разработка алгоритма обратного распространения сыграла важную роль в возрождении интереса к искусственным нейронным сетям. Обратное распространение – это систематический метод для обучения многослойных искусственных нейронных сетей. Он имеет солидное математическое обоснование. Несмотря на некоторые ограничения, процедура обратного распространения сильно расширила область проблем, в которых могут быть использованы искусственные нейронные сети, и убедительно продемонстрировала свою мощь.

Интересна история разработки процедуры. В [7] было дано ясное и полное описание процедуры. Но как только эта работа была опубликована, оказалось, что она была предвосхищена в [4]. А вскоре выяснилось, что еще раньше метод был описан в [12]. Авторы работы [7] сэкономили бы свои усилия, зная о работе [12]. Хотя подобное дублирование является обычным явлением для каждой научной области, в искусственных нейронных сетях положение с этим намного серьезнее из-за пограничного характера самого предмета исследования. Исследования по нейронным сетям публикуются в столь различных книгах и журналах, что даже самому квалифицированному исследователю требуются значительные усилия, чтобы быть осведомленным о всех важных работах в этой области.

#### ОБУЧАЮЩИЙ АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ

##### Сетевые конфигурации

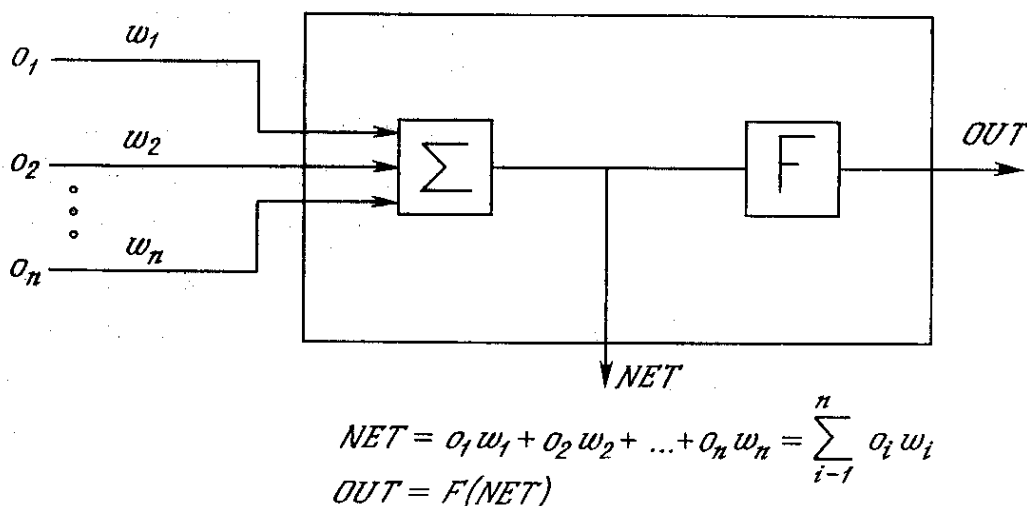


Рис. 3.1. Искусственный нейрон с активационной функцией

Нейрон. На рис. 3.1 показан нейрон, используемый в качестве основного строительного блока в сетях обратного распространения. Подается множество входов, идущих либо извне, либо от предшествующего слоя. Каждый из них умножается на вес, и произведения суммируются. Эта сумма, обозначаемая NET, должна быть вычислена для каждого нейрона сети. После того, как величина NET вычислена, она модифицируется с помощью активационной функции и получается сигнал OUT.

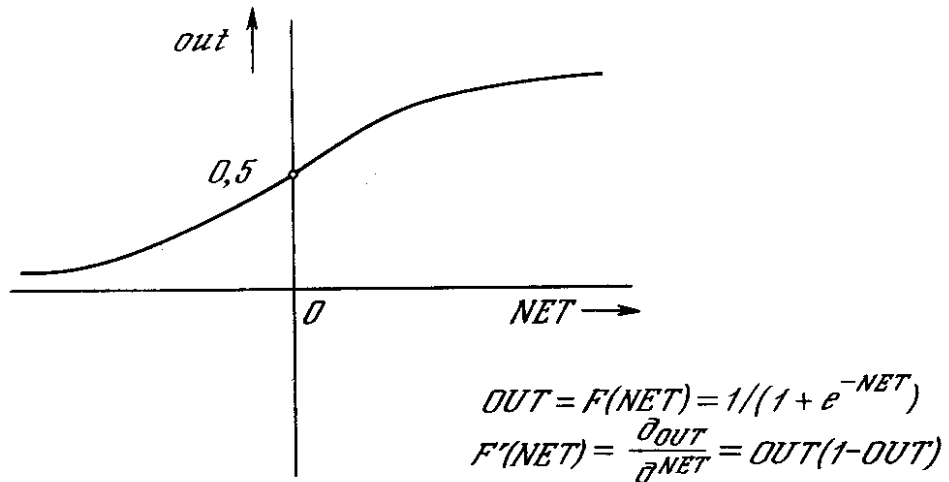


Рис. 3.2. Сигмоидальная активационная функция.

На рис. 3.2 показана активационная функция, обычно используемая для обратного распространения.

$$OUT = \frac{1}{1 + e^{-NET}} \quad (3.1)$$

Как показывает уравнение (3.2), эта функция, называемая *сигмоидом*, весьма удобна, так как имеет простую производную, что используется при реализации алгоритма обратного распространения.

$$\frac{\partial OUT}{\partial NET} = OUT(1 - OUT) \quad (3.2)$$

Сигмоид, который иногда называется также *логистической*, или *сжимающей функцией*, сужает диапазон изменения NET так, что значение OUT лежит между нулем и единицей. Как указывалось выше, многослойные нейронные сети обладают большей представляющей мощностью, чем однослойные, только в случае присутствия нелинейности. Сжимающая функция обеспечивает требуемую нелинейность.

В действительности имеется множество функций, которые могли бы быть использованы. Для алгоритма обратного распространения требуется лишь, чтобы функция была всюду дифференцируема. Сигмоид удовлетворяет этому требованию. Его дополнительное преимущество состоит в автоматическом контроле усиления. Для слабых сигналов (величина NET близка к нулю) кривая вход-выход имеет сильный наклон, дающий большое усиление. Когда величина сигнала становится больше, усиление падает. Таким образом, большие сигналы воспринимаются сетью без насыщения, а слабые сигналы проходят по сети без чрезмерного ослабления.

### Многослойная сеть.

На рис. 3.3 изображена многослойная сеть, которая может обучаться с помощью процедуры обратного распространения. (Для ясности рисунок упрощен.) Первый слой нейронов (соединенный с входами) служит лишь в качестве распределительных точек, суммирования входов здесь не производится. Входной сигнал просто проходит через них к весам на их выходах. А каждый нейрон последующих слоев выдает сигналы NET и OUT, как описано выше.

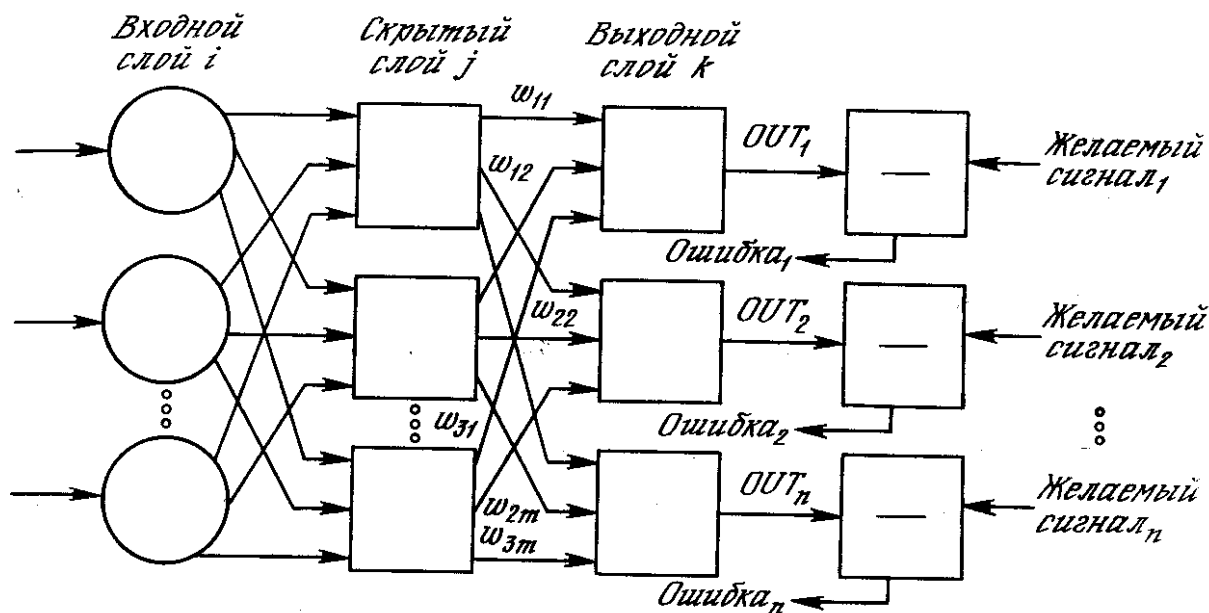


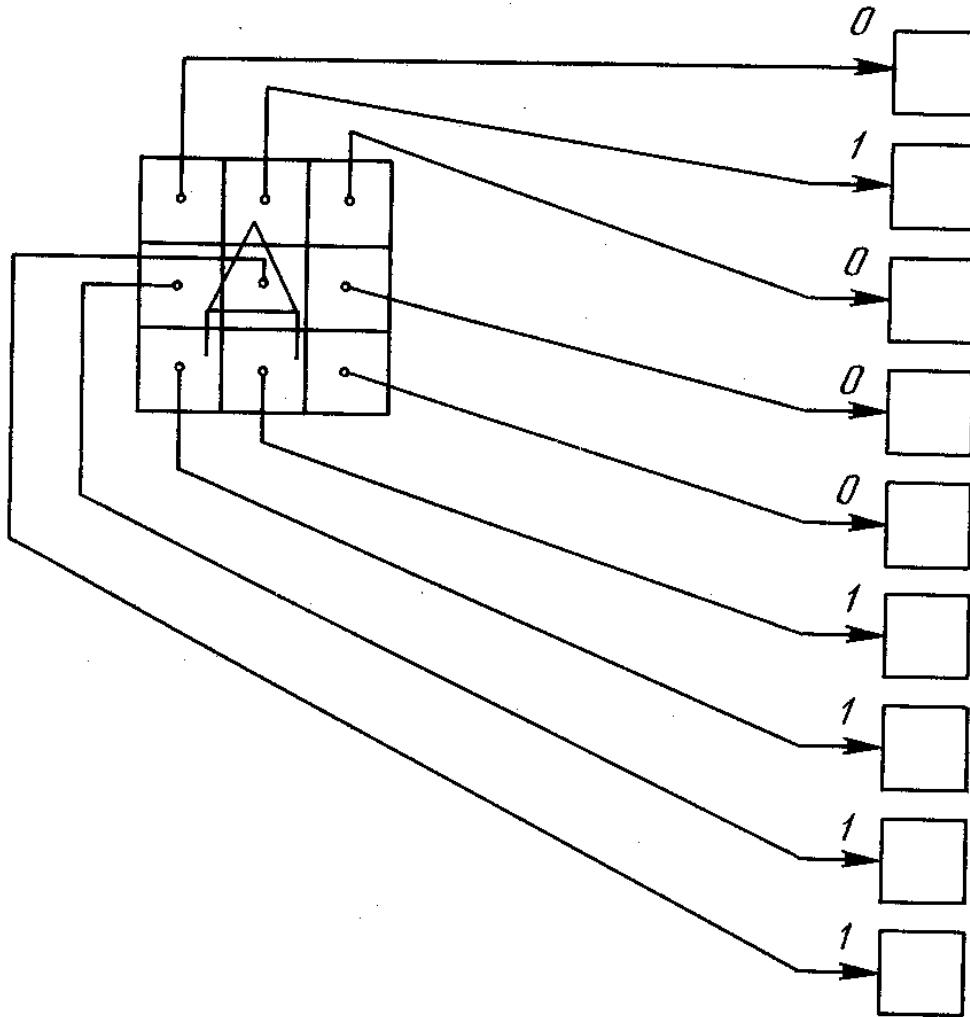
Рис. 3.3. Двухслойная сеть обратного распространения ( $\epsilon$  – желаемый сигнал).

В литературе нет единообразия относительно того, как считать число слоев в таких сетях. Одни авторы используют число слоев нейронов (включая несуммирующий входной слой), другие – число слоев весов. Так как последнее определение функционально описательное, то оно будет использоваться на протяжении книги. Согласно этому определению, сеть на рис. 3.3 рассматривается как двухслойная. Нейрон объединен с множеством весов, присоединенных к его входу. Таким образом, веса первого слоя оканчиваются на нейронах первого слоя. Вход распределительного слоя считается нулевым слоем.

Процедура обратного распространения применима к сетям с любым числом слоев. Однако для того, чтобы продемонстрировать алгоритм, достаточно двух слоев. Сейчас будут рассматриваться лишь сети прямого действия, хотя обратное распространение применимо и к сетям с обратными связями. Эти случаи будут рассмотрены в данной главе позднее.

## ОБЗОР ОБУЧЕНИЯ

Целью обучения сети является такая подстройка ее весов, чтобы приложение некоторого множества входов приводило к требуемому множеству выходов. Для краткости эти множества входов и выходов будут называться *векторами*. При обучении предполагается, что для каждого входного вектора существует парный ему целевой вектор, задающий требуемый выход. Вместе они называются *обучающей парой*. Как правило, сеть обучается на многих парах. Например, входная часть обучающей пары может состоять из набора нулей и единиц, представляющего двоичный образ некоторой буквы алфавита. На рис. 3.4 показано множество входов для буквы «А», нанесенной на сетке. Если через квадрат проходит линия, то соответствующий нейронный вход равен единице, в противном случае он равен нулю. Выход может быть числом, представляющим букву «А», или другим набором из нулей и единиц, который может быть использован для получения выходного образа. При необходимости распознавать с помощью сети все буквы алфавита, потребовалось бы 26 обучающих пар. Такая группа обучающих пар называется *обучающим множеством*.



**Рис. 3.4. Распознавание изображений**

Перед началом обучения всем весам должны быть присвоены небольшие начальные значения, выбранные случайным образом. Это гарантирует, что в сети не произойдет насыщения большими значениями весов, и предотвращает ряд других патологических случаев. Например, если всем весам придать одинаковые начальные значения, а для требуемого функционирования нужны неравные значения, то сеть не сможет обучиться.

Обучение сети обратного распространения требует выполнения следующих операций:

1. Выбрать очередную обучающую пару из обучающего множества; подать входной вектор на вход сети.
2. Вычислить выход сети.
3. Вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары).
4. Подкорректировать веса сети так, чтобы минимизировать ошибку.
5. Повторять шаги с 1 по 4 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

Операции, выполняемые шагами 1 и 2, сходны с теми, которые выполняются при функционировании уже обученной сети, т. е. подается входной вектор и вычисляется получающийся выход. Вычисления выполняются послойно. На рис. 3.3 сначала вычисляются выходы нейронов слоя  $j$ , затем они используются в качестве входов слоя  $k$ , вычисляются выходы нейронов слоя  $k$ , которые и образуют выходной вектор сети.

На шаге 3 каждый из выходов сети, которые на рис. 3.3 обозначены OУТ, вычитается из соответствующей компоненты целевого вектора, чтобы получить ошибку. Эта ошибка используется на шаге 4 для коррекции весов сети, причем знак и величина изменений весов определяются алгоритмом обучения (см. ниже).

После достаточного числа повторений этих четырех шагов разность между действительными выходами и целевыми выходами должна уменьшиться до приемлемой величины, при этом говорят, что сеть обучилась. Теперь сеть используется для распознавания и веса не изменяются.

На шаги 1 и 2 можно смотреть как на «проход вперед», так как сигнал распространяется по сети от входа к выходу. Шаги 3, 4 составляют «обратный проход», здесь вычисляемый сигнал ошибки распространяется обратно по сети и используется для подстройки весов. Эти два прохода теперь будут детализированы и выражены в более математической форме.

**Проход вперед.** Шаги 1 и 2 могут быть выражены в векторной форме следующим образом: подается входной вектор  $\mathbf{X}$  и на выходе получается вектор  $\mathbf{Y}$ . Векторная пара вход-цель  $\mathbf{X}$  и  $\mathbf{T}$  берется из обучающего множества. Вычисления проводятся над вектором  $\mathbf{X}$ , чтобы получить выходной вектор  $\mathbf{Y}$ .

Как мы видели, вычисления в многослойных сетях выполняются слой за слоем, начиная с ближайшего к входу слоя. Величина NET каждого нейрона первого слоя вычисляется как взвешенная сумма входов нейрона. Затем активационная функция  $F$  «сжимает» NET и дает величину OUT для каждого нейрона в этом слое. Когда множество выходов слоя получено, оно является входным множеством для следующего слоя. Процесс повторяется слой за слоем, пока не будет получено заключительное множество выходов сети.

Этот процесс может быть выражен в сжатой форме с помощью векторной нотации. Веса между нейронами могут рассматриваться как матрица  $\mathbf{W}$ . Например, вес от нейрона 8 в слое 2 к нейрону 5 слоя 3 обозначается  $w_{8,5}$ . Тогда NET-вектор слоя  $\mathbf{N}$  может быть выражен не как сумма произведений, а как произведение  $\mathbf{X}$  и  $\mathbf{W}$ . В векторном обозначении  $\mathbf{N} = \mathbf{XW}$ . Покомпонентным применением функции  $F$  к NET-вектору  $\mathbf{N}$  получается выходной вектор  $\mathbf{O}$ . Таким образом, для данного слоя вычислительный процесс описывается следующим выражением:

$$\mathbf{O} = F(\mathbf{XW}). \quad (3.3)$$

Выходной вектор одного слоя является входным вектором для следующего, поэтому вычисление выходов последнего слоя требует применения уравнения (3.3) к каждому слою от входа сети к ее выходу.

**Обратный проход.** *Подстройка весов выходного слоя.* Так как для каждого нейрона выходного слоя задано целевое значение, то подстройка весов легко осуществляется с использованием модифицированного дельта-правила из гл. 2. Внутренние слои называют «скрытыми слоями», для их выходов не имеется целевых значений для сравнения. Поэтому обучение усложняется.

На рис. 3.5 показан процесс обучения для одного веса от нейрона  $p$  в скрытом слое  $j$  к нейрону  $q$  в выходном слое  $k$ . Выход нейрона слоя  $k$ , вычитаясь из целевого значения (Target), дает сигнал ошибки. Он умножается на производную сжимающей функции  $[\text{OUT}(1 - \text{OUT})]$ , вычисленную для этого нейрона слоя  $k$ , давая, таким образом, величину  $\delta$ .

$$\delta = \text{OUT}(1 - \text{OUT})(\text{Target} - \text{OUT}) \quad (3.4)$$

Затем  $\delta$  умножается на величину OUT нейрона  $j$ , из которого выходит рассматриваемый вес. Это произведение в свою очередь умножается на коэффициент скорости обучения  $\eta$  (обычно от 0,01 до 1,0), и результат прибавляется к весу. Такая же процедура выполняется для каждого веса от нейрона скрытого слоя к нейрону в выходном слое.

Следующие уравнения иллюстрируют это вычисление:

$$\Delta w_{pq,k} = \eta \delta_{q,k} \text{OUT} \quad (3.5)$$

$$w_{pq,k}(n+1) = w_{pq,k}(n) + \Delta w_{pq,k} \quad (3.6)$$

где  $w_{pq,k}(n)$  – величина веса от нейрона  $p$  в скрытом слое к нейрону  $q$  в выходном слое на шаге  $n$  (до коррекции); отметим, что индекс  $k$  относится к слою, в котором заканчивается данный вес, т. е., согласно принятому в этой книге соглашению, с которым он объединен;  $w_{pq,k}(n+1)$  – величина веса на шаге  $n + 1$  (после коррекции);  $\delta_{q,k}$  – величина  $\delta$  для нейрона  $q$ , в выходном слое  $k$ ;  $\text{OUT}_{pj}$  – величина OUT для нейрона  $p$  в скрытом слое  $j$ .

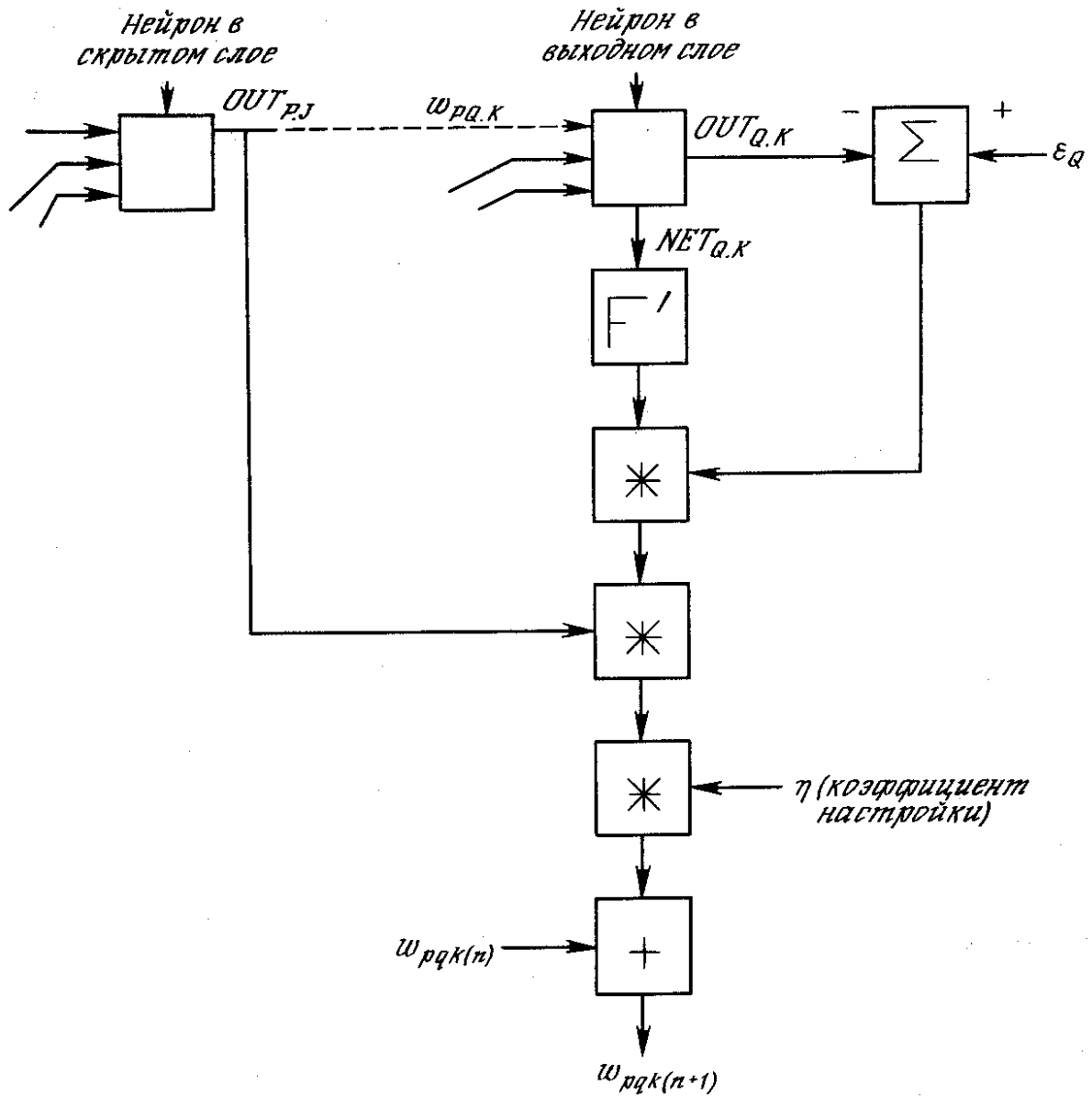


Рис. 3.5. Настройка веса в выходном слое

**Подстройка весов скрытого слоя.** Рассмотрим один нейрон в скрытом слое, предшествующем выходному слою. При проходе вперед этот нейрон передает свой выходной сигнал нейронам в выходном слое через соединяющие их веса. Во время обучения эти веса функционируют в обратном порядке, пропуская величину  $\delta$  от выходного слоя назад к скрытому слою. Каждый из этих весов умножается на величину  $\delta$  нейрона, к которому он присоединен в выходном слое. Величина  $\delta$ , необходимая для нейрона скрытого слоя, получается суммированием всех таких произведений и умножением на производную сжимающей функции:

$$\delta_{q,k} = \text{OUT}_{p,j} (1 - \text{OUT}_{p,j}) \left[ \sum_q \delta_{q,k} w_{pq,k} \right] \quad (3.7)$$

(см. рис. 3.6). Когда значение  $\delta$  получено, веса, питающие первый скрытый уровень, могут быть подкорректированы с помощью уравнений (3.5) и (3.6), где индексы модифицируются в соответствии со слоем.



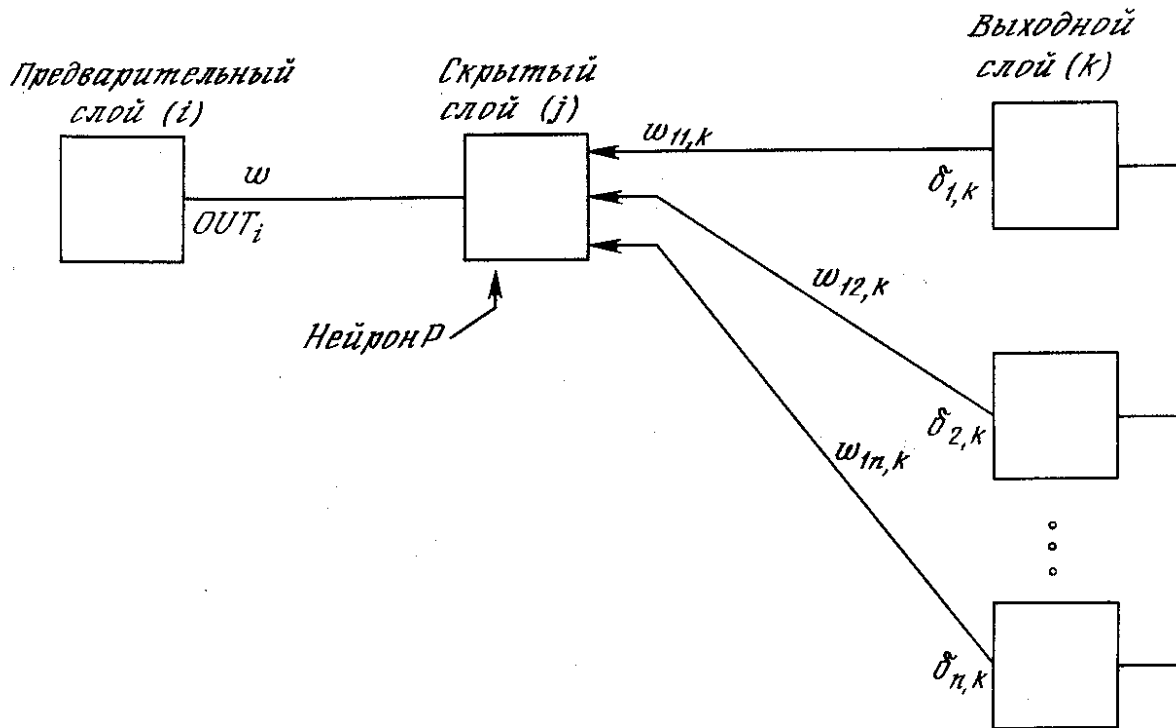


Рис. 3.6. Настройка веса в скрытом слое

Для каждого нейрона в данном скрытом слое должно быть вычислено  $\delta$  и подстроены все веса, ассоциированные с этим слоем. Этот процесс повторяется слой за слоем по направлению к входу, пока все веса не будут подкорректированы.

С помощью векторных обозначений операция обратного распространения ошибки может быть записана значительно компактнее. Обозначим множество величин  $\delta$  выходного слоя через  $\mathbf{D}_k$  и множество весов выходного слоя как массив  $\mathbf{W}_k$ . Чтобы получить  $\mathbf{D}_j$ ,  $\delta$ -вектор выходного слоя, достаточно следующих двух операций:

1. Умножить  $\delta$ -вектор выходного слоя  $\mathbf{D}_k$  на транспонированную матрицу весов  $\mathbf{W}'_k$ , соединяющую скрытый уровень с выходным уровнем.
2. Умножить каждую компоненту полученного произведения на производную сжимающей функции соответствующего нейрона в скрытом слое.

В символьной записи

$$\mathbf{D}_j = \mathbf{D}_k \mathbf{W}'_k \mathcal{S}[\mathbf{O}_j \mathcal{S}[\mathbf{I} - \mathbf{O}_j]], \quad (3.8)$$

где оператор  $\mathcal{S}$  в данной книге обозначает покомпонентное произведение векторов,  $\mathbf{O}_j$  – выходной вектор слоя  $j$  и  $\mathbf{I}$  – вектор, все компоненты которого равны 1.

**Добавление нейронного смещения.** Во многих случаях желательно наделять каждый нейрон обучаемым смещением. Это позволяет сдвигать начало отсчета логистической функции, давая эффект, аналогичный подстройке порога перцептронного нейрона, и приводит к ускорению процесса обучения. Эта возможность может быть легко введена в обучающий алгоритм с помощью добавляемого к каждому нейрону веса, присоединенного к +1. Этот вес обучается так же, как и все остальные веса, за исключением того, что подаваемый на него сигнал всегда равен +1, а не выходу нейрона предыдущего слоя.

**Импульс.** В работе [7] описан метод ускорения обучения для алгоритма обратного распространения, увеличивающий также устойчивость процесса. Этот метод, названный импульсом, заключается в добавлении к коррекции веса члена, пропорционального величине предыдущего изменения веса. Как только происходит коррекция, она «запоминается» и служит для модификации всех последующих коррекций. Уравнения коррекции модифицируются следующим образом:

$$\Delta w_{pq,k}(n+1) = \eta \delta_{q,k} \text{OUT}_{p,j} + \alpha \Delta w_{pq,k}(n) \quad (3.9)$$

$$w_{pq,k}(n+1) = w_{pq,k}(n) + \Delta w_{pq,k}(n+1) \quad (3.10)$$

где  $\alpha$  – коэффициент импульса, обычно устанавливается около 0,9.

Используя метод импульса, сеть стремится идти по дну узких оврагов поверхности ошибки (если таковые имеются), а не двигаться от склона к склону. Этот метод, по-видимому, хорошо работает на некоторых задачах, но дает слабый или даже отрицательный эффект на других.

В работе [8] описан сходный метод, основанный на экспоненциальном сглаживании, который может иметь преимущество в ряде приложений.

$$\Delta w_{pq,k}(n+1) = (1-\alpha) \delta_{q,k} \text{OUT}_{pj} + \alpha \Delta w_{pq,k}(n) \quad (3.9)$$

Затем вычисляется изменение веса

$$w_{pq,k}(n+1) = w_{pq,k}(n) + \eta \Delta w_{pq,k}(n+1), \quad (3.10)$$

где  $\alpha$  коэффициент сглаживания, варьируемый в диапазоне от 0,0 до 1,0. Если  $\alpha$  равен 1,0, то новая коррекция игнорируется и повторяется предыдущая. В области между 0 и 1 коррекция веса сглаживается величиной, пропорциональной  $\alpha$ . По-прежнему,  $\eta$  является коэффициентом скорости обучения, служащим для управления средней величиной изменения веса.

## ДАЛЬНЕЙШИЕ АЛГОРИТМИЧЕСКИЕ РАЗРАБОТКИ

Многими исследователями были предложены улучшения и обобщения описанного выше основного алгоритма обратного распространения. Литература в этой области слишком обширна, чтобы ее можно было здесь охватить. Кроме того, сейчас еще слишком рано давать окончательные оценки. Некоторые из этих подходов могут оказаться действительно фундаментальными, другие же со временем исчезнут. Некоторые из наиболее многообещающих разработок обсуждаются в этом разделе.

В [5] описан метод ускорения сходимости алгоритма обратного распространения. Названный обратным распространением второго порядка, он использует вторые производные для более точной оценки требуемой коррекции весов. В [5] показано, что этот алгоритм оптимален в том смысле, что невозможно улучшить оценку, используя производные более высокого порядка. Метод требует дополнительных вычислений по сравнению с обратным распространением первого порядка, и необходимы дальнейшие эксперименты для доказательства оправданности этих затрат.

В [9] описан привлекательный метод улучшения характеристик обучения сетей обратного распространения. В работе указывается, что общепринятый от 0 до 1 динамический диапазон входов и выходов скрытых нейронов неоптимален. Так как величина коррекции веса  $\Delta w_{pq,k}$  пропорциональна выходному уровню нейрона, порождающего  $\text{OUT}_{pj}$ , то нулевой уровень ведет к тому, что вес не меняется. При двоичных входных векторах половина входов в среднем будет равна нулю, и веса, с которыми они связаны, не будут обучаться! Решение состоит в приведении входов к значениям  $\pm 1/2$  и добавлении смещения к сжимающей функции, чтобы она также принимала значения  $\pm 1/2$ . Новая сжимающая функция выглядит следующим образом:

$$\text{OUT} = -\frac{1}{2} + \frac{1}{1 + e^{-\text{NET}}} \quad (3.13)$$

С помощью таких простых средств время сходимости сокращается в среднем от 30 до 50%. Это является одним из примеров практической модификации, существенно улучшающей характеристику алгоритма.

В [6] и [1] описана методика применения обратного распространения к сетям с обратными связями, т. е. к таким сетям, у которых выходы подаются через обратную связь на входы. Как показано в этих работах, обучение в подобных системах может быть очень быстрым и критерии устойчивости легко удовлетворяются.

## ПРИМЕНЕНИЯ

Обратное распространение было использовано в широкой сфере прикладных исследований. Некоторые из них описываются здесь, чтобы продемонстрировать мощь этого метода.

Фирма NEC в Японии объявила недавно, что обратное распространение было ею использовано для визуального распознавания букв, причем точность превысила 99%. Это улучшение было достигнуто с помощью комбинации обычных алгоритмов с сетью обратного распространения, обеспечивающей дополнительную проверку.

В работе [8] достигнут впечатляющий успех с Net-Talk, системой, которая превращает печатный английский текст в высококачественную речь. Магнитофонная запись процесса обучения сильно напоминает звуки ребенка на разных этапах обучения речи.

В [2] обратное распространение использовалось в машинном распознавании рукописных английских слов. Буквы, нормализованные по размеру, наносились на сетку, и брались проекции линий, пересекающих квадраты сетки. Эти проекции

служили затем входами для сети обратного распространения. Сообщалось о точности 99,7% при использовании словарного фильтра.

В [3] сообщалось об успешном применении обратного распространения к сжатию изображений, когда образы представлялись одним битом на пиксель, что было восьмикратным улучшением по сравнению с входными данными.

## **ПРЕДОСТЕРЕЖЕНИЕ**

Несмотря на многочисленные успешные применения обратного распространения, оно не является панацеей. Больше всего неприятностей приносит неопределенно долгий процесс обучения. В сложных задачах для обучения сети могут потребоваться дни или даже недели, она может и вообще не обучиться. Длительное время обучения может быть результатом неоптимального выбора длины шага. Неудачи в обучении обычно возникают по двум причинам: паралича сети и попадания в локальный минимум.

### **Паралич сети**

В процессе обучения сети значения весов могут в результате коррекции стать очень большими величинами. Это может привести к тому, что все или большинство нейронов будут функционировать при очень больших значениях ОУТ, в области, где производная сжимающей функции очень мала. Так как посылаемая обратно в процессе обучения ошибка пропорциональна этой производной, то процесс обучения может практически замереть. В теоретическом отношении эта проблема плохо изучена. Обычно этого избегают уменьшением размера шага  $\eta$ , но это увеличивает время обучения. Различные эвристики использовались для предохранения от паралича или для восстановления после него, но пока что они могут рассматриваться лишь как экспериментальные.

### **Локальные минимумы**

Обратное распространение использует разновидность градиентного спуска, т. е. осуществляет спуск вниз по поверхности ошибки, непрерывно подстраивая веса в направлении к минимуму. Поверхность ошибки сложной сети сильно изрезана и состоит из холмов, долин, складок и оврагов в пространстве высокой размерности. Сеть может попасть в локальный минимум (неглубокую долину), когда рядом имеется гораздо более глубокий минимум. В точке локального минимума все направления ведут вверх, и сеть неспособна из него выбраться. Статистические методы обучения могут помочь избежать этой ловушки, но они медленны. В [10] предложен метод, объединяющий статистические методы машины Коши с градиентным спуском обратного распространения и приводящий к системе, которая находит глобальный минимум, сохраняя высокую скорость обратного распространения. Это обсуждается в гл. 5.

### **Размер шага**

Внимательный разбор доказательства сходимости в [7] показывает, что коррекции весов предполагаются бесконечно малыми. Ясно, что это неосуществимо на практике, так как ведет к бесконечному времени обучения. Размер шага должен браться конечным, и в этом вопросе приходится опираться только на опыт. Если размер шага очень мал, то сходимость слишком медленная, если же очень велик, то может возникнуть паралич или постоянная неустойчивость. В [11] описан адаптивный алгоритм выбора шага, автоматически корректирующий размер шага в процессе обучения.

### **Временная неустойчивость**

Если сеть учится распознавать буквы, то нет смысла учить «Б», если при этом забывается «А». Процесс обучения должен быть таким, чтобы сеть обучалась на всем обучающем множестве без пропусков того, что уже выучено. В доказательстве сходимости [7] это условие выполнено, но требуется также, чтобы сети предъявлялись все векторы обучающего множества прежде, чем выполняется коррекция весов. Необходимые изменения весов должны вычисляться на всем множестве, а это требует дополнительной памяти; после ряда таких обучающих циклов веса сойдутся к минимальной ошибке. Этот метод может оказаться бесполезным, если сеть находится в постоянно меняющейся внешней среде, так что второй раз один и тот же вектор может уже не повториться. В этом случае процесс обучения может никогда не сойтись, бесцельно блуждая или сильно осциллируя. В этом смысле обратное распространение не похоже на биологические системы. Как будет указано в гл. 8, это несоответствие (среди прочих) привело к системе ART, принадлежащей Гроссбергу.

## **Литература**

1. Almeida L. B. 1987. Neural computaters. Proceedings of NATO ARW on Neural Computers, Dusseldorf. Heidelberg: Springer-Verlag.

2. Burr D. J. 1987. Experiments with a connectionist text reader. In Proceedings of the IEEE First International Conference on Neural Networks, eds. M. Caudill and C. Butler, vol. 4, pp. 717-24. San Diego, CA: SOS Printing.
3. Cottrell G. W., Munro P., Zipser D. 1987. Image compression by backpropagation: An example of extensional programming. ICS Report 8702, University of California, San Diego.
4. Parker D. B. 1982. Learning logic. Invention Report S81-64, File 1, Office of Technology Licensing, Stanford University, Stanford, CA.
5. Parker D. B. 1987. Second order back propagation: Implementing an optimal  $O(n)$  approximation to Newton's method as an artificial neural network. Manuscript submitted for publication.
6. Pineda F. J. 1988. Generalization of backpropagation to recurrent and higher order networks. In Neural information processing systems, ed. Dana Z. Anderson, pp. 602-11. New York: American Institute of Physics.
7. Rumelhart D. E., Hinton G. E., Williams R. J. 1986. Learning internal representations by error propagation. In Parallel distributed processing, vol. 1, pp. 318-62. Cambridge, MA: MIT Press.
8. Sejnowski T. J., Rosenberg C. R. 1987. Parallel networks that learn to pronounce English text. Complex Systems 1:145-68.
9. Stornetta W. S., Huberman B. A. 1987. An improved three-layer, backpropagation algorithm. In Proceedings of the IEEE First International Conference on Neural Networks, eds. M. Caudill and C. Butler. San Diego, CA: SOS Printing.
10. Wasserman P. D. 1988a. Combined backpropagation/Cauchy machine. Proceedings of the International Neural Network Society. New York: Pergamon Press.
11. Wasserman P. D. 1988b. Experiments in translating Chinese characters using backpropagation. Proceedings of the Thirty-Third IEEE Computer Society International Conference. Washington, D. C.: Computer Society Press of the IEEE.
12. Werbos P. J. 1974. Beyond regression: New tools for prediction and analysis in the behavioral sciences. Masters thesis, Harvard University.

## Глава 4.

### Сети встречного распространения

#### ВВЕДЕНИЕ В СЕТИ ВСТРЕЧНОГО РАСПРОСТРАНЕНИЯ

Возможности сети встречного распространения, разработанной в [5-7], превосходят возможности однослойных сетей. Время же обучения по сравнению с обратным распространением может уменьшаться в сто раз. Встречное распространение не столь общо, как обратное распространение, но оно может давать решение в тех приложениях, где долгая обучающая процедура невозможна. Будет показано, что помимо преодоления ограничений других сетей встречное распространение обладает собственными интересными и полезными свойствами.

Во встречном распространении объединены два хорошо известных алгоритма: самоорганизующаяся карта Кохонена [8] и звезда Гроссберга [2-4] (см. приложение Б). Их объединение ведет к свойствам, которых нет ни у одного из них в отдельности.

Методы, которые подобно встречному распространению, объединяют различные сетевые парадигмы как строительные блоки, могут привести к сетям, более близким к мозгу по архитектуре, чем любые другие однородные структуры. Похоже, что в мозгу именно каскадные соединения модулей различной специализации позволяют выполнять требуемые вычисления.

Сеть встречного распространения функционирует подобно столу справок, способному к обобщению. В процессе обучения входные векторы ассоциируются с соответствующими выходными векторами. Эти векторы могут быть двоичными, состоящими из нулей и единиц, или непрерывными. Когда сеть обучена, приложение входного вектора приводит к требуемому выходному вектору. Обобщающая способность сети позволяет получать правильный выход даже при приложении входного вектора, который является неполным или слегка неверным. Это позволяет использовать данную сеть для распознавания образов, восстановления образов и усиления сигналов.

#### СТРУКТУРА СЕТИ

На рис. 4.1 показана упрощенная версия прямого действия сети встречного распространения. На нем иллюстрируются функциональные свойства этой парадигмы. Полная двунаправленная сеть основана на тех же принципах, она обсуждается в этой главе позднее.

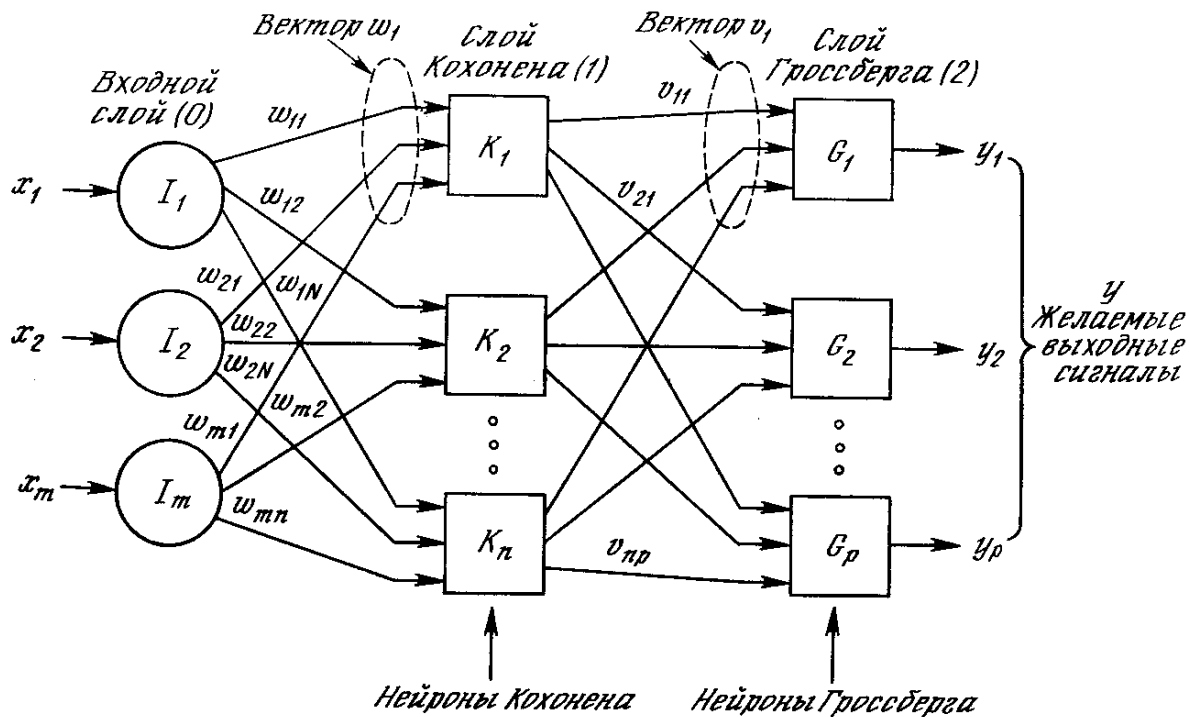


Рис. 4.1. Сеть с встречным распознаванием без обратных связей

Нейроны слоя 0 (показанные кружками) служат лишь точками разветвления и не выполняют вычислений. Каждый нейрон слоя 0 соединен с каждым нейроном слоя 1 (называемого слоем Кохонена) отдельным весом  $w_{mn}$ . Эти веса в целом рассматриваются как матрица весов  $W$ . Аналогично, каждый нейрон в слое Кохонена (слое 1) соединен с каждым нейроном в слое Гроссберга (слое 2) весом  $v_{np}$ . Эти веса образуют матрицу весов  $V$ . Все это весьма напоминает другие сети, встречавшиеся в предыдущих главах, различие, однако, состоит в операциях, выполняемых нейронами Кохонена и Гроссберга.

Как и многие другие сети, встречное распространение функционирует в двух режимах: в нормальном режиме, при котором принимается входной вектор  $X$  и выдается выходной вектор  $Y$ , и в режиме обучения, при котором подается входной вектор и веса корректируются, чтобы дать требуемый выходной вектор.

## НОРМАЛЬНОЕ ФУНКЦИОНИРОВАНИЕ

### Слой Кохонена

В своей простейшей форме слой Кохонена функционирует в духе «победитель забирает все», т. е. для данного входного вектора один и только один нейрон Кохонена выдает на выходе логическую единицу, все остальные выдают ноль. Нейроны Кохонена можно воспринимать как набор электрических лампочек, так что для любого входного вектора загорается одна из них.

Ассоциированное с каждым нейроном Кохонена множество весов соединяет его с каждым входом. Например, на рис. 4.1 нейрон Кохонена  $K_1$  имеет веса  $w_{11}, w_{21}, \dots, w_{m1}$ , составляющие весовой вектор  $W_1$ . Они соединяются через входной слой с входными сигналами  $x_1, x_2, \dots, x_m$ , составляющими входной вектор  $X$ . Подобно нейронам большинства сетей выход NET каждого нейрона Кохонена является просто суммой взвешенных входов. Это может быть выражено следующим образом:

$$NET_j = w_{1j}x_1 + w_{2j}x_2 + \dots + w_{mj}x_m \quad (4.1)$$

где  $NET_j$  – это выход NET нейрона Кохонена  $j$ ,

$$NET_j = \sum_i x_i w_{ij} \quad (4.2)$$

или в векторной записи

$$N = XW, \quad (4.3)$$

где  $N$  – вектор выходов NET слоя Кохонена.

Нейрон Кохонена с максимальным значением NET является «победителем». Его выход равен единице, у остальных он равен нулю.

### Слой Гроссберга

Слой Гроссберга функционирует в сходной манере. Его выход NET является взвешенной суммой выходов  $k_1, k_2, \dots, k_n$  слоя Кохонена, образующих вектор  $\mathbf{K}$ . Вектор соединяющих весов, обозначенный через  $\mathbf{V}$ , состоит из весов  $v_{11}, v_{21}, \dots, v_{np}$ . Тогда выход NET каждого нейрона Гроссберга есть

$$NET_j = \sum_i k_i w_{ij}, \quad (4.4)$$

где  $NET_j$  – выход  $j$ -го нейрона Гроссберга, или в векторной форме

$$\mathbf{Y} = \mathbf{KV}, \quad (4.5)$$

где  $\mathbf{Y}$  – выходной вектор слоя Гроссберга,  $\mathbf{K}$  – выходной вектор слоя Кохонена,  $\mathbf{V}$  – матрица весов слоя Гроссберга.

Если слой Кохонена функционирует таким образом, что лишь у одного нейрона величина NET равна единице, а у остальных равна нулю, то лишь один элемент вектора  $\mathbf{K}$  отличен от нуля, и вычисления очень просты. Фактически каждый нейрон слоя Гроссберга лишь выдает величину веса, который связывает этот нейрон с единственным ненулевым нейроном Кохонена.

## ОБУЧЕНИЕ СЛОЯ КОХОНЕНА

Слой Кохонена классифицирует входные векторы в группы схожих. Это достигается с помощью такой подстройки весов слоя Кохонена, что близкие входные векторы активируют один и тот же нейрон данного слоя. Затем задачей слоя Гроссберга является получение требуемых выходов.

Обучение Кохонена является самообучением, протекающим без учителя. Поэтому трудно (и не нужно) предсказывать, какой именно нейрон Кохонена будет активироваться для заданного входного вектора. Необходимо лишь гарантировать, чтобы в результате обучения разделялись несхожие входные векторы.

### Предварительная обработка входных векторов

Весьма желательно (хотя и не обязательно) нормализовать входные векторы перед тем, как предъявлять их сети. Это выполняется с помощью деления каждой компоненты входного вектора на длину вектора. Эта длина находится извлечением квадратного корня из суммы квадратов компонент вектора. В алгебраической записи

$$x'_i = \frac{x_i}{\sqrt{x_1^2 + x_2^2 + \dots + x_n^2}} \quad (4.6)$$

Это превращает входной вектор в единичный вектор с тем же самым направлением, т. е. в вектор единичной длины в  $n$ -мерном пространстве.

Уравнение (4.6) обобщает хорошо известный случай двух измерений, когда длина вектора равна гипотенузе прямоугольного треугольника, образованного его  $x$  и  $y$  компонентами, как это следует из известной теоремы Пифагора. На рис. 4.2а такой двумерный вектор  $\mathbf{V}$  представлен в координатах  $x$ - $y$ , причем координата  $x$  равна четырем, а координата  $y$  – трем. Квадратный корень из суммы квадратов этих компонент равен пяти. Деление каждой компоненты  $\mathbf{V}$  на пять дает вектор  $\mathbf{V}'$  с компонентами  $4/5$  и  $3/5$ , где  $\mathbf{V}'$  указывает в том же направлении, что и  $\mathbf{V}$ , но имеет единичную длину.

На рис. 4.2б показано несколько единичных векторов. Они оканчиваются в точках единичной окружности (окружности единичного радиуса), что имеет место, когда у сети лишь два входа. В случае трех входов векторы представлялись бы стрелками, оканчивающимися на поверхности единичной сферы. Эти представления могут быть перенесены на сети, имеющие произвольное число входов, где каждый входной вектор является стрелкой, оканчивающейся на поверхности единичной гиперсферы (полезной абстракцией, хотя и не допускающей непосредственной визуализации).

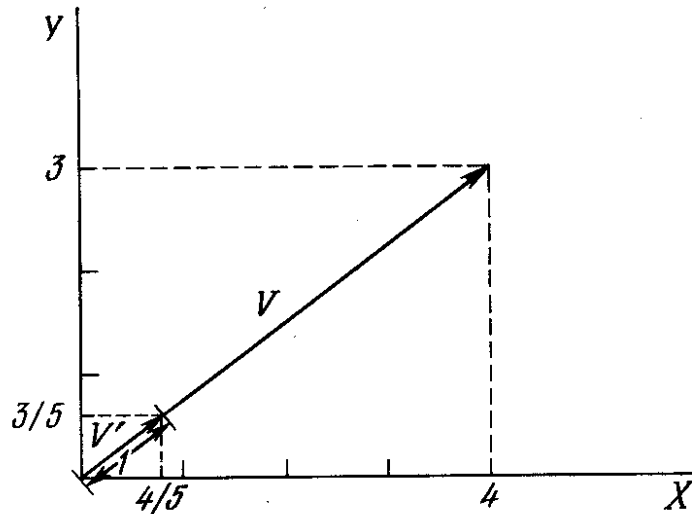


Рис. 4.2а. Единичный входной вектор

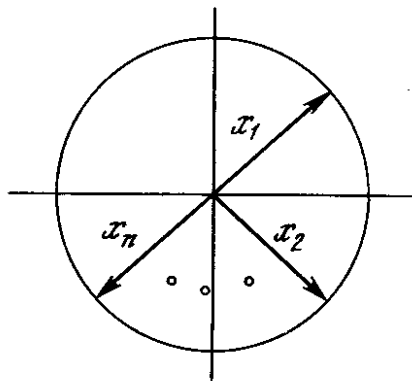


Рис. 4.2б. Двумерные единичные векторы на единичной окружности

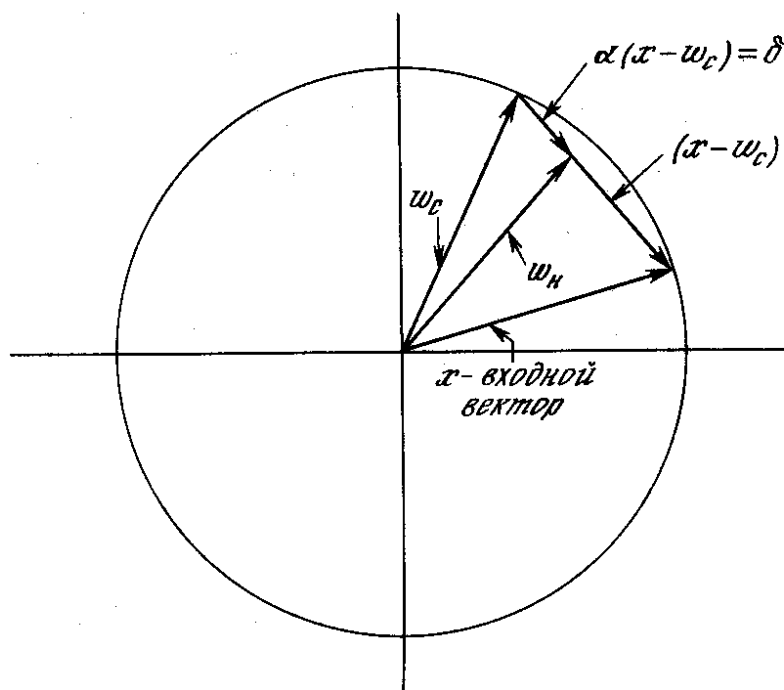
При обучении слоя Кохонена на вход подается входной вектор и вычисляются его скалярные произведения с векторами весов, связанными со всеми нейронами Кохонена. Нейрон с максимальным значением скалярного произведения объявляется «победителем» и его веса подстраиваются. Так как скалярное произведение, используемое для вычисления величин NET, является мерой сходства между входным вектором и вектором весов, то процесс обучения состоит в выборе нейрона Кохонена с весовым вектором, наиболее близким к входному вектору, и дальнейшем приближении весового вектора к входному. Снова отметим, что процесс является самообучением, выполняемым без учителя. Сеть самоорганизуется таким образом, что данный нейрон Кохонена имеет максимальный выход для данного входного вектора. Уравнение, описывающее процесс обучения имеет следующий вид:

$$w_n = w_c + \alpha(x - w_c), \quad (4.7)$$

где  $w_n$  – новое значение веса, соединяющего входную компоненту  $x$  с выигравшим нейроном;  $w_c$  – предыдущее значение этого веса;  $\alpha$  – коэффициент скорости обучения, который может варьироваться в процессе обучения.

Каждый вес, связанный с выигравшим нейроном Кохонена, изменяется пропорционально разности между его величиной и величиной входа, к которому он присоединен. Направление изменения минимизирует разность между весом и его входом.

На рис. 4.3 этот процесс показан геометрически в двумерном виде. Сначала находится вектор  $X - W_c$ , для этого проводится отрезок из конца  $W$  в конец  $X$ . Затем этот вектор укорачивается умножением его на скалярную величину  $\alpha$ , меньшую единицы, в результате чего получается вектор изменения  $\delta$ . Окончательно новый весовой вектор  $W_n$  является отрезком, направленным из начала координат в конец вектора  $\delta$ . Отсюда можно видеть, что эффект обучения состоит во вращении весового вектора в направлении входного вектора без существенного изменения его длины.



**Рис. 4.3. Вращение весового вектора в процессе обучения ( $W_n$  – вектор новых весовых коэффициентов,  $W_c$  – вектор старых весовых коэффициентов)**

Переменная  $\alpha$  является коэффициентом скорости обучения, который вначале обычно равен  $\sim 0,7$  и может постепенно уменьшаться в процессе обучения. Это позволяет делать большие начальные шаги для быстрого грубого обучения и меньшие шаги при подходе к окончательной величине.

Если бы с каждым нейроном Кохонена ассоциировался один входной вектор, то слой Кохонена мог бы быть обучен с помощью одного вычисления на вес. Веса нейрона-победителя приравнивались бы к компонентам обучающего вектора ( $\alpha = 1$ ). Как правило, обучающее множество включает много сходных между собой входных векторов, и сеть должна быть обучена активировать один и тот же нейрон Кохонена для каждого из них. В этом случае веса этого нейрона должны получаться усреднением входных векторов, которые должны его активировать. Постепенное уменьшение величины  $\alpha$  уменьшает воздействие каждого обучающего шага, так что окончательное значение будет средней величиной от входных векторов, на которых происходит обучение. Таким образом, веса, ассоциированные с нейроном, примут значение вблизи «центра» входных векторов, для которых данный нейрон является «победителем».

#### **Выбор начальных значений весовых векторов**

Всем весам сети перед началом обучения следует придать начальные значения. Общепринятой практикой при работе с нейронными сетями является присваивание весам небольших случайных значений. При обучении слоя Кохонена случайно выбранные весовые векторы следует нормализовать. Окончательные значения весовых векторов после обучения совпадают с нормализованными входными векторами. Поэтому нормализация перед началом обучения приближает весовые векторы к их окончательным значениям, сокращая, таким образом, обучающий процесс.

Рандомизация весов слоя Кохонена может породить серьезные проблемы при обучении, так как в результате ее весовые векторы распределяются равномерно по поверхности гиперсферы. Из-за того, что входные векторы, как правило, распределены неравномерно и имеют тенденцию группироваться на относительно малой части поверхности гиперсферы, большинство весовых векторов будут так удалены от любого входного вектора, что они никогда не будут давать наилучшего соответствия. Эти нейроны Кохонена будут всегда иметь нулевой выход и окажутся бесполезными. Более того, оставшихся весов, дающих наилучшие соответствия, может оказаться слишком мало, чтобы разделить входные векторы на классы, которые расположены близко друг к другу на поверхности гиперсферы.

Допустим, что имеется несколько множеств входных векторов, все множества сходные, но должны быть разделены на различные классы. Сеть должна быть обучена активировать отдельный нейрон Кохонена для каждого класса. Если начальная плотность весовых векторов в окрестности обучающих векторов слишком мала, то может оказаться невозможным разделить сходные классы из-за того, что не будет достаточного количества весовых векторов в интересующей нас окрестности, чтобы приписать по одному из них каждому классу входных векторов.



Наоборот, если несколько входных векторов получены незначительными изменениями из одного и того же образца и должны быть объединены в один класс, то они должны включать один и тот же нейрон Кохонена. Если же плотность весовых векторов очень высока вблизи группы слегка различных входных векторов, то каждый входной вектор может активировать отдельный нейрон Кохонена. Это не является катастрофой, так как слой Гроссберга может отобразить различные нейроны Кохонена в один и тот же выход, но это расточительная трата нейронов Кохонена.

Наиболее желательное решение состоит в том, чтобы распределять весовые векторы в соответствии с плотностью входных векторов, которые должны быть разделены, помещая тем самым больше весовых векторов в окрестности большого числа входных векторов. На практике это невыполнимо, однако существует несколько методов приближенного достижения тех же целей.

Одно из решений, известное под названием *метода выпуклой комбинации* (convex combination method), состоит в том, что все веса приравниваются одной и той же величине

$$w_i = \frac{1}{\sqrt{n}},$$

где  $n$  – число входов  $i$ , следовательно, число компонент каждого весового вектора. Благодаря этому все весовые векторы совпадают и имеют единичную длину. Каждой же компоненте входа  $X$  придается значение

$$x_i = \alpha x_i + \frac{1 - \alpha}{\sqrt{n}},$$

где  $n$  – число входов. В начале  $\alpha$  очень мало, вследствие чего все входные векторы имеют длину, близкую к  $\frac{1}{\sqrt{n}}$ , и почти

совпадают с векторами весов. В процессе обучения сети  $\alpha$  постепенно возрастает, приближаясь к единице. Это позволяет разделять входные векторы и окончательно приписывает им их истинные значения. Весовые векторы отслеживают один или небольшую группу входных векторов и в конце обучения дают требуемую картину выходов. Метод выпуклой комбинации хорошо работает, но замедляет процесс обучения, так как весовые векторы подстраиваются к изменяющейся цели. Другой подход состоит в добавлении шума к входным векторам. Тем самым они подвергаются случайным изменениям, схватывая в конце концов весовой вектор. Этот метод также работоспособен, но еще более медленен, чем метод выпуклой комбинации.

Третий метод начинает со случайных весов, но на начальной стадии обучающего процесса подстраивает все веса, а не только связанные с выигравшим нейроном Кохонена. Тем самым весовые векторы перемещаются ближе к области входных векторов. В процессе обучения коррекция весов начинается производиться лишь для ближайших к победителю нейронов Кохонена. Этот радиус коррекции постепенно уменьшается, так что в конце концов корректируются только веса, связанные с выигравшим нейроном Кохонена.

Еще один метод наделяет каждый нейрон Кохонена «Чувством справедливости». Если он становится победителем чаще своей законной доли времени (примерно  $1/k$ , где  $k$  – число нейронов Кохонена), он временно увеличивает свой порог, что уменьшает его шансы на выигрыш, давая тем самым возможность обучаться и другим нейронам.

Во многих приложениях точность результата существенно зависит от распределения весов. К сожалению, эффективность различных решений исчерпывающим образом не оценена и остается проблемой.

### Режим интерполяции

До сих пор мы обсуждали алгоритм обучения, в котором для каждого входного вектора активировался лишь один нейрон Кохонена. Это называется *методом аккредитации*. Его точность ограничена, так как выход полностью является функцией лишь одного нейрона Кохонена.

В *методе интерполяции* целая группа нейронов Кохонена, имеющих наибольшие выходы, может передавать свои выходные сигналы в слой Гроссберга. Число нейронов в такой группе должно выбираться в зависимости от задачи, и убедительных данных относительно оптимального размера группы не имеется. Как только группа определена, ее множество выходов NET рассматривается как вектор, длина которого нормализуется на единицу делением каждого значения NET на корень квадратный из суммы квадратов значений NET в группе. Все нейроны вне группы имеют нулевые выходы.

Метод интерполяции способен устанавливать более сложные соответствия и может давать более точные результаты. По-прежнему, однако, нет убедительных данных, позволяющих сравнить режимы интерполяции и аккредитации.

### Статистические свойства обученной сети

Метод обучения Кохонена обладает полезной и интересной способностью извлекать статистические свойства из множества входных данных. Как показано Кохоненом [8], для полностью обученной сети вероятность того, что случайно выбранный входной вектор (в соответствии с функцией плотности вероятности входного множества) будет ближайшим к любому заданному весовому вектору, равна  $1/k$ , где  $k$  – число нейронов Кохонена. Это является оптимальным распределением весов на гиперсфере. (Предполагается, что используются все весовые векторы, что имеет место лишь в том случае, если используется один из обсуждавшихся методов распределения весов.)

### ОБУЧЕНИЕ СЛОЯ ГРОССБЕРГА

Слой Гроссберга обучается относительно просто. Входной вектор, являющийся выходом слоя Кохонена, подается на слой нейронов Гроссберга, и выходы слоя Гроссберга вычисляются, как при нормальном функционировании. Далее, каждый вес корректируется лишь в том случае, если он соединен с нейроном Кохонена, имеющим ненулевой выход. Величина коррекции веса пропорциональна разности между весом и требуемым выходом нейрона Гроссберга, с которым он соединен. В символической записи

$$v_{ijn} = v_{ije} + \beta(y_j - v_{ije})k_i, \quad (4.8)$$

где  $k_i$  – выход  $i$ -го нейрона Кохонена (только для одного нейрона Кохонена он отличен от нуля);  $y_j$  –  $j$ -ая компонента вектора желаемых выходов.

Первоначально  $\beta$  берется равным  $\sim 0,1$  и затем постепенно уменьшается в процессе обучения.

Отсюда видно, что веса слоя Гроссберга будут сходиться к средним величинам от желаемых выходов, тогда как веса слоя Кохонена обучаются на средних значениях входов. Обучение слоя Гроссберга – это обучение с учителем, алгоритм располагает желаемым выходом, по которому он обучается. Обучающийся без учителя, самоорганизующийся слой Кохонена дает выходы в недетерминированных позициях. Они отображаются в желаемые выходы слоем Гроссберга.

### СЕТЬ ВСТРЕЧНОГО РАСПРОСТРАНЕНИЯ ПОЛНОСТЬЮ

На рис. 4.4 показана сеть встречного распространения целиком. В режиме нормального функционирования предъявляются входные векторы  $\mathbf{X}$  и  $\mathbf{Y}$ , и обученная сеть дает на выходе векторы  $\mathbf{X}'$  и  $\mathbf{Y}'$ , являющиеся аппроксимациями соответственно для  $\mathbf{X}$  и  $\mathbf{Y}$ . Векторы  $\mathbf{X}$  и  $\mathbf{Y}$  предполагаются здесь нормализованными единичными векторами, следовательно, порождаемые на выходе векторы также будут иметь тенденцию быть нормализованными.

В процессе обучения векторы  $\mathbf{X}$  и  $\mathbf{Y}$  подаются одновременно и как входные векторы сети, и как желаемые выходные сигналы. Вектор  $\mathbf{X}$  используется для обучения выходов  $\mathbf{X}'$ , а вектор  $\mathbf{Y}$  – для обучения выходов  $\mathbf{Y}'$  слоя Гроссберга. Сеть встречного распространения целиком обучается с использованием того же самого метода, который описывался для сети прямого действия. Нейроны Кохонена принимают входные сигналы как от векторов  $\mathbf{X}$ , так и от векторов  $\mathbf{Y}$ . Но это неотличимо от ситуации, когда имеется один большой вектор, составленный из векторов  $\mathbf{X}$  и  $\mathbf{Y}$ , и не влияет на алгоритм обучения.

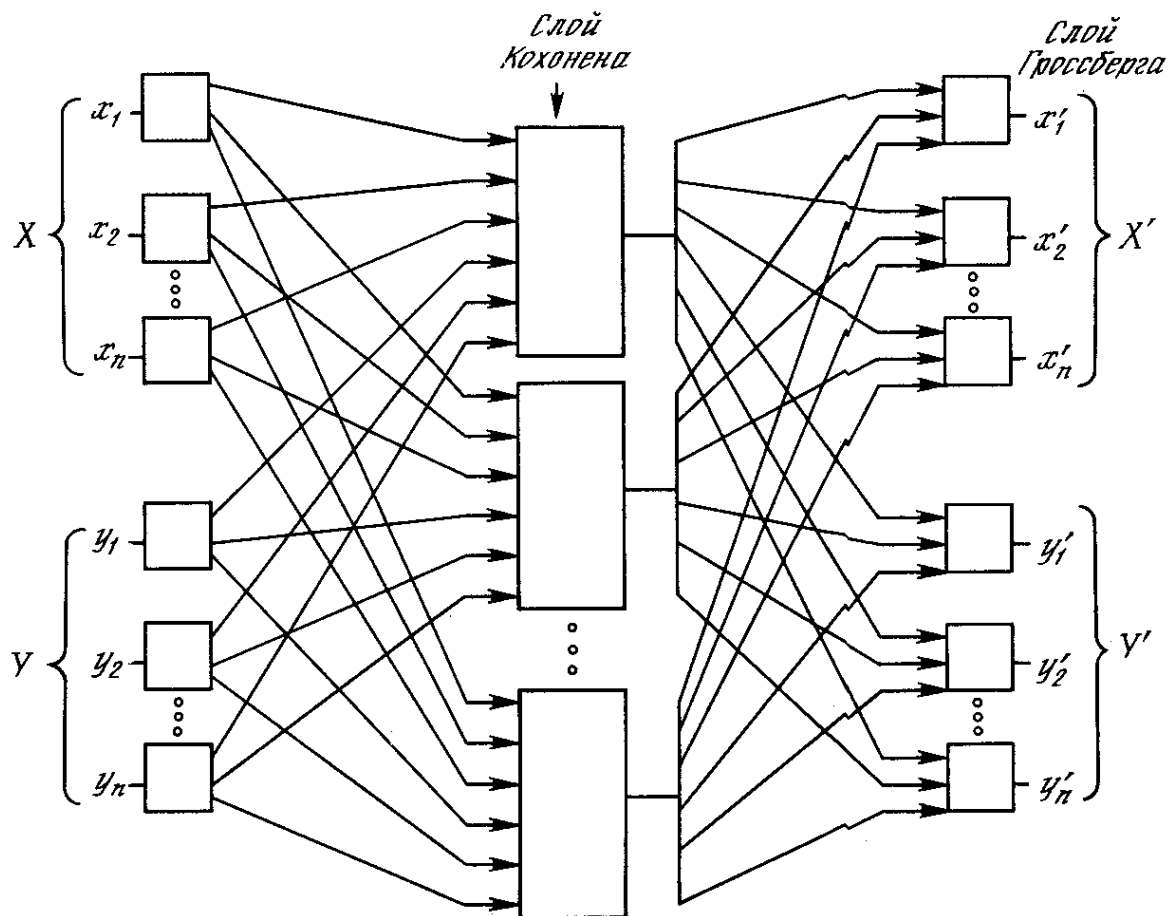


Рис. 4.4. Полная сеть встречного распространения

В качестве результирующего получается единичное отображение, при котором предъявление пары входных векторов порождает их копии на выходе. Это не представляется особенно интересным, если не заметить, что предъявление только вектора  $X$  (с вектором  $Y$ , равным нулю) порождает как выходы  $X'$ , так и выходы  $Y'$ . Если  $F$  – функция, отображающая  $X$  в  $Y'$ , то сеть аппроксимирует ее. Также, если  $F$  обратима, то предъявление только вектора  $Y$  (приравнивая  $X$  нулю) порождает  $X'$ . Уникальная способность порождать функцию и обратную к ней делает сеть встречного распространения полезной в ряде приложений.

Рис. 4.4 в отличие от первоначальной конфигурации [5] не демонстрирует противоток в сети, по которому она получила свое название. Такая форма выбрана потому, что она также иллюстрирует сеть без обратных связей и позволяет обобщить понятия, развитые в предыдущих главах.

#### ПРИЛОЖЕНИЕ: СЖАТИЕ ДАННЫХ

В дополнение к обычным функциям отображения векторов встречное распространение оказывается полезным и в некоторых менее очевидных прикладных областях. Одним из наиболее интересных примеров является сжатие данных.

Сеть встречного распространения может быть использована для сжатия данных перед их передачей, уменьшая тем самым число битов, которые должны быть переданы. Допустим, что требуется передать некоторое изображение. Оно может быть разбито на подизображения  $S$ , как показано на рис. 4.5. Каждое подизображение разбито на пиксели (мельчайшие элементы изображения). Тогда каждое подизображение является вектором, элементами которого являются пиксели, из которых состоит подизображение. Допустим для простоты, что каждый пиксель – это единица (свет) или нуль (чернота). Если в подизображении имеется  $n$  пикселей, то для его передачи потребуется  $n$  бит. Если допустимы некоторые искажения, то для передачи типичного изображения требуется существенно меньшее число битов, что позволяет передавать изображения быстрее. Это возможно из-за статистического распределения векторов подизображений. Некоторые из них встречаются часто, тогда как другие встречаются так редко, что могут быть грубо аппроксимированы. Метод, называемый *векторным квантованием*, находит более короткие последовательности битов, наилучшим образом представляющие эти подизображения.

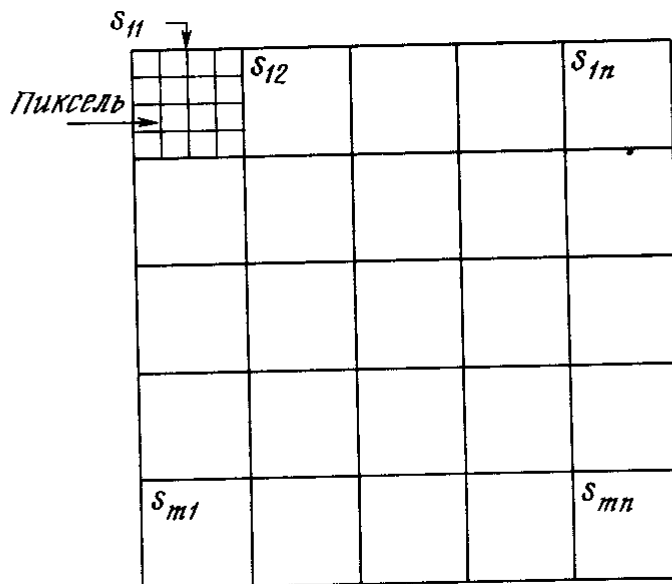


Рис. 4.5. Система сжатия изображений.

Сеть встречного распространения может быть использована для выполнения векторного квантования. Множество векторов подизображений используется в качестве входа для обучения слоя Кохонена по методу аккредитации, когда лишь выход одного нейрона равен 1. Веса слоя Гроссберга обучаются выдавать бинарный код номера того нейрона Кохонена, выход которого равен 1. Например, если выходной сигнал нейрона 7 равен 1 (а все остальные равны 0), то слой Гроссберга будет обучаться выдавать 00...000111 (двоичный код числа 7). Это и будет являться более короткой битовой последовательностью передаваемых символов.

На приемном конце идентичным образом обученная сеть встречного распространения принимает двоичный код и реализует обратную функцию, аппроксимирующую первоначальное подизображение.

Этот метод применялся как к речи, так и к изображениям, с коэффициентом сжатия данных от 10:1 до 100:1. Качество было приемлемым, хотя некоторые искажения данных на приемном конце неизбежны.

## ОБСУЖДЕНИЕ

Роберт Хехт-Нильсон, создатель сети встречного распространения (СВР), осознавал ее ограничения: «СВР, конечно, уступает обратному распространению в большинстве приложений, связанных с сетевыми отображениями. Ее преимущества в том, что она проста и дает хорошую статистическую модель для своей среды входных векторов» ([5], с. 27).

К этому можно добавить, что сеть встречного распространения быстро обучается, и при правильном использовании она может сэкономить значительное количество машинного времени. Она полезна также для быстрого моделирования систем, где большая точность обратного распространения вынуждает отдать ему предпочтение в окончательном варианте, но важна быстрая начальная аппроксимация. Возможность породить функцию и обратную к ней также нашло применение в ряде систем.

## Литература

13. DeSieno D. 1988. Adding a conscience to competitive learning Proceedings of the IEEE International Conference on Neural Networks, pp. 117-24. San Diego, CA: SOS Printing.
14. Grossberg S. 1969. Some networks that can learn, remember and reproduce any number of complicated space-time patterns. Journal of Mathematics and Mechanics, 19:53-91.
15. Grossberg S. 1971. Embedding fields: Underlying philosophy, mathematics, and applications of psychology, physiology, and anatomy. Journal of Cybernetics, 1:28-50.
16. Grossberg S. 1982. Studies of mind and brain. Boston: Reidel.
17. Hecht-Nielsen R. 1987a. Counterpropagation networks. In Proceedings of the IEEE First International Conference on Neural Networks, eds. M. Caudill and C. Butler, vol. 2, pp. 19-32. San Diego, CA: SOS Printing.
18. Hecht-Nielsen R. 1987b. Counterpropagation networks. Applied Optics 26(23): 4979-84.

19. Hecht-Nielsen R. 1988. Applications of Counterpropagation networks. Neural Networks 1: 131-39.
20. Kohonen T. 1988. Self-organization and associative memory. 2d ed. New-York, Springer-Verlag.

## Глава 5.

### Стохастические методы

Стохастические методы полезны как для обучения искусственных нейронных сетей, так и для получения выхода от уже обученной сети. Стохастические методы обучения приносят большую пользу, позволяя исключать локальные минимумы в процессе обучения. Но с ними также связан ряд проблем.

Использование стохастических методов для получения выхода от уже обученной сети рассматривалось в работе [2] и обсуждается нами в гл. 6. Данная глава посвящена методам обучения сети.

### ИСПОЛЬЗОВАНИЕ ОБУЧЕНИЯ

Искусственная нейронная сеть обучается посредством некоторого процесса, модифицирующего ее веса. Если обучение успешно, то предъявление сети множества входных сигналов приводит к появлению желаемого множества выходных сигналов. Имеется два класса обучающих методов: детерминистский и стохастический.

*Детерминистский метод обучения* шаг за шагом осуществляет процедуру коррекции весов сети, основанную на использовании их текущих значений, а также величин входов, фактических выходов и желаемых выходов. Обучение персептрона является примером подобного детерминистского подхода (см. гл. 2).

*Стохастические методы обучения* выполняют псевдослучайные изменения величин весов, сохраняя те изменения, которые ведут к улучшениям. Чтобы увидеть, как это может быть сделано, рассмотрим рис. 5.1, на котором изображена типичная сеть, в которой нейроны соединены с помощью весов. Выход нейрона является здесь взвешенной суммой его входов, которая преобразована с помощью нелинейной функции (подробности см. гл. 2). Для обучения сети может быть использована следующая процедура:

1. Выбрать вес случайным образом и подкорректировать его на небольшое случайное. Предъявить множество входов и вычислить получающиеся выходы.
2. Сравнить эти выходы с желаемыми выходами и вычислить величину разности между ними. Общепринятый метод состоит в нахождении разности между фактическим и желаемым выходами для каждого элемента обучаемой пары, возведение разностей в квадрат и нахождение суммы этих квадратов. Целью обучения является минимизация этой разности, часто называемой *целевой функцией*.
3. Выбрать вес случайным образом и подкорректировать его на небольшое случайное значение. Если коррекция помогает (уменьшает целевую функцию), то сохранить ее, в противном случае вернуться к первоначальному значению веса.
4. Повторять шаги с 1 до 3 до тех пор, пока сеть не будет обучена в достаточной степени.

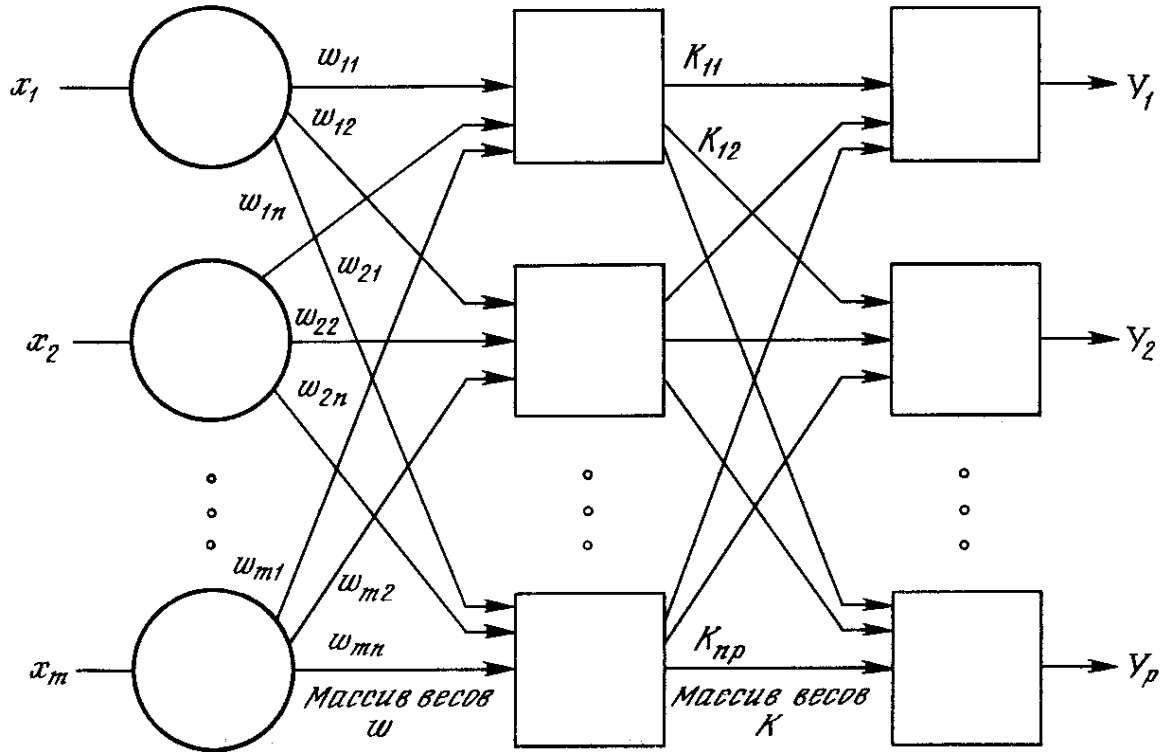


Рис. 5.1. Двухслойная сеть без обратных связей

Этот процесс стремится минимизировать целевую функцию, но может попасть, как в ловушку, в неудачное решение. На рис. 5.2 показано, как это может иметь место в системе с единственным весом. Допустим, что первоначально вес взят равным значению в точке А. Если случайные шаги по весу малы, то любые отклонения от точки А увеличивают целевую функцию и будут отвергнуты. Лучшее значение веса, принимаемое в точке В, никогда не будет найдено, и система будет поймана в ловушку локальным минимумом, вместо глобального минимума в точке В. Если же случайные коррекции веса очень велики, то как точка А, так и точка В будут часто посещаться, но то же самое будет иметь место и для каждой другой точки. Вес будет меняться так резко, что он никогда не установится в желаемом минимуме.

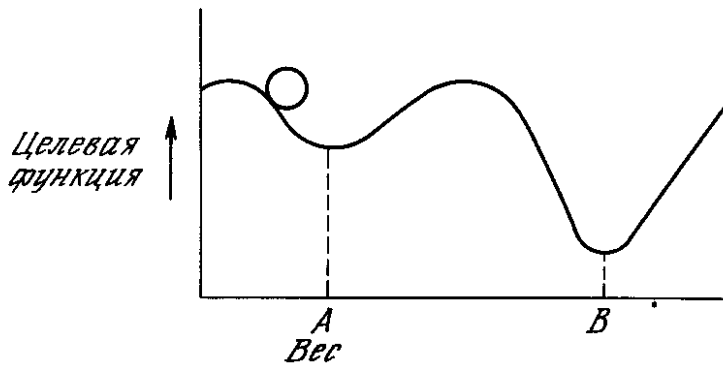


Рис.5.2. Проблема локальных минимумов.

Полезная стратегия для избежания подобных проблем состоит в больших начальных шагах и постепенном уменьшении размера среднего случайного шага. Это позволяет сети вырваться из локальных минимумов и в то же время гарантирует окончательную стабилизацию сети.

Ловушки локальных минимумов досаждают всем алгоритмам обучения, основанным на поиске минимума, включая перцептрон и сети обратного распространения, и представляют серьезную и широко распространенную трудность, которой часто не замечают. Стохастические методы позволяют решить эту проблему. Стратегия коррекции весов, вынуждающая веса принимать значение глобального оптимума в точке В, возможна.

В качестве объясняющей аналогии предположим, что на рис. 5.2 изображен шарик на поверхности в коробке. Если коробку сильно потрясти в горизонтальном направлении, то шарик будет быстро перекаатываться от одного края к другому. Нигде не задерживаясь, в каждый момент шарик будет с равной вероятностью находиться в любой точке поверхности.

Если постепенно уменьшать силу встряхивания, то будет достигнуто условие, при котором шарик будет на короткое время «застревать» в точке В. При еще более слабом встряхивании шарик будет на короткое время останавливаться как в точке А, так и в точке В. При непрерывном уменьшении силы встряхивания будет достигнута критическая точка, когда сила встряхивания достаточна для перемещения шарика из точки А в точку В, но недостаточна для того, чтобы шарик мог вскарабкаться из В в А. Таким образом, окончательно шарик остановится в точке глобального минимума, когда амплитуда встряхивания уменьшится до нуля.

Искусственные нейронные сети могут обучаться по существу тем же самым образом посредством случайной коррекции весов. Вначале делаются большие случайные коррекции с сохранением только тех изменений весов, которые уменьшают целевую функцию. Затем средний размер шага постепенно уменьшается, и глобальный минимум в конце концов достигается.

Это сильно напоминает отжиг металла, поэтому для ее описания часто используют термин «имитация отжига». В металле, нагретом до температуры, превышающей его точку плавления, атомы находятся в сильном беспорядочном движении. Как и во всех физических системах, атомы стремятся к состоянию минимума энергии (единому кристаллу в данном случае), но при высоких температурах энергия атомных движений препятствует этому. В процессе постепенного охлаждения металла возникают все более низкоэнергетические состояния, пока в конце концов не будет достигнуто наинизшее из возможных состояний, глобальный минимум. В процессе отжига распределение энергетических уровней описывается следующим соотношением:

$$P(e) = \exp(-e/kT) \quad (5.1)$$

где  $P(e)$  – вероятность того, что система находится в состоянии с энергией  $e$ ;  $k$  – постоянная Больцмана;  $T$  – температура по шкале Кельвина.

При высоких температурах  $P(e)$  приближается к единице для всех энергетических состояний. Таким образом, высокоэнергетическое состояние почти столь же вероятно, как и низкоэнергетическое. По мере уменьшения температуры вероятность высокоэнергетических состояний уменьшается по сравнению с низкоэнергетическими. При приближении температуры к нулю становится весьма маловероятным, чтобы система находилась в высокоэнергетическом состоянии.

### Больцмановское обучение

Этот стохастический метод непосредственно применим к обучению искусственных нейронных сетей:

1. Определить переменную  $T$ , представляющую искусственную температуру. Придать  $T$  большое начальное значение.
2. Предъявить сети множество входов и вычислить выходы и целевую функцию.
3. Дать случайное изменение весу и пересчитать выход сети и изменение целевой функции в соответствии со сделанным изменением веса.
4. Если целевая функция уменьшилась (улучшилась), то сохранить изменение веса.

Если изменение веса приводит к увеличению целевой функции, то вероятность сохранения этого изменения вычисляется с помощью распределения Больцмана:

$$P(c) = \exp(-c/kT) \quad (5.2)$$

где  $P(c)$  – вероятность изменения  $c$  в целевой функции;  $k$  – константа, аналогичная константе Больцмана, выбираемая в зависимости от задачи;  $T$  – искусственная температура.

Выбирается случайное число  $r$  из равномерного распределения от нуля до единицы. Если  $P(c)$  больше, чем  $r$ , то изменение сохраняется, в противном случае величина веса возвращается к предыдущему значению.

Это позволяет системе делать случайный шаг в направлении, портящем целевую функцию, позволяя ей тем самым вырваться из локальных минимумов, где любой малый шаг увеличивает целевую функцию.

Для завершения больцмановского обучения повторяют шаги 3 и 4 для *каждого* из весов сети, постепенно уменьшая температуру  $T$ , пока не будет достигнуто допустимо низкое значение целевой функции. В этот момент предъявляется другой входной вектор и процесс обучения повторяется. Сеть обучается на всех векторах обучающего множества, с возможным повторением, пока целевая функция не станет допустимой для всех них.

Величина случайного изменения веса на шаге 3 может определяться различными способами. Например, подобно тепловой системе весовое изменение  $w$  может выбираться в соответствии с гауссовским распределением:

$$P(w) = \exp(-w^2/T^2) \quad (5.2)$$

где  $P(w)$  – вероятность изменения веса на величину  $w$ ,  $T$  – искусственная температура.

Такой выбор изменения веса приводит к системе, аналогичной [3].

Так как нужна величина изменения веса  $\Delta w$ , а не вероятность изменения веса, имеющего величину  $w$ , то метод Монте-Карло может быть использован следующим образом:

1. Найти кумулятивную вероятность, соответствующую  $P(w)$ . Это есть интеграл от  $P(w)$  в пределах от 0 до  $w$ . Так как в данном случае  $P(w)$  не может быть проинтегрирована аналитически, она должна интегрироваться численно, а результат необходимо затабулировать.
2. Выбрать случайное число из равномерного распределения на интервале (0,1). Используя эту величину в качестве значения  $P(w)$ , найти в таблице соответствующее значение для величины изменения веса.

Свойства машины Больцмана широко изучались. В работе [1] показано, что скорость уменьшения температуры должна быть обратно пропорциональна логарифму времени, чтобы была достигнута сходимость к глобальному минимуму. Скорость охлаждения в такой системе выражается следующим образом:

$$T(t) = \frac{T_0}{\log(1+t)} \quad (5.4)$$

где  $T(t)$  – искусственная температура как функция времени;  $T_0$  – начальная искусственная температура;  $t$  – искусственное время.

Этот разочарывающий результат предсказывает очень медленную скорость охлаждения (и данные вычисления). Этот вывод подтвердился экспериментально. Машины Больцмана часто требуют для обучения очень большого ресурса времени.

### Обучение Коши

В работе [6] развит метод быстрого обучения подобных систем. В этом методе при вычислении величины шага распределение Больцмана заменяется на распределение Коши. Распределение Коши имеет, как показано на рис. 5.3, более длинные «хвосты», увеличивая тем самым вероятность больших шагов. В действительности распределение Коши имеет бесконечную (неопределенную) дисперсию. С помощью такого простого изменения максимальная скорость уменьшения температуры становится обратно пропорциональной линейной величине, а не логарифму, как для алгоритма обучения Больцмана. Это резко уменьшает время обучения. Эта связь может быть выражена следующим образом:

$$T(t) = \frac{T_0}{1+t} \quad (5.5)$$

Распределение Коши имеет вид

$$P(x) = \frac{T(t)}{T(t)^2 + x^2} \quad (5.6)$$

где  $P(x)$  есть вероятность шага величины  $x$ .

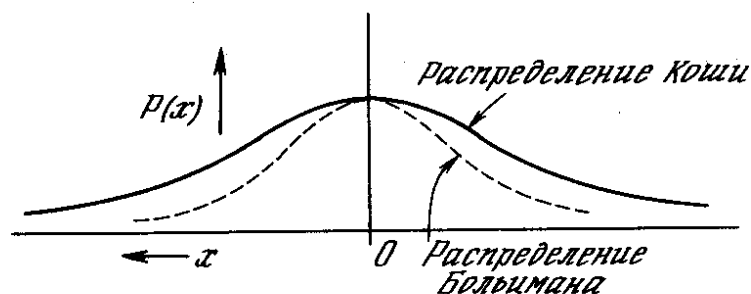


Рис. 5.3. Распределение Коши и распределение Больцмана

В уравнении (5.6)  $P(x)$  может быть проинтегрирована стандартными методами. Решая относительно  $x$ , получаем



$$x_c = \rho T(t) \operatorname{tg}(P(x)), \quad (5.7)$$

где  $\rho$  – коэффициент скорости обучения;  $x_c$  – изменение веса.

Теперь применение метода Монте Карло становится очень простым. Для нахождения  $x$  в этом случае выбирается случайное число из равномерного распределения на открытом интервале  $(-\pi/2, \pi/2)$  (необходимо ограничить функцию тангенса). Оно подставляется в формулу (5.7) в качестве  $P(x)$ , и с помощью текущей температуры вычисляется величина шага.

### Метод искусственной теплоемкости

Несмотря на улучшение, достигаемое с помощью метода Коши, время обучения может оказаться все еще слишком большим. Способ, уходящий своими корнями в термодинамику, может быть использован для ускорения этого процесса. В этом методе скорость уменьшения температуры изменяется в соответствии с искусственной «теплоемкостью», вычисляемой в процессе обучения.

Во время отжига металла происходят фазовые переходы, связанные с дискретными изменениями уровней энергии. При каждом фазовом переходе может иметь место резкое изменение величины, называемой теплоемкостью. *Теплоемкость* определяется как скорость изменения температуры с энергией. Изменения теплоемкости происходят из-за попадания системы в локальные энергетические минимумы.

Искусственные нейронные сети проходят аналогичные фазы в процессе обучения. На границе фазового перехода искусственная теплоемкость может скачкообразно измениться. Эта псевдотеплоемкость определяется как средняя скорость изменения температуры с целевой функцией. В примере шарика в коробке сильная начальная встряска делает среднюю величину целевой функции фактически не зависящей от малых изменений температуры, т. е. теплоемкость близка к константе. Аналогично при очень низких температурах система замерзает в точке минимума, так что теплоемкость снова близка к константе. Ясно, что в каждой из этих областей допустимы сильные изменения температуры, так как не происходит улучшения целевой функции.

При критических температурах небольшое уменьшение температуры приводит к большому изменению средней величины целевой функции. Возвращаясь к аналогии с шариком, при «температуре», когда шарик обладает достаточной средней энергией, чтобы перейти из А в В, но недостаточной для перехода из В в А, средняя величина целевой функции испытывает скачкообразное изменение. В этих критических точках алгоритм должен изменять температуру очень медленно, чтобы гарантировать, что система не замерзнет случайно в точке А, оказавшись пойманной в локальный минимум. Критическая температура может быть обнаружена по резкому уменьшению искусственной теплоемкости, т. е. средней скорости изменения температуры с целевой функцией. При достижении критической температуры скорость изменения температуры должна замедляться, чтобы гарантировать сходимости к глобальному минимуму. При всех остальных температурах может без риска использоваться более высокая скорость снижения температуры, что приводит к значительному снижению времени обучения.

## ПРИЛОЖЕНИЯ К ОБЩИМ НЕЛИНЕЙНЫМ ЗАДАЧАМ ОПТИМИЗАЦИИ

До сих пор в обсуждении предполагалось, что мы корректируем веса в традиционных искусственных нейронных сетях. Фактически, однако, это есть лишь некоторый частный случай. Эти статистические методы носят значительно более общий характер и способны решать множество задач нелинейной оптимизации.

*Нелинейная оптимизационная задача* включает множество независимых переменных, детерминистским образом связанных с значением целевой функции. Целью является нахождение такого множества значений независимых переменных, которое минимизирует (или максимизирует) целевую функцию. Рассмотрим, например, нахождение минимума функции  $F(x) = 3x^3 + 6x^2 - 2x + 3$ .

Здесь имеется единственная независимая переменная  $x$ , управляющая значением целевой функции  $F(x)$ , которая должна быть минимизирована. Эта простая функция легко минимизируется с помощью методов дифференциального исчисления, однако минимизировать подобным образом более сложные функции от большого числа переменных может оказаться затруднительным.

Во многих практических ситуациях функциональная связь между независимыми переменными и целевой функцией неизвестна и фактически не может быть известной. Сложный химический процесс может не иметь адекватной математической модели. Единственными измеряемыми величинами могут быть «выход», «качество», «цена» и т. д., которые являются неизвестными функциями от большого числа таких независимых переменных, как температура, время и характеристики сырья.

Подобная задача может решаться следующим образом:

1. Система наблюдается и собираются данные для составления обучающего множества. Каждый элемент обучающего множества состоит из замеров во время наблюдений и включает значения всех входов (входной вектор) и всех выходов (выходной вектор).
2. Сеть обучается на этом обучающем множестве. Обучение состоит из предъявления входного вектора, вычисления выходного вектора, сравнения выходного вектора с входным вектором, полученным в процессе наблюдений, и коррекции весов.

минимизирующей разность между ними. Каждый входной вектор предъявляется по очереди, и сеть частично обучается. После большого числа предъявления входных векторов сеть сойдется к решению, которое минимизирует разность между желаемыми и измеренными выходами системы. Фактически сеть строит внутреннюю модель неизвестной системы. Если обучающее множество достаточно велико, сеть сойдется к точной модели системы. Если сети предъявить некоторый входной вектор, отличный от любого из векторов, предъявленных при обучении, то полностью обученная сеть выдаст тот же самый выходной вектор, что и настоящая система.

3. Максимизируется целевая функция. Целевая функция выходов должна быть сконструирована таким образом, чтобы выражать степень «удовлетворительности» результата. Теперь входы становятся переменными для обученной сети. Они подстраиваются с помощью того же самого обучающего алгоритма, который применялся для выставления весов на шаге 2, однако используются для максимизации целевой функции.

Во многих случаях могут присутствовать ограничения, накладываемые задачей. Например, может быть невозможно физически брать значения переменных вне некоторого диапазона. Эти ограничения (которые могут быть сложными выражениями) могут быть легко учтены отбрасыванием на шаге 3 любого изменения входной переменной, которое нарушает ограничение.

Это обобщение метода стохастической оптимизации позволяет его использовать для широкого круга оптимизационных задач. Можно применять и другие методы, но стохастический метод позволяет преодолеть трудности, обусловленные локальными минимумами, с которыми сталкивается метод обратного распространения и другие методы градиентного спуска. К сожалению, вероятностная природа процесса обучения может приводить к большому времени сходимости. Использование методов псевдотеплоемкости может существенно уменьшить это время, но процесс все равно остается медленным.

## ОБРАТНОЕ РАСПРОСТРАНЕНИЕ И ОБУЧЕНИЕ КОШИ

Обратное распространение обладает преимуществом прямого поиска, т. е. веса всегда корректируются в направлении, минимизирующем функцию ошибки. Хотя время обучения и велико, оно существенно меньше, чем при случайном поиске, выполняемом машиной Коши, когда находится глобальный минимум, но многие шаги выполняются в неверном направлении, что отнимает много времени.

Соединение этих двух методов дало хорошие результаты [7]. Коррекция весов, равная сумме, вычисленной алгоритмом обратного распространения, и случайный шаг, задаваемый алгоритмом Коши, приводят к системе, которая сходится и находит глобальный минимум быстрее, чем система, обучаемая каждым из методов в отдельности. Простая эвристика используется для избежания паралича сети, который может иметь место как при обратном распространении, так и при обучении по методу Коши.

### Трудности, связанные с обратным распространением

Несмотря на мощь, продемонстрированную методом обратного распространения, при его применении возникает ряд трудностей, часть из которых, однако, облегчается благодаря использованию нового алгоритма.

**Сходимость.** В работе [5] доказательство сходимости дается на языке дифференциальных уравнений в частных производных, что делает его справедливым лишь в том случае, когда коррекция весов выполняется с помощью бесконечно малых шагов. Так как это ведет к бесконечному времени сходимости, то оно теряет силу в практических применениях. В действительности нет доказательства, что обратное распространение будет сходиться при конечном размере шага. Эксперименты показывают, что сети обычно обучаются, но время обучения велико и непредсказуемо.

**Локальные минимумы.** В обратном распространении для коррекции весов сети используется градиентный спуск, продвигающийся к минимуму в соответствии с локальным наклоном поверхности ошибки. Он хорошо работает в случае сильно изрезанных невыпуклых поверхностей, которые встречаются в практических задачах. В одних случаях локальный минимум является приемлемым решением, в других случаях он неприемлем.

Даже после того как сеть обучена, невозможно сказать, найден ли с помощью обратного распространения глобальный минимум. Если решение неудовлетворительно, приходится давать весам новые начальные случайные значения и повторно обучать сеть без гарантии, что обучение закончится на этой попытке или что глобальный минимум вообще будет когда либо найден.

**Паралич.** При некоторых условиях сеть может при обучении попасть в такое состояние, когда модификация весов не ведет к действительным изменениям сети. Такой «паралич сети» является серьезной проблемой: один раз возникнув, он может увеличить время обучения на несколько порядков.

Паралич возникает, когда значительная часть нейронов получает веса, достаточно большие, чтобы дать большие значения NET. Это приводит к тому, что величина OUT приближается к своему предельному значению, а производная от сжимающей функции приближается к нулю. Как мы видели, алгоритм обратного распространения при вычислении величины изменения веса использует эту производную в формуле в качестве коэффициента. Для пораженных параличом нейронов близость производной к нулю приводит к тому, что изменение веса становится близким к нулю.

Если подобные условия возникают во многих нейронах сети, то обучение может замедлиться до почти полной остановки.

Нет теории, способной предсказывать, будет ли сеть парализована во время обучения или нет. Экспериментально установлено, что малые размеры шага реже приводят к параличу, но шаг, малый для одной задачи, может оказаться большим для другой. Цена же паралича может быть высокой. При моделировании многие часы машинного времени могут уйти на то, чтобы выйти из паралича.

### Трудности с алгоритмом обучения Коши

Несмотря на улучшение скорости обучения, даваемое машиной Коши по сравнению с машиной Больцмана, время сходимости все еще может в 100 раз превышать время для алгоритма обратного распространения. Отметим, что сетевой паралич особенно опасен для алгоритма обучения Коши, в особенности для сети с нелинейностью типа логистической функции. Бесконечная дисперсия распределения Коши приводит к изменениям весов неограниченной величины. Далее, большие изменения весов будут иногда приниматься даже в тех случаях, когда они неблагоприятны, часто приводя к сильному насыщению сетевых нейронов с вытекающим отсюда риском паралича.

### Комбинирование обратного распространения с обучением Коши

Коррекция весов в комбинированном алгоритме, использующем обратное распространение и обучение Коши, состоит из двух компонент: (1) направленной компоненты, вычисляемой с использованием алгоритма обратного распространения, и (2) случайной компоненты, определяемой распределением Коши.

Эти компоненты вычисляются для каждого веса, и их сумма является величиной, на которую изменяется вес. Как и в алгоритме Коши, после вычисления изменения веса вычисляется целевая функция. Если имеет место улучшение, изменение сохраняется. В противном случае оно сохраняется с вероятностью, определяемой распределением Больцмана.

Коррекция веса вычисляется с использованием представленных ранее уравнений для каждого из алгоритмов:

$$w_{mn,k}(n+1) = w_{mn,k}(n) + \eta [\alpha \Delta w_{mn,k}(n) + (1 - \alpha) \delta_{n,k} \text{OUT}_{mj}] + (1 - \eta) x_c,$$

где  $\eta$  – коэффициент, управляющий относительными величинами Коши и обратного распространения в компонентах весового шага. Если  $\eta$  приравняется нулю, система становится полностью машиной Коши. Если  $\eta$  приравняется единице, система становится машиной обратного распространения.

Изменение лишь одного весового коэффициента между вычислениями весовой функции неэффективно. Оказалось, что лучше сразу изменять все веса целого слоя, хотя для некоторых задач может оказаться выгоднее иная стратегия.

**Преодоление сетевого паралича комбинированным методом обучения.** Как и в машине Коши, если изменение веса ухудшает целевую функцию, – с помощью распределения Больцмана решается, сохранить ли новое значение веса или восстановить предыдущее значение. Таким образом, имеется конечная вероятность того, что ухудшающее множество приращений весов будет сохранено. Так как распределение Коши имеет бесконечную дисперсию (диапазон изменения тангенса простирается от  $-\infty$  до  $+\infty$  на области определения), то весьма вероятно возникновение больших приращений весов, часто приводящих к сетевому параличу.

Очевидное решение, состоящее в ограничении диапазона изменения весовых шагов, ставит вопрос о математической корректности полученного таким образом алгоритма. В работе [6] доказана сходимость системы к глобальному минимуму лишь для исходного алгоритма. Подобного доказательства при искусственном ограничении размера шага не существует. В действительности экспериментально выявлены случаи, когда для реализации некоторой функции требуются большие веса, и два больших веса, вычитаясь, дают малую разность.

Другое решение состоит в рандомизации весов тех нейронов, которые оказались в состоянии насыщения. Недостатком его является то, что оно может серьезно нарушить обучающий процесс, иногда затягивая его до бесконечности.

Для решения проблемы паралича был найден метод, не нарушающий достигнутого обучения. Насыщенные нейроны выявляются с помощью измерения их сигналов OUT. Когда величина OUT приближается к своему предельному значению, положительному или отрицательному, на веса, питающие этот нейрон, действует сжимающая функция. Она подобна используемой для получения нейронного сигнала OUT, за исключением того, что диапазон ее изменения является интервал  $(+5, -5)$  или другое подходящее множество. Тогда модифицированные весовые значения равны

$$w_{mn} = \frac{-5 + 10}{1 + \exp(-w_{mn}/5)}.$$

Эта функция сильно уменьшает величину очень больших весов, воздействие на малые веса значительно более слабое. Далее она поддерживает симметрию, сохраняя небольшие различия между большими весами. Экспериментально было показано, что эта функция выводит нейроны из состояния насыщения без нарушения достигнутого в сети обучения. Не было затрачено серьезных усилий для оптимизации используемой функции, другие значения констант могут оказаться лучшими.

**Экспериментальные результаты.** Комбинированный алгоритм, использующий обратное распространение и обучение Коши, применялся для обучения нескольких больших сетей. Например, этим методом была успешно обучена система, распознающая рукописные китайские иероглифы [6]. Все же время обучения может оказаться большим (приблизительно 36 часов машинного времени уходило на обучение).

В другом эксперименте эта сеть обучалась на задаче ИСКЛЮЧАЮЩЕЕ ИЛИ, которая была использована в качестве теста для сравнения с другими алгоритмами. Для сходимости сети в среднем требовалось около 76 предъявлении обучающего множества. В качестве сравнения можно указать, что при использовании обратного распространения в среднем требовалось около 245 предъявлении для решения этой же задачи [5] и 4986 итераций при использовании обратного распространения второго порядка.

Ни одно из обучений не привело к локальному минимуму, о которых сообщалось в [5]. Более того, ни одно из 160 обучений не обнаружило неожиданных патологий, сеть всегда правильно обучалась.

Эксперименты же с чистой машиной Коши привели к значительно большим временам обучения. Например, при  $\rho = 0,002$  для обучения сети в среднем требовалось около 2284 предъявлении обучающего множества.

### Обсуждение

Комбинированная сеть, использующая обратное распространение и обучение Коши, обучается значительно быстрее, чем каждый из алгоритмов в отдельности, и относительно нечувствительна к величинам коэффициентов. Сходимость к глобальному минимуму гарантируется алгоритмом Коши, в сотнях экспериментов по обучению сеть ни разу не попадала в ловушки локальных минимумов. Проблема сетевого паралича была решена с помощью алгоритма селективного сжатия весов, который обеспечил сходимость во всех предъявленных тестовых задачах без существенного увеличения обучающего времени.

Несмотря на такие обнадеживающие результаты, метод еще не исследован до конца, особенно на больших задачах. Значительно большая работа потребуется для определения его достоинств и недостатков.

### Литература

1. Geman S., Geman D. 1984. Stochastic relaxation, Gibbs distribution and Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6:721-41.
2. Hinton G. E., Sejnowski T. J. 1986. Learning and relearning in Boltzmann machines. In Parallel distributed processing, vol. 1, p. 282-317. Cambridge, MA: MIT Press.
3. Metropolis N., Rosenbluth A. W.-Rosenbluth M. N., Teller A. N., Teller E. 1953. Equations of state calculations by fast computing machines. Journal of Chemistry and Physics. 21:1087-91.
4. Parker D. B. 1987. Optimal algorithms for adaptive networks. Second order Hebbian learning. In Proceedings of the IEEE First International Conference on Neural Networks, eds. M. Caudill and C. Buller, vol. 2, pp. 593-600. San Diego, CA: SOS Printing.
5. Rumelhart D. E. Hinton G. E. Williams R. J. 1986. Learning internal representations by error propagation. In Parallel distributed processing, vol. 1, pp. 318-62. Cambridge, MA: MIT Press.
6. Szu H., Hartley R. 1987. Fast Simulated annealing. Physics Letters. 1222(3,4): 157-62.
7. Wassermann P. D. 1988. Combined backpropagation/Cauchy machine. Neural Networks. Abstracts of the First INNS Meeting, Boston 1988, vol. 1, p. 556. Elmsford, NY. Pergamon Press.

## Глава 6.

### Сети Хопфилда

Сети, рассмотренные в предыдущих главах, не имели обратных связей, т. е. связей, идущих от выходов сетей и их входам. Отсутствие обратной связи гарантирует безусловную устойчивость сетей. Они не могут войти в режим, когда выход беспрерывно блуждает от состояния к состоянию и не пригоден к использованию. Но это весьма желательное свойство достигается не бесплатно, сети без обратных связей обладают более ограниченными возможностями по сравнению с сетями с обратными связями.

Так как сети с обратными связями имеют пути, передающие сигналы от выходов к входам, то отклик таких сетей является динамическим, т. е. после приложения нового входа вычисляется выход и, передаваясь по сети обратной связи, модифицирует вход. Затем выход повторно вычисляется, и процесс повторяется снова и снова. Для устойчивой сети

последовательные итерации приводят к все меньшим изменениям выхода, пока в конце концов выход не становится постоянным. Для многих сетей процесс никогда не заканчивается, такие сети называют неустойчивыми. Неустойчивые сети обладают интересными свойствами и изучались в качестве примера хаотических систем. Однако такой большой предмет, как хаос, находится за пределами этой книги. Вместо этого мы сконцентрируем внимание на устойчивых сетях, т. е. на тех, которые в конце концов дают постоянный выход.

Проблема устойчивости ставила в тупик первых исследователей. Никто не был в состоянии предсказать, какие из сетей будут устойчивыми, а какие будут находиться в постоянном изменении. Более того, проблема представлялась столь трудной, что многие исследователи были настроены пессимистически относительно возможности ее решения. К счастью, в работе [2] была получена теорема, описавшая подмножество сетей с обратными связями, выходы которых в конце концов достигают устойчивого состояния. Это замечательное достижение открыло дорогу дальнейшим исследованиям и сегодня многие ученые занимаются исследованием сложного поведения и возможностей этих систем.

Дж. Хопфилд сделал важный вклад как в теорию, так и в применение систем с обратными связями. Поэтому некоторые из конфигураций известны как сети Хопфилда. Из обзора литературы видно, что исследованием этих и сходных систем занимались многие. Например, в работе [4] изучались общие свойства сетей, аналогичных многим, рассмотренным здесь. Работы, цитируемые в списке литературы в конце главы, не направлены на то, чтобы дать исчерпывающую библиографию по системам с обратными связями. Скорее они являются лишь доступными источниками, которые могут служить для объяснения, расширения и обобщения содержания этой книги.

## КОНФИГУРАЦИИ СЕТЕЙ С ОБРАТНЫМИ СВЯЗЯМИ

На рис. 6.1 показана сеть с обратными связями, состоящая из двух слоев. Способ представления несколько отличается от использованного в работе Хопфилда и других, но эквивалентен им с функциональной точки зрения, а также хорошо связан с сетями, рассмотренными в предыдущих главах. Нулевой слой, как и на предыдущих рисунках, не выполняет вычислительной функции, а лишь распределяет выходы сети обратно на входы. Каждый нейрон первого слоя вычисляет взвешенную сумму своих входов, давая сигнал NET, который затем с помощью нелинейной функции  $F$  преобразуется в сигнал OUT. Эти операции сходны с нейронами других сетей (см. гл. 2).

### Бинарные системы

В первой работе Хопфилда [6] функция  $F$  была просто пороговой функцией. Выход такого нейрона равен единице, если взвешенная сумма выходов с других нейронов больше порога  $T_j$ , в противном случае она равна нулю. Он вычисляется следующим образом:

$$NET_j = \sum_{i \neq j} w_{ij} OUT_i + IN_j, \quad (6.1)$$

$$OUT_j = 1, \text{ если } NET_j > T_j,$$

$$OUT_j = 0, \text{ если } NET_j < T_j,$$

$$OUT_j \text{ не изменяется, если } NET_j = T_j,$$

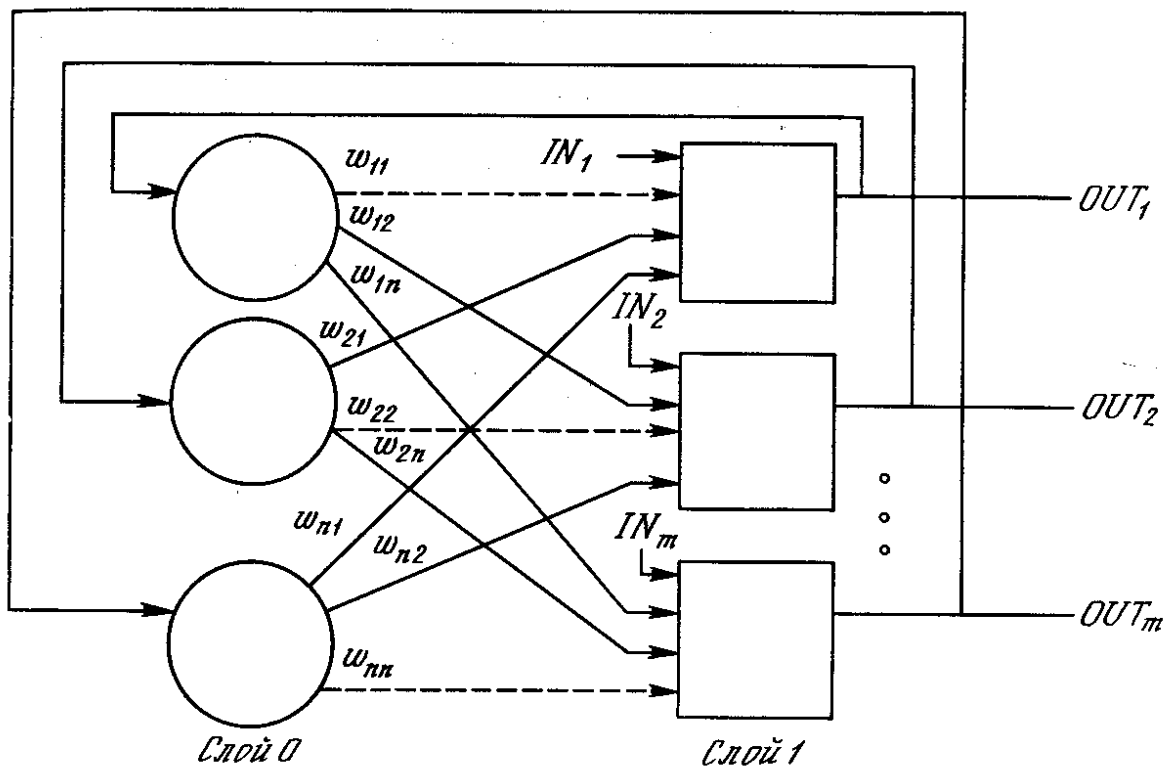


Рис. 6.1. Однослойная сеть с обратными связями.  
Пунктирные линии обозначают нулевые веса

Состояние сети – это просто множество текущих значений сигналов OUT от всех нейронов. В первоначальной сети Хопфилда состояние каждого нейрона менялось в дискретные случайные моменты времени, в последующей работе состояния нейронов могли меняться одновременно. Так как выходом бинарного нейрона может быть только ноль или единица (промежуточных уровней нет), то текущее состояние сети является двоичным числом, каждый бит которого является сигналом OUT некоторого нейрона.

Функционирование сети легко визуализируется геометрически. На рис. 6.2а показан случай двух нейронов в выходном слое, причем каждой вершине квадрата соответствует одно из четырех состояний системы (00, 01, 10, 11). На рис. 6.2б показана трехнейронная система, представленная кубом (в трехмерном пространстве), имеющим восемь вершин, каждая из которых помечена трехбитовым бинарным числом. В общем случае система с  $n$  нейронами имеет  $2^n$  различных состояний и представляется  $n$ -мерным гиперкубом.

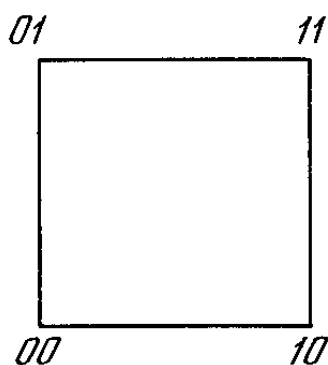


Рис. 6.2а. Два нейрона порождают систему с четырьмя состояниями

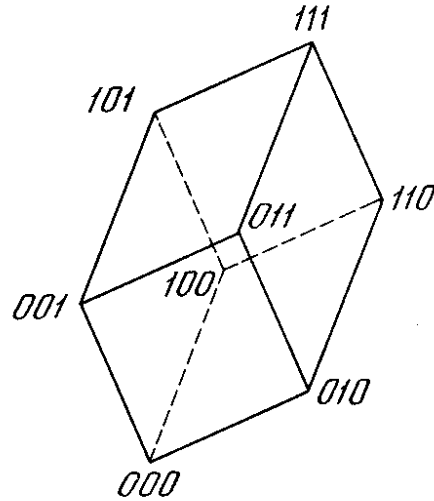


Рис. 6.26. Три нейрона порождают систему с восемью состояниями

Когда подается новый входной вектор, сеть переходит из вершины в вершину, пока не стабилизируется. Устойчивая вершина определяется сетевыми весами, текущими входами и величиной порога. Если входной вектор частично неправилен или неполон, то сеть стабилизируется в вершине, ближайшей к желаемой.

#### Устойчивость

Как и в других сетях, веса между слоями в этой сети могут рассматриваться в виде матрицы  $\mathbf{W}$ . В работе [2] показано, что сеть с обратными связями является устойчивой, если ее матрица симметрична и имеет нули на главной диагонали, т. е. если  $w_{ij} = w_{ji}$  и  $w_{ii} = 0$  для всех  $i$ .

Устойчивость такой сети может быть доказана с помощью элегантного математического метода. Допустим, что найдена функция, которая всегда убывает при изменении состояния сети. В конце концов эта функция должна достичь минимума и прекратить изменение, гарантируя тем самым устойчивость сети. Такая функция, называемая функцией Ляпунова, для рассматриваемых сетей с обратными связями может быть введена следующим образом:

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} \text{OUT}_i \text{OUT}_j - \sum_j I_j \text{OUT}_j + \sum_j T_j \text{OUT}_j \quad (6.2)$$

где  $E$  – искусственная энергия сети;  $w_{ij}$  – вес от выхода нейрона  $i$  к входу нейрона  $j$ ;  $\text{OUT}_j$  – выход нейрона  $j$ ;  $I_j$  – внешний вход нейрона  $j$ ;  $T_j$  – порог нейрона  $j$ .

Изменение энергии  $E$ , вызванное изменением состояния  $j$ -нейрона, есть

$$\delta E = \left[ \sum_{i \neq j} (w_{ij} \text{OUT}_i) + I_j - T_j \right] \delta \text{OUT}_j = -[\text{NET}_j - T_j] \delta \text{OUT}_j \quad (6.3)$$

где  $\delta \text{OUT}_j$  – изменение выхода  $j$ -го нейрона.

Допустим, что величина  $\text{NET}$  нейрона  $j$  больше порога. Тогда выражение в скобках будет положительным, а из Уравнения (6.1) следует, что выход нейрона  $j$  должен измениться в положительную сторону (или остаться без изменения). Это значит, что  $\delta \text{OUT}_j$  может быть только положительным или нулем и  $\delta E$  должно быть отрицательным. Следовательно, энергия сети должна либо уменьшиться, либо остаться без изменения.

Далее, допустим, что величина  $\text{NET}$  меньше порога. Тогда величина  $\delta \text{OUT}_j$  может быть только отрицательной или нулем. Следовательно, опять энергия должна уменьшиться или остаться без изменения.

И окончательно, если величина  $\text{NET}$  равна порогу,  $\delta_j$  равна нулю и энергия остается без изменения.

Это показывает, что любое изменение состояния нейрона либо уменьшит энергию, либо оставит ее без изменения. Благодаря такому непрерывному стремлению к уменьшению энергии в конце концов должна достигнута минимума и прекратить изменение. По определению такая сеть является устойчивой.

Симметрия сети является достаточным, но не необходимым условием для устойчивости системы. Имеется много устойчивых систем (например, все сети прямого действия!), которые ему не удовлетворяют. Можно продемонстрировать примеры, в которых незначительное отклонение от симметрии может приводить к непрерывным осцилляциям. Однако приближенной симметрии обычно достаточно для устойчивости систем.

### Ассоциативная память

Человеческая память ассоциативна, т. е. некоторое воспоминание может порождать большую связанную с ним область. Например, несколько музыкальных тактов могут вызвать целую гамму чувственных воспоминаний, включая пейзажи, звуки и запахи. Напротив, обычная компьютерная память является локально адресуемой, предъявляется адрес и извлекается информация по этому адресу.

Сеть с обратной связью формирует ассоциативную память. Подобно человеческой памяти по заданной части нужной информации вся информация извлекается из «памяти». Чтобы организовать ассоциативную память с помощью сети с обратными связями, веса должны выбираться так, чтобы образовывать энергетические минимумы в нужных вершинах единичного гиперкуба.

Хопфилд разработал ассоциативную память с непрерывными выходами, изменяющимися в пределах от +1 до -1, соответствующих двоичным значениям 0 и 1, Запоминаемая информация кодируется двоичными векторами и хранится в весах согласно следующей формуле:

$$w_{ij} = \sum_{d=1..m} (\text{OUT}_{i,d} \text{OUT}_{j,d}) \quad (6.4)$$

где  $m$  – число запоминаемых выходных векторов;  $d$  – номер запоминаемого выходного вектора;  $\text{OUT}_{ij}$  –  $i$ -компонента запоминаемого выходного вектора.

Это выражение может стать более ясным, если заметить, что весовой массив  $\mathbf{W}$  может быть найден вычислением внешнего произведения каждого запоминаемого вектора с самим собой (если требуемый вектор имеет  $n$  компонент, то эта операция образует матрицу размером  $n \times n$ ) и суммированием матриц, полученных таким образом. Это может быть записано в виде

$$\mathbf{W} = \sum_i \mathbf{D}_i^t \mathbf{D}_i, \quad (6.5)$$

где  $\mathbf{D}_i$  –  $i$ -й запоминаемый вектор-строка.

Как только веса заданы, сеть может быть использована для получения запомненного выходного вектора по данному входному вектору, который может быть частично неправильным или неполным. Для этого выходам сети сначала придают значения этого входного вектора. Затем входной вектор убирается и сети предоставляется возможность «расслабиться», опустившись в ближайший глубокий минимум. Сеть идущая по локальному наклону функции энергии, может быть захвачена локальным минимумом, не достигнув наилучшего в глобальном смысле решения.

### Непрерывные системы

В работе [7] рассмотрены модели с непрерывной активационной функцией  $F$ , точнее моделирующей биологический нейрон. В общем случае это S-образная или логистическая функция

$$F(x) = \frac{1}{1 + \exp(-\lambda \text{NET})}, \quad (6.6)$$

где  $\lambda$  – коэффициент, определяющий крутизну сигмоидальной функции. Если  $\lambda$  велико,  $F$  приближается к описанной ранее пороговой функции. Небольшие значения  $\lambda$  дают более пологий наклон.

Как и для бинарных систем, устойчивость гарантируется, если веса симметричны, т. е.  $w_{ij} = w_{ji}$  и  $w_{ii} = 0$  при всех  $i$ . Функция энергии, доказывающая устойчивость подобных систем, была сконструирована, но она не рассматривается здесь из-за своего концептуального сходства с дискретным случаем. Интересующиеся читатели могут обратиться к работе [2] для более полного рассмотрения этого важного предмета.



Если  $\lambda$  велико, непрерывные системы функционируют подобно дискретным бинарным системам, окончательно стабилизируясь со всеми выходами, близкими нулю или единице, т. е. в вершине единичного гиперкуба. С уменьшением  $\lambda$  устойчивые точки удаляются от вершин, последовательно исчезая по мере приближения  $\lambda$  к нулю. На рис. 6.3 показаны линии энергетических уровней непрерывной системы с двумя нейронами.

### Сети Хопфилда и машина Больцмана

Недостатком сетей Хопфилда является их тенденция стабилизироваться в локальном, а не глобальном минимуме функции энергии. Эта трудность преодолевается в основном с помощью класса сетей, известных под названием машин Больцмана, в которых изменения состояний нейронов обусловлены статистическими, а не детерминированными закономерностями. Существует тесная аналогия между этими методами и отжигом металла, поэтому и сами методы часто называют имитацией отжига.

### Термодинамические системы

Металл отжигают, нагревая его до температуры, превышающей точку его плавления, а затем давая ему медленно остыть. При высоких температурах атомы, обладая высокими энергиями и свободой перемещения, случайным образом принимают все возможные конфигурации. При постепенном снижении температуры энергии атомов уменьшаются, и система в целом стремится принять конфигурацию с минимальной энергией. Когда охлаждение завершено, достигается состояние глобального минимума энергии.

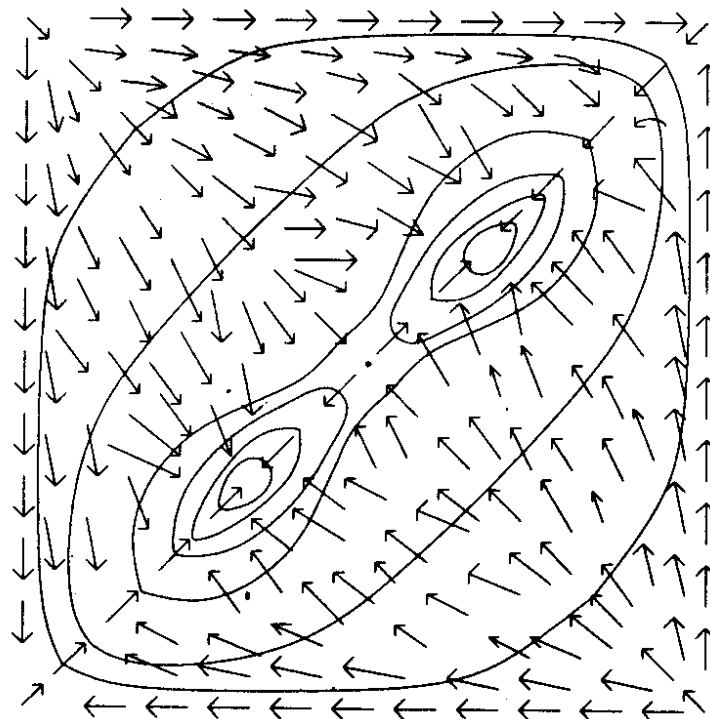


Рис. 6.3. Линии энергетических уровней

При фиксированной температуре распределение энергий системы определяется вероятностным фактором Больцмана

$$\exp(-E/kT),$$

где  $E$  – энергия системы;  $k$  – постоянная Больцмана;  $T$  – температура.

Отсюда можно видеть, что имеется конечная вероятность того, что система обладает высокой энергией даже при низких температурах. Сходным образом имеется небольшая, но вычисляемая вероятность, что чайник с водой на огне замерзнет, прежде чем закипеть.

Статистическое распределение энергий позволяет системе выходить из локальных минимумов энергии. В то же время вероятность высокоэнергетических состояний быстро уменьшается со снижением температуры. Следовательно, при низких температурах имеется сильная тенденция занять низкоэнергетическое состояние.

### Статистические сети Хопфилда

Если правила изменения состояний для бинарной сети Хопфилда заданы статистически, а не детерминированно, как в уравнении (6.1), то возникает система, имитирующая отжиг. Для ее реализации вводится вероятность изменения веса как функция от величины, на которую выход нейрона OUT превышает его порог. Пусть

$$E_k = NET_k - \theta_k,$$

где  $NET_k$  – выход NET нейрона  $k$ ;  $\theta$  – порог нейрона  $k$ , и

$$P_k = \frac{1}{1 + \exp(-\delta E_k / T)},$$

(отметьте вероятностную функцию Больцмана в знаменателе), где  $T$  – искусственная температура.

В стадии функционирования искусственной температуре  $T$  приписывается большое значение, нейроны устанавливаются в начальном состоянии, определяемом входным вектором, и сети предоставляется возможность искать минимум энергии в соответствии с нижеследующей процедурой:

1. Приписать состоянию каждого нейрона с вероятностью  $p_k$  значение единица, а с вероятностью  $1-p_k$  – нуль.
2. Постепенно уменьшать искусственную температуру и повторять шаг 1, пока не будет достигнуто равновесие.

### Обобщенные сети

Принцип машины Больцмана может быть перенесен на сети практически любой конфигурации, хотя устойчивость не гарантируется. Для этого достаточно выбрать одно множество нейронов в качестве входов и другое множество в качестве выходов. Затем придать входному множеству значения входного вектора и предоставить сети возможность релаксировать в соответствии с описанными выше правилами 1 и 2.

Процедура обучения для такой сети, описанная в [5], состоит из следующих шагов:

1. Вычислить закрепленные вероятности.
  - а) придать входным и выходным нейронам значения обучающего вектора;
  - б) предоставить сети возможность искать равновесие;
  - в) записать выходные значения для всех нейронов;
  - г) повторить шаги от а до в для всех обучающих векторов;
  - д) вычислить вероятность  $P_{ij}^+$ , т. е. по всему множеству обучающих векторов вычислить вероятность того, что значения обоих нейронов равны единице.
2. Вычислить незакрепленные вероятности.
  - а) предоставить сети возможность «свободного движения» без закрепления входов или выходов, начав со случайного состояния;
  - б) повторить шаг 2а много раз, регистрируя значения всех нейронов;
  - в) вычислить вероятность  $P_{ij}^-$ , т. е. вероятность того, что значения обоих нейронов равны единице.
3. Скорректировать веса сети следующим образом:

$$\delta w_{ij} = \eta(P_{ij}^+ - P_{ij}^-),$$

где  $\delta w_{ij}$  – изменение веса  $w_{ij}$ ,  $\eta$  – коэффициент скорости обучения.

## ПРИЛОЖЕНИЯ

### Аналого-цифровой преобразователь

В недавних работах [8,10] рассматривалась электрическая схема, основанная на сети с обратной связью, реализующая четырехбитовый аналого-цифровой преобразователь. На рис. 6.4 показана блок-схема этого устройства с усилителями, выполняющими роль искусственных нейронов. Сопротивления, выполняющие роль весов, соединяют выход каждого нейрона с входами всех остальных. Чтобы удовлетворить условию устойчивости, выход нейрона не соединялся сопротивлением с его собственным входом, а веса брались симметричными, т. е. сопротивление от выхода нейрона  $i$  к входу нейрона  $j$  имело ту же величину, что и сопротивление от выхода нейрона  $j$  к входу нейрона  $i$ .

Заметим, что усилители имеют прямой и инвертированный выходы. Это позволяет с помощью обычных положительных сопротивлений реализовывать и те случаи, когда веса должны быть отрицательными. На рис. 6.4 показаны все возможные сопротивления, при этом никогда не возникает необходимости присоединять как прямой, так и инвертированный выходы нейрона к входу другого нейрона.

В реальной системе каждый усилитель обладает конечным входным сопротивлением и входной емкостью, что должно учитываться при расчете динамической характеристики. Для устойчивости сети не требуется равенства этих параметров для всех усилителей и их симметричности. Так как эти параметры влияют лишь на время получения решения, а не на само решение, для упрощения анализа они исключены.

Предполагается, что используется пороговая функция (предел сигмоидальной функции при  $\lambda$ , стремящемся к бесконечности). Далее, все выходы изменяются в начале дискретных интервалов времени, называемых эпохами. В начале каждой эпохи исследуется сумма входов каждого нейрона. Если она больше порога, выход принимает единичное значение, если меньше – нулевое. На протяжении эпохи выходы нейронов не изменяются.

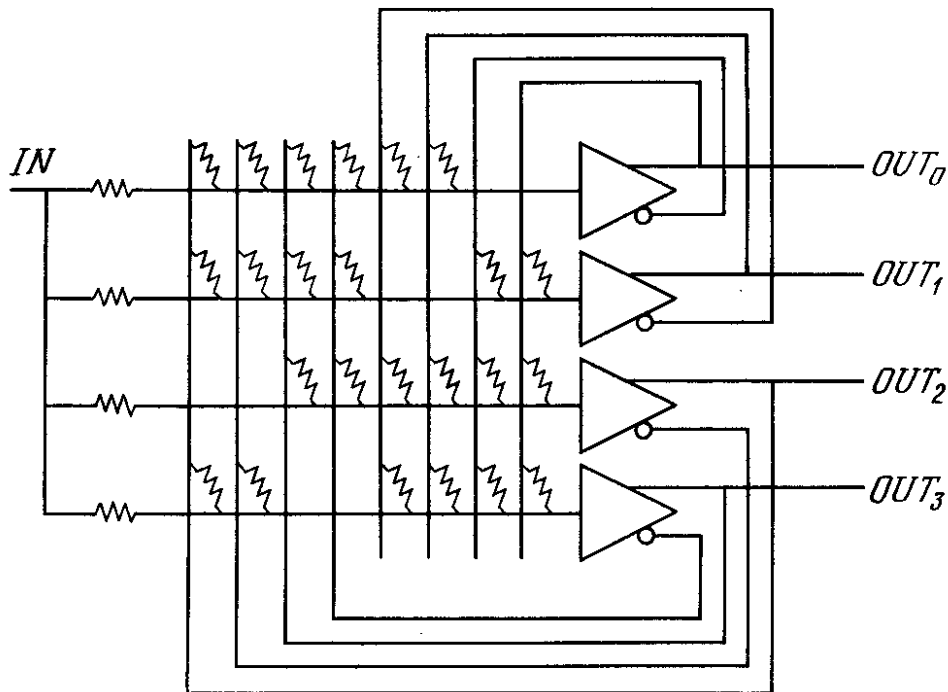


Рис. 6.4. Четырехбитовый аналого-цифровой преобразователь, использующий сеть Хопфилда

Целью является такой выбор сопротивлений (весов), что непрерывно растущее напряжение  $X$ , приложенное к одноходовому терминалу, порождает множество из четырех выходов, представляющих двоичную запись числа, величина которого приблизительно равна входному напряжению (рис. 6.5). Определим сначала функцию энергии следующим образом:

$$E = -\frac{1}{2} \left( X - \sum_j 2^j \text{OUT}_j \right)^2 + \sum_j 2^{2j-1} \text{OUT}_j (1 - \text{OUT}_j), \quad (6.7)$$

где  $X$  – входное напряжение.

Когда  $E$  минимизировано, то получаются нужные выходы. Первое выражение в скобках минимизируется, когда двоичное число, образованное выходами, наиболее близко (в среднеквадратичном смысле) к аналоговой величине входа  $X$ . Второе выражение в скобках обращается в нуль, когда все выходы равны 1 или 0, тем самым накладывая ограничение, что выходы принимают только двоичные значения.

Если уравнение (6.7) перегруппировать и сравнить с уравнением (6.2), то получим следующее выражение для весов:

$$W_{ij} = -2^{i+j}, y_i = 2^i, \quad (6.8)$$

где  $w_{ij}$  – проводимость (величина, обратная сопротивлению) от выхода нейрона  $i$  к входу нейрона  $j$  (равная также проводимости от выхода нейрона  $j$  к входу нейрона  $i$ );  $y_i$  – проводимость от входа  $X$  к входу нейрона  $i$ .

Чтобы получить схему с приемлемыми значениями сопротивлений и потребляемой мощности, все веса должны быть промасштабированы.

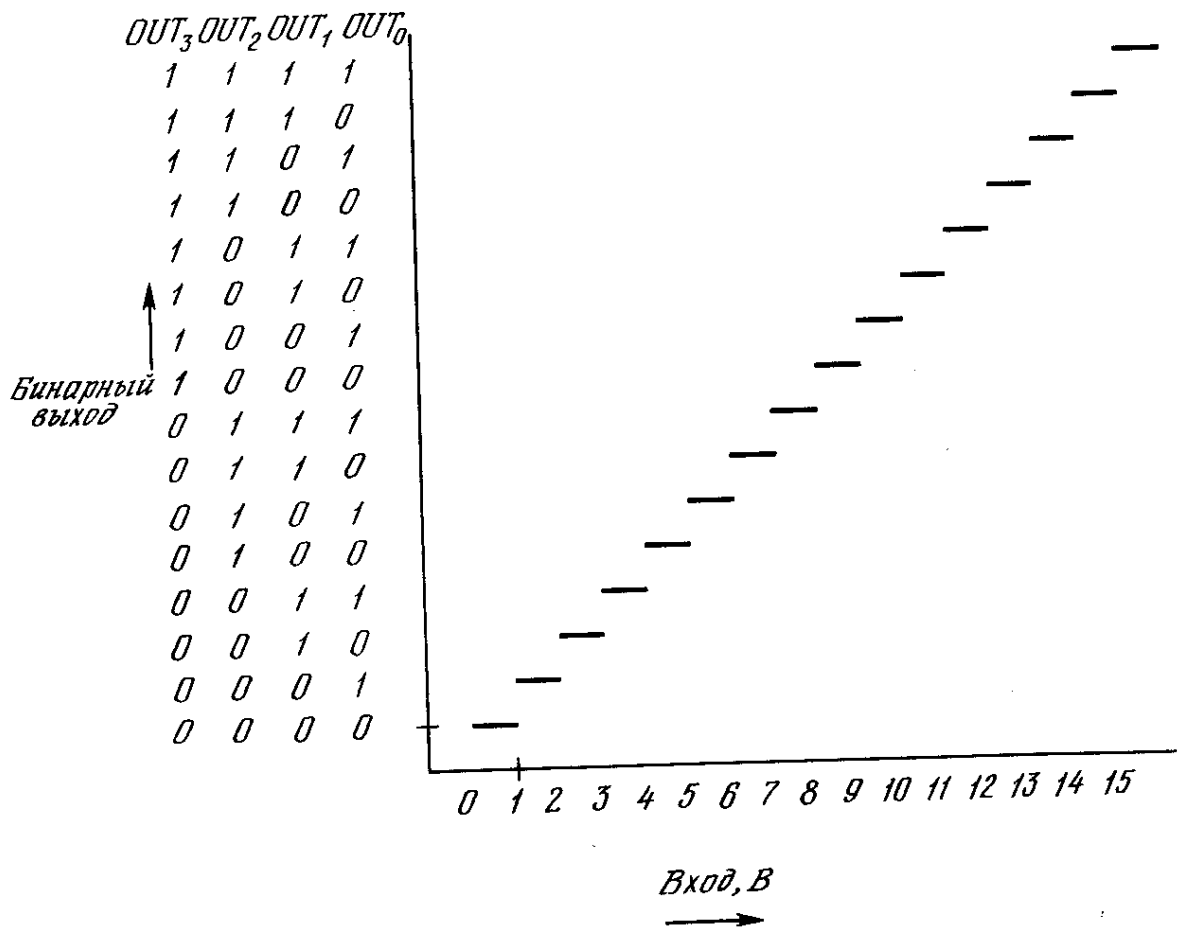


Рис. 6.5. Идеальная характеристика четырехбитового аналого-цифрового преобразователя

Идеальная выходная характеристика, изображенная на рис. 6.5, будет реализована лишь в том случае, если входы устанавливаются в нуль перед выполнением преобразования. Если этого не делать, сеть может попасть в локальный минимум энергии и дать неверный выход.

#### Задача коммивояжера

Задача коммивояжера является оптимизационной задачей, часто возникающей на практике. Она может быть сформулирована следующим образом: для некоторой группы городов с заданными расстояниями между ними требуется найти кратчайший маршрут с посещением каждого города один раз и с возвращением в исходную точку. Было доказано, что эта задача принадлежит большому множеству задач, называемых «NP-полными» (недетерминистски полиномиальными) [3]. Для NP-полных задач не известно лучшего метода решения, чем полный перебор всех возможных вариантов, и, по мнению большинства

математиков, маловероятно, чтобы лучший метод был когда либо найден. Так как такой полный поиск практически неосуществим для большого числа городов, то эвристические методы используются для нахождения приемлемых, хотя и неоптимальных решений.

Описанное в работе [8] решение, основанное на сетях с обратными связями, является типичным в этом отношении. Все же ответ получается так быстро, что в определенных случаях метод может оказаться полезным.

Допустим, что города, которые необходимо посетить, помечены буквами А, В, С и D, а расстояния между парами городов есть  $d_{ab}$ ,  $d_{bc}$  и т. д.

Решением является упорядоченное множество из  $n$  городов. Задача состоит в отображении его в вычислительную сеть с использованием нейронов в режиме с большой крутизной характеристики ( $\lambda$  приближается к бесконечности). Каждый город представлен строкой из  $n$  нейронов. Выход одного и только одного нейрона из них равен единице (все остальные равны нулю). Этот равный единице выход нейрона показывает порядковый номер, в котором данный город посещается при обходе. На рис. 6.6 показан случай, когда город С посещается первым, город А – вторым, город D – третьим и город В – четвертым. Для такого представления требуется  $n^2$  нейронов – число, которое быстро растет с увеличением числа городов. Длина такого маршрута была бы равна  $d_{ca} + d_{ad} + d_{db} + d_{bc}$ . Так как каждый город посещается только один раз и в каждый момент посещается лишь один город, то в каждой строке и в каждом столбце имеется по одной единице. Для задачи с  $n$  городами всего имеется  $n!$  различных маршрутов обхода. Если  $n = 60$ , то имеется  $6934155 \times 10^{78}$  возможных маршрутов. Если принять во внимание, что в нашей галактике (Млечном Пути) имеется лишь  $10^{11}$  звезд, то станет ясным, что полный перебор всех возможных маршрутов для 1000 городов даже на самом быстром в мире компьютере займет время, сравнимое с геологической эпохой.

Продемонстрируем теперь, как сконструировать сеть для решения этой NP-полной проблемы. Каждый нейрон снабжен двумя индексами, которые соответствуют городу и порядковому номеру его посещения в маршруте. Например,  $OUT_{xj} = 1$  показывает, что город  $x$  был  $j$ -ым по порядку городом маршрута.

Функция энергии должна удовлетворять двум требованиям: во-первых, должна быть малой только для тех решений, которые имеют по одной единице в каждой строке и в каждом столбце; во-вторых, должна оказывать предпочтение решениям с короткой длиной маршрута.

Первое требование удовлетворится введением следующей, состоящей из трех сумм, функции энергии:

$$E = \frac{A}{2} \sum_x \sum_i \sum_{j \neq i} OUT_{xi} OUT_{xj} + \frac{B}{2} \sum_i \sum_x \sum_{y \neq x} OUT_{xi} OUT_{yi} + \frac{C}{2} \left[ \left( \sum_x \sum_i OUT_{xi} \right) - n \right]^2, \quad (6.9)$$

где А, В и С – некоторые константы. Этим достигается выполнение следующих условий:

1. Первая тройная сумма равна нулю в том и только в том случае, если каждая строка (город) содержит не более одной единицы.
2. Вторая тройная сумма равна нулю в том и только в том случае, если каждый столбец (порядковый номер посещения) содержит не более одной единицы.
3. Третья сумма равна нулю в том и только в том случае, если матрица содержит ровно  $n$  единиц.

город	Порядок следования			
	1	2	3	4
А	0	1	0	0
В	0	0	0	1
С	1	0	0	0
Д	0	0	1	0

Рис. 6.6. Маршрут коммивояжера

Второе требование – предпочтение коротким маршрутам – удовлетворяется с помощью добавления следующего члена к функции энергии:

$$E = \frac{D}{2} \sum_x \sum_{y \neq x} \sum_i d_{xy} OUT_{xi} (OUT_{y,i+1} + OUT_{y,i-1}), \quad (6.10)$$

Заметим, что этот член представляет собой длину любого допустимого маршрута. Для удобства индексы определяются по модулю  $n$ , т. е.  $OUT_{n+j} = OUT_j$ , а  $D$  – некоторая константа.

При достаточно больших значениях А, В и С низкоэнергетические состояния будут представлять допустимые маршруты, а большие значения D гарантируют, что будет найден короткий маршрут.

Теперь зададим значения весов, т. е. установим соответствие между членами в функции энергии и членами общей формы (см. уравнение 6.2)).

Получаем

$$w_{x_i, y_j} = \begin{aligned} & -A\delta_{xy}(1 - \delta_{ij}) && \text{(не допускает более одной единицы в строке)} \\ & -B\delta_{ij}(1 - \delta_{xy}) && \text{(не допускает более одной единицы в столбце)} \\ & -C && \text{(глобальное ограничение)} \\ & -Dd_{xy}(\delta_{j,i+1} + \delta_{j,i-1}) && \text{(член, отвечающий за длину цикла),} \end{aligned}$$

где  $\delta_{ij} = 1$ , если  $i = j$ , в противном случае  $\delta_{ij} = 0$ . Кроме того, каждый нейрон имеет смещающий вес  $x_i$ , соединенный с +1 и равный  $Cn$ .

В работе [8] сообщается об эксперименте, в котором задача коммивояжера была решена для 10 городов. В этом случае возбуждающая функция была равна

$$\text{OUT} = \frac{1}{2} [1 + \text{th}(\text{NET}/U_0)].$$

Как показали результаты, 16 из 20 прогонов сошлись к допустимому маршруту и около 50% решений оказались кратчайшими маршрутами, как это было установлено с помощью полного перебора. Этот результат станет более впечатляющим, если осознать, что имеется 181440 допустимых маршрутов.

Сообщалось, что сходимость решений, полученных по методу Хопфилда для задачи коммивояжера, в сильной степени зависит от коэффициентов, и не имеется систематического метода определения их значений [11]. В этой работе предложена другая функция энергии с единственным коэффициентом, значение которого легко определяется. В дополнение предложен новый сходящийся алгоритм. Можно ожидать, что новые более совершенные методы будут разрабатываться, так как полностью удовлетворительное решение нашло бы массу применений.

## ОБСУЖДЕНИЕ

### Локальные минимумы

Сеть, выполняющая аналого-цифровое преобразование, всегда находит единственное оптимальное решение. Это обусловлено простой природой поверхности энергии в этой задаче. В задаче коммивояжера поверхность энергии сильно изрезана, изобилует склонами, долинами и локальными минимумами и нет гарантии, что будет найдено глобальное оптимальное решение и что полученное решение будет допустимым. При этом возникают серьезные вопросы относительно надежности сети и доверия к ее решениям. Эти недостатки сети смягчаются тем обстоятельством, что нахождение глобальных минимумов для NP-полных задач является очень трудной задачей, которая не может быть решена в приемлемое время никаким другим методом. Другие методы значительно более медленны и дают не лучшие результаты.

### Скорость

Способность сети быстро производить вычисления является ее главным достоинством. Она обусловлена высокой степенью распараллеливания вычислительного процесса. Если сеть реализована на аналоговой электронике, то решение редко занимает промежутки времени, больший нескольких постоянных времени сети. Более того, время сходимости слабо зависит от размерности задачи. Это резко контрастирует с более чем экспоненциальным ростом времени решения при использовании обычных подходов. Моделирование с помощью однопроцессорных систем не позволяет использовать преимущества параллельной архитектуры, но современные мультипроцессорные системы типа Connection Machine (65536 процессоров!) весьма многообещающи для решения трудных задач.

### Функция энергии

Определение функции энергии сети в зависимости от задачи не является тривиальным. Существующие решения были получены с помощью изобретательности, математического опыта и таланта, которые не разбросаны в изобилии. Для некоторых задач существуют систематические методы нахождения весов сети. Эти методы излагаются в гл. 7.

## Емкость сети

Актуальным предметом исследований является максимальное количество запоминаемой информации, которое может храниться в сети Хопфилда. Так как сеть из  $n$  двоичных нейронов может иметь  $2^n$  состояний, то исследователи были удивлены, обнаружив, что максимальная емкость памяти оказалась значительно меньшей.

Если бы могло запоминаться большое количество информационных единиц, то сеть не стабилизировалась бы на некоторых из них. Более того, она могла бы помнить то, чему ее не учили, т. е. могла стабилизироваться на решении, не являющемся требуемым вектором. Эти свойства ставили в тупик первых исследователей, которые не имели математических методов для предварительной оценки емкости памяти сети.

Последние исследования пролили свет на эту проблему. Например, предполагалось, что максимальное количество запоминаемой информации, которое может храниться в сети из  $N$  нейронов и безошибочно извлекаться, меньше чем  $cN^2$ , где  $c$  – положительная константа, большая единицы. Хотя этот предел и достигается в некоторых случаях, в общем случае он оказался слишком оптимистическим. В работе [4] было экспериментально показано, что в общем случае предельное значение емкости ближе к  $0,15N$ . В работе [1] было показано, что число таких состояний не может превышать  $N$ , что согласуется с наблюдениями над реальными системами и является наилучшей на сегодняшний день оценкой.

## ВЫВОДЫ

Сети с обратными связями являются перспективным объектом для дальнейших исследований. Их динамическое поведение открывает новые интересные возможности и ставит специфические проблемы. Как отмечается в гл. 9, эти возможности и проблемы сохраняются при реализации нейронных сетей в виде оптических систем.

## Литература

1. Abu-Mostafa Y. S., St. Jacques, J. 1985. Information capacity of the Hopfield model. *IEEE Transactions on Information Theory* 31(4):461-64.
2. Cohen M. A., Grossberg S. G. 1983. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man and Cybernetics* 13:815-26.
3. Qarey M. R., Johnson D. S. 1979. *Computers and intrac-tality*. New York: W.H. Freeman.
4. Grossberg S. 1987. *The adapptive brain*, vol. 1 and 2. Amsterdam: North-Holland.
5. Hinton G. E., Sejnowski T. J. 1986. Learning and relearning in Boltzmann machines. In *Parallel distributed processing*, vol. 1, pp. 282-317. Cambridge, MA: MIT Press.
6. Horfield J. J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Science* 79:2554-58.
7. Horfield J. J. 1984. Neural with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Science* 81:3088-92.
8. Horfield J. J., Tank D. W. 1985. Neural computation of decisions in optimization problems. *Biological Cybernetics* 52:141-52.
9. Horfield J. J., Tank D. W. 1986. Computing with neural circuits: A model. *Science* 233:625-33.
10. Tank D. W., Horfield J. J. 1986. Simple «neural» optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit. *Circuits and Systems IEEE Transactions on CAS-33(5):533-41*.
11. Van den Bout D. E. and Miller T. K. 1988. A traveling salesman objective function that works. *Proceedings of the IEEE International Conference on Neural Networks*, vol. 2, pp. 299-304. San Diego, CA: SOS Printing.

## Глава 7.

### Двунаправленная ассоциативная память

Память человека часто является ассоциативной; один предмет напоминает нам о другом, а этот другой о третьем. Если позволить нашим мыслям, они будут перемещаться от предмета к предмету по цепочке умственных ассоциаций. Кроме того, возможно использование способности к ассоциациям для восстановления забытых образов. Если мы забыли, где оставили свои очки, то пытаемся вспомнить, где видели их, в последний раз, с кем разговаривали и что делали. Посредством этого устанавливается конец цепочки ассоциаций, что позволяет нашей памяти соединять ассоциации для получения требуемого образа.

Ассоциативная память, рассмотренная в гл. 6, является, строго говоря, автоассоциативной, это означает, что образ может быть завершен или исправлен, но не может быть ассоциирован с другим образом. Данный факт является результатом одноуровневой структуры ассоциативной памяти, в которой вектор появляется на выходе тех же нейронов, на которые поступает входной вектор.

Двунаправленная ассоциативная память (ДАП) является гетероассоциативной; входной вектор поступает на один набор нейронов, а соответствующий выходной вектор вырабатывается на другом наборе нейронов. Как и сеть Хопфилда, ДАП способна к обобщению, вырабатывая правильные реакции, несмотря на искаженные входы. Кроме того, могут быть реализованы адаптивные версии ДАП, выделяющие эталонный образ из зашумленных экземпляров. Эти возможности сильно напоминают процесс мышления человека и позволяют искусственным нейронным сетям сделать шаг в направлении моделирования мозга.

В последних публикациях [9,12] представлено несколько форм реализации двунаправленной ассоциативной памяти. Как большинство важных идей, изложенные в этих работах идеи имеют глубокие корни; например, в работе Гроссберга [6] представлены некоторые важные для ДАП концепции. В данной работе ссылки приводятся не с целью разрешения вопроса о приоритете исследовательских работ, а исключительно для освещения их вклада в исследовательскую тематику.

#### СТРУКТУРА ДАП

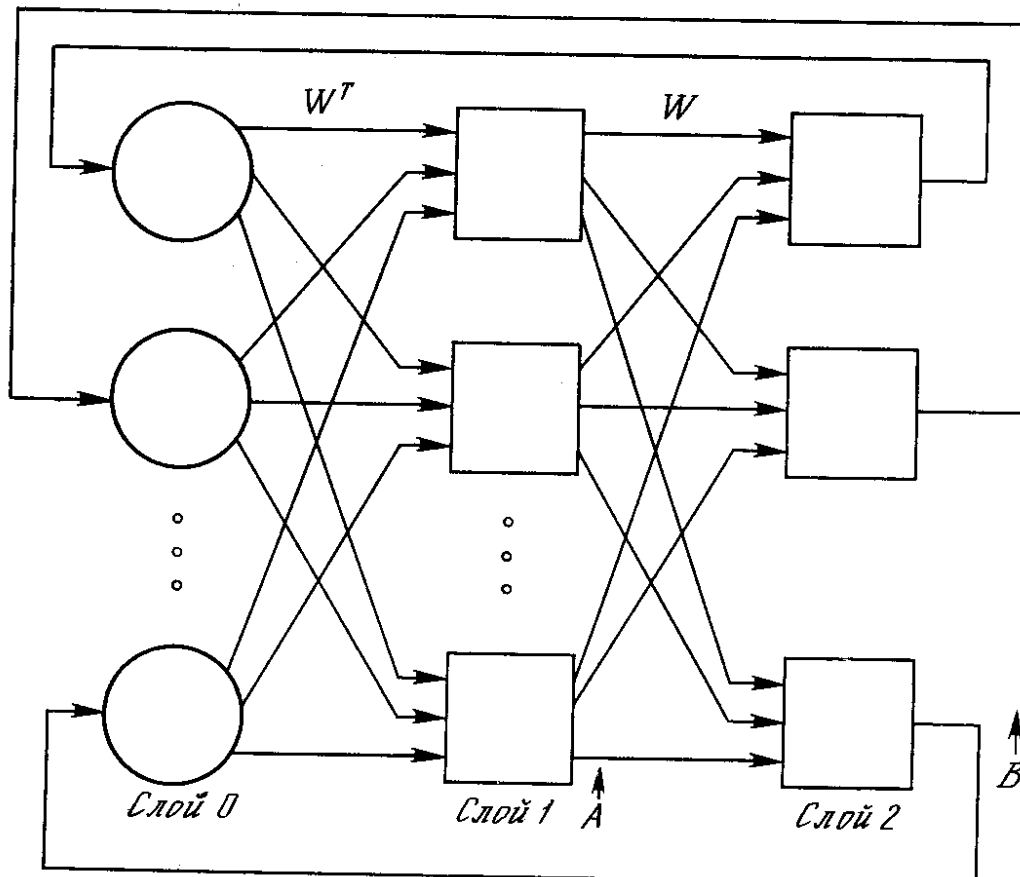


Рис. 7.1. Конфигурация двунаправленной ассоциативной памяти



На рис. 7.1 приведена базовая конфигурация ДАП. Эта конфигурация существенно отличается от используемой в работе [9]. Она выбрана таким образом, чтобы подчеркнуть сходство с сетями Хопфилда и предусмотреть увеличения количества слоев. На рис. 7.1 входной вектор  $\mathbf{A}$  обрабатывается матрицей весов  $\mathbf{W}$  сети, в результате чего вырабатывается вектор выходных сигналов нейронов  $\mathbf{B}$ . Вектор  $\mathbf{B}$  затем обрабатывается транспонированной матрицей  $\mathbf{W}^t$  весов сети, которая вырабатывает новые выходные сигналы, представляющие собой новый входной вектор  $\mathbf{A}$ . Этот процесс повторяется до тех пор, пока сеть не достигнет стабильного состояния, в котором ни вектор  $\mathbf{A}$ , ни вектор  $\mathbf{B}$  не изменяются. Заметим, что нейроны в слоях 1 и 2 функционируют, как и в других парадигмах, вычисляя сумму взвешенных входов и вычисляя по ней значение функции активации  $F$ . Этот процесс может быть выражен следующим образом:

$$b_i = F\left(\sum_j a_j w_{ij}\right) \quad (7.1)$$

или в векторной форме:

$$\mathbf{B} = F(\mathbf{AW}), \quad (7.2)$$

где  $\mathbf{B}$  – вектор выходных сигналов нейронов слоя 2,  $\mathbf{A}$  – вектор выходных сигналов нейронов слоя 1,  $\mathbf{W}$  – матрица весов связей между слоями 1 и 2,  $F$  – функция активации.

Аналогично

$$\mathbf{A} = F(\mathbf{BW}^t) \quad (7.3)$$

где  $\mathbf{W}^t$  является транспозицией матрицы  $\mathbf{W}$ .

Как отмечено в гл. 1, Гроссберг показал преимущества использования сигмоидальной (логистической) функции активации

$$\text{OUT}_i = \frac{1}{1 + \exp(-\lambda \text{NET}_i)} \quad (7.3)$$

где  $\text{OUT}_i$  – выход нейрона  $i$ ,  $\text{NET}_i$  – взвешенная сумма входных сигналов нейрона  $i$ ,  $\lambda$  – константа, определяющая степень кривизны.

В простейших версиях ДАП значение константы  $\lambda$  выбирается большим, в результате чего функция активации приближается к простой пороговой функции. В дальнейших рассуждениях будем предполагать, что используется пороговая функция активации.

Примем также, что существует память внутри каждого нейрона в слоях 1 и 2 и что выходные сигналы нейронов изменяются одновременно с каждым тактом синхронизации, оставаясь постоянными между этими тактами. Таким образом, поведение нейронов может быть описано следующими правилами:

$$\text{OUT}_i(n+1) = 1, \text{ если } \text{NET}_i(n) > 0,$$

$$\text{OUT}_i(n+1) = 0, \text{ если } \text{NET}_i(n) < 0,$$

$$\text{OUT}_i(n+1) = \text{OUT}_i(n), \text{ если } \text{NET}_i(n) = 0,$$

где  $\text{OUT}_i(n)$  представляет собой величину выходного сигнала нейрона  $i$  в момент времени  $n$ .

Заметим, что как и в описанных ранее сетях слой 0 не производит вычислений и не имеет памяти; он является только средством распределения выходных сигналов слоя 2 к элементам матрицы  $\mathbf{W}^t$ .

## ВОССТАНОВЛЕНИЕ ЗАПОМНЕННЫХ АССОЦИАЦИЙ

Долговременная память (или ассоциации) реализуется в весовых массивах  $\mathbf{W}$  и  $\mathbf{W}^t$ . Каждый образ состоит из двух векторов: вектора  $\mathbf{A}$ , являющегося выходом слоя 1, и вектора  $\mathbf{B}$ , ассоциированного образа, являющегося выходом слоя 2. Для восстановления ассоциированного образа вектор  $\mathbf{A}$  или его часть кратковременно устанавливаются на выходах слоя 1. Затем вектор  $\mathbf{A}$  удаляется и сеть приводится в стабильное состояние, вырабатывая ассоциированный вектор  $\mathbf{B}$  на выходе слоя 2. Затем вектор  $\mathbf{B}$  воздействует через транспонированную матрицу  $\mathbf{W}^t$ , воспроизводя воздействие исходного входного вектора  $\mathbf{A}$  на выходе слоя 1. Каждый такой цикл вызывает уточнение выходных векторов слоя 1 и 2 до тех пор, пока не будет достигнута точка стабильности в сети. Эта точка может быть рассмотрена как резонансная, так как вектор передается обратно и вперед между слоями сети, всегда

обрабатывая текущие выходные сигналы, но больше не изменяя их. Состояние нейронов представляет собой кратковременную память (КП), так как оно может быстро изменяться при появлении другого входного вектора. Значения коэффициентов весовой матрицы образуют долговременную память и могут изменяться только на более длительном отрезке времени, используя представленные ниже в данном разделе методы.

В работе [9] показано, что сеть функционирует в направлении минимизации функции энергии Ляпунова в основном таким же образом, как и сети Хопфилда в процессе сходимости (см. гл. 6). Таким образом, каждый цикл модифицирует систему в направлении энергетического минимума, расположение которого определяется значениями весов.

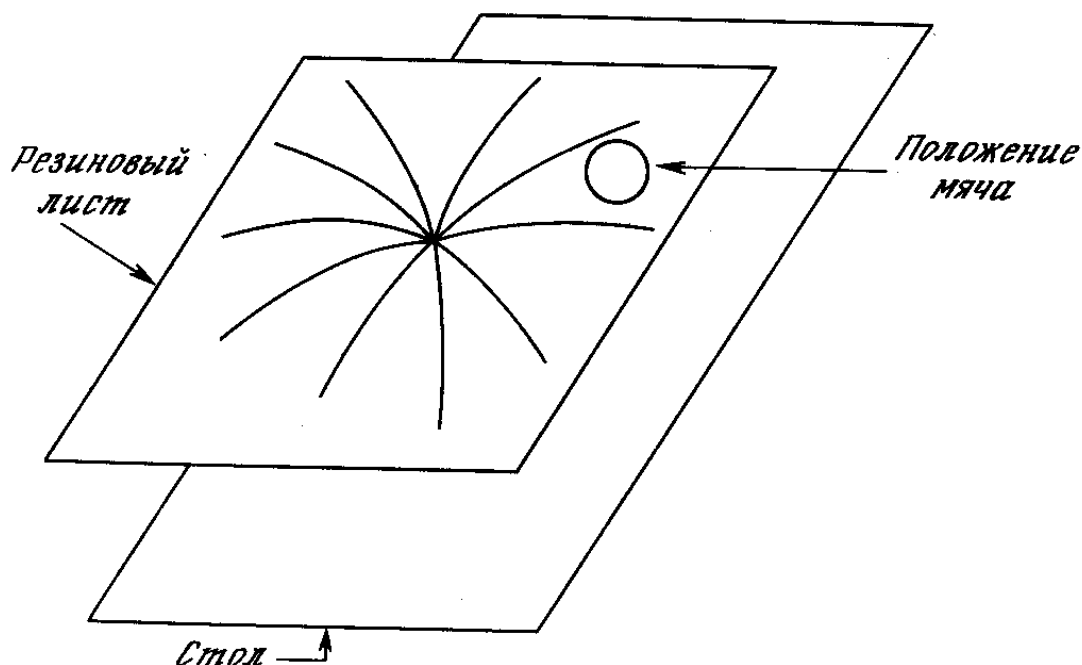


Рис. 7.2. Энергетическая поверхность двунаправленной ассоциативной памяти

Этот процесс может быть визуально представлен в форме направленного движения мяча по резиновой ленте, вытянутой над столом, причем каждому запомненному образу соответствует точка, «вдавленная» в направлении поверхности стола. Рис. 7.2 иллюстрирует данную аналогию с одним запомненным образом. Данный процесс формирует минимум гравитационной энергии в каждой точке, соответствующей запомненному образу, с соответствующим искривлением поля притяжения в направлении к данной точке. Свободно движущийся мяч попадает в поле притяжения и в результате будет двигаться в направлении энергетического минимума, где и остановится.

## КОДИРОВАНИЕ АССОЦИАЦИЙ

Обычно сеть обучается распознаванию множества образов. Обучение производится с использованием обучающего набора, состоящего из пар векторов  $\mathbf{A}$  и  $\mathbf{B}$ . Процесс обучения реализуется в форме вычислений; это означает, что весовая матрица вычисляется как сумма произведения всех векторных пар обучающего набора. В символьной форме

$$\mathbf{W} = \sum_i \mathbf{A}_i^t \mathbf{B}_i$$

Предположим, что все запомненные образы представляют собой двоичные векторы. Это ограничение покажется менее строгим, если вспомнить, что все содержимое Библиотеки Конгресса может быть закодировано в один очень длинный двоичный вектор. В работе [11] показана возможность достижения более высокой производительности при использовании биполярных векторов. При этом векторная компонента, большая чем 0, становится +1, а компонента, меньшая или равная 0, становится -1.

Предположим, что требуется обучить сеть с целью запоминания трех пар двоичных векторов, причем векторы  $\mathbf{A}_i$  имеют размерность такую же, как и векторы  $\mathbf{B}_i$ . Надо отметить, что это не является необходимым условием для работы алгоритма; ассоциации могут быть сформированы и между векторами различной размерности.

Исходный вектор

Ассоциированный вектор

Бинарная версия

$$\begin{array}{llll}
\mathbf{A}_1 = (1,0,0) & \mathbf{B}_1 = (0,0,1) & \mathbf{A}'_1 = (1,-1,-1) & \mathbf{B}'_1 = (-1,-1,1) \\
\mathbf{A}_2 = (0,1,0) & \mathbf{B}_2 = (0,1,0) & \mathbf{A}'_2 = (-1,1,-1) & \mathbf{B}'_2 = (-1,1,-1) \\
\mathbf{A}_3 = (0,0,1) & \mathbf{B}_3 = (1,0,0) & \mathbf{A}'_3 = (-1,-1,1) & \mathbf{B}'_3 = (1,-1,-1)
\end{array}$$

Вычисляем весовую матрицу

$$\mathbf{W} = \mathbf{A}'_1{}^t \mathbf{B}'_1 + \mathbf{A}'_2{}^t \mathbf{B}'_2 + \mathbf{A}'_3{}^t \mathbf{B}'_3$$

$$\begin{array}{cccccccccccccccc}
-1 & -1 & 1 & & 1 & -1 & 1 & & -1 & 1 & 1 & & -1 & -1 & 3 \\
1 & 1 & -1 & + & -1 & 1 & -1 & + & -1 & -1 & 1 & = & -1 & 3 & -1 \\
1 & 1 & -1 & & 1 & -1 & 1 & & 1 & -1 & -1 & & 3 & -1 & -1
\end{array}$$

Далее прикладывая входной вектор  $\mathbf{A} = (1,0,0)$ , вычисляем выходной вектор  $\mathbf{O}$

$$\mathbf{O} = \mathbf{A}_1{}^t \mathbf{W} = (1,0,0) \times \begin{array}{ccc} 1 & -1 & 3 \\ -1 & 3 & -1 \\ 3 & -1 & -1 \end{array} = (-1,-1,3)$$

Используя пороговое правило

$$\begin{array}{l}
b_i = 1, \text{ если } o_i > 0, \\
b_i = 0, \text{ если } o_i < 0, \\
b_i = 0, \text{ не изменяется, если } o_i = 0
\end{array}$$

вычисляем

$$\mathbf{B}'_1 = (0,0,1),$$

что является требуемой ассоциацией. Затем, подавая вектор  $\mathbf{B}'_1$  через обратную связь на вход первого слоя к  $\mathbf{W}^t$  получаем

$$\mathbf{O} = \mathbf{B}'_1{}^t \mathbf{W}^t = (0,0,1) \times \begin{array}{ccc} 1 & -1 & 3 \\ -1 & 3 & -1 \\ 3 & -1 & -1 \end{array} = (3,-1,-1)$$

что дает значение  $(1,0,0)$  после применения пороговой функции, образуя величину вектора  $\mathbf{A}_1$ .

Этот пример показывает, как входной вектор  $\mathbf{A}$  с использованием матрицы  $\mathbf{W}$  производит выходной вектор  $\mathbf{B}$ . В свою очередь вектор  $\mathbf{B}$  с использованием матрицы  $\mathbf{W}^t$  производит вектор  $\mathbf{A}$ , таким образом в системе формируется устойчивое состояние и резонанс.

ДАП обладает способностью к обобщению. Например, если незавершенный или частично искаженный вектор подается в качестве  $\mathbf{A}$ , сеть имеет тенденцию к выработке запомненного вектора  $\mathbf{B}$ , который в свою очередь стремится исправить ошибки в  $\mathbf{A}$ . Возможно, для этого потребуется несколько проходов, но сеть сходится к воспроизведению ближайшего запомненного образа.

Системы с обратной связью могут иметь тенденцию к колебаниям; это означает, что они могут переходить от состояния к состоянию, никогда не достигая стабильности. В [9] доказано, что все ДАП безусловно стабильны при любых значениях весов сети. Это важное свойство возникает из отношения транспонирования между двумя весовыми матрицами и означает, что любой набор ассоциаций может быть изучен без риска возникновения нестабильности.

Существует взаимосвязь между ДАП и рассмотренными в гл. 6 сетями Хопфилда. Если весовая матрица  $\mathbf{W}$  является квадратной и симметричной, то  $\mathbf{W} = \mathbf{W}^t$ . В этом случае, если слои 1 и 2 являются одним и тем же набором нейронов, ДАП превращается в автоассоциативную сеть Хопфилда.

## ЕМКОСТЬ ПАМЯТИ

Как и сети Хопфилда, ДАП имеет ограничения на максимальное количество ассоциаций, которые она может точно воспроизвести. Если этот лимит превышен, сеть может выработать неверный выходной сигнал, воспроизводя ассоциации, которым не обучена.

В работе [9] приведены оценки, в соответствии с которыми количество запомненных ассоциаций не может превышать количества нейронов в меньшем слое. При этом предполагается, что емкость памяти максимизирована посредством специального кодирования, при котором количество компонент со значениями +1 равно количеству компонент со значениями -1 в каждом биполярном векторе. Эта оценка оказалась слишком оптимистичной. Работа [13] по оценке емкости сетей Хопфилда может быть легко расширена для ДАП. Можно показать, что если  $L$  векторов выбраны случайно и представлены в указанной выше форме, и если  $L$  меньше чем  $n/(2 \log_2 n)$ , где  $n$  – количество нейронов в наименьшем слое, тогда все запомненные образы, за исключением «малой части», могут быть восстановлены. Например, если  $n = 1024$ , тогда  $L$  должно быть меньше 51. Если все образы должны восстанавливаться,  $L$  должно быть меньше  $ge/(4 \log_2 n)$ , то есть меньше 25. Эти, скорее озадачивающие, результаты показывают, что большие системы могут запоминать только умеренное количество ассоциаций.

В работе [7] показано, что ДАП может иметь до  $2^n$  стабильных состояний, если пороговое значение  $T$  выбирается для каждого нейрона. Такая конфигурация, которую авторы назвали негомогенной ДАП, является расширением исходной гомогенной ДАП, в которой все пороги были нулевыми. Модифицированная передаточная функция нейрона принимает в этом случае следующий вид:

$$\begin{aligned} \text{OUT}_i(n+1) &= 1, \text{ если } \text{NET}_i(n) > T_i, \\ \text{OUT}_i(n+1) &= -1, \text{ если } \text{NET}_i(n) < T_i, \\ \text{OUT}_i(n+1) &= \text{OUT}_i(n), \text{ если } \text{NET}_i(n) = T_i, \end{aligned}$$

где  $\text{OUT}_i(t)$  – выход нейрона  $i$  в момент времени  $t$ .

Посредством выбора соответствующего порога для каждого нейрона количество стабильных состояний может быть сделано любым в диапазоне от 1 до  $2^n$ , где  $n$  есть количество нейронов в меньшем слое. К сожалению, эти состояния не могут быть выбраны случайно; они определяются жесткой геометрической процедурой. Если пользователь выбирает  $L$  состояний случайным образом, причем  $L$  меньше  $(0,68)n^2/(\lceil \log_2(n) \rceil + 4)^2$ , и если каждый вектор имеет  $4 + \log_2 n$  компонент, равных +1, и остальные, равные -1, то можно сконструировать негомогенную ДАП, имеющую 98% этих векторов в качестве стабильных состояний. Например, если  $n = 1024$ ,  $L$  должно быть меньше 3637, что является существенным улучшением по сравнению с гомогенными ДАП, но это намного меньше  $2^{1024}$  возможных состояний.

Ограничение количества единиц во входных векторах представляет серьезную проблему, тем более, что теория, которая позволяет перекодировать произвольный набор векторов в такой «разреженный» набор, отсутствует. Возможно, однако, что еще более серьезной является проблема некорректной сходимости. Суть этой проблемы заключается в том, что сеть может не производить точных ассоциаций вследствие природы поля притяжения; об ее форме известно очень немногое. Это означает, что ДАП не является ассоциатором по отношению к ближайшему соседнему образу. В действительности она может производить ассоциации, имеющие слабое отношение ко входному вектору. Как и в случае гомогенных ДАП, могут встречаться ложные стабильные состояния и немного известно об их количестве и природе.

Несмотря на эти проблемы, ДАП остается объектом интенсивных исследований. Основная привлекательность ДАП заключается в ее простоте. Кроме того, она может быть реализована в виде СБИС (либо аналоговых, либо цифровых), что делает ее потенциально недорогой. Так как наши знания постоянно растут, ограничения ДАП могут быть сняты. В этом случае как в экспериментальных, так и в практических приложениях ДАП будет являться весьма перспективным и полезным классом искусственных нейронных сетей.

## НЕПРЕРЫВНАЯ ДАП

В предшествующем обсуждении нейроны в слоях 1 и 2 рассматривались как синхронные, каждый нейрон обладает памятью, причем все нейроны изменяют состояния одновременно под воздействием импульса от центральных часов. В асинхронной системе любой нейрон свободен изменять состояние в любое время, когда его вход предписывает это сделать.

Кроме того, при определении функции активации нейрона использовался простой порог, тем самым образуя разрывность передаточной функции нейронов. Как синхронность функционирования, так и разрывность функций, являются биологически неправдоподобными и совсем необязательными; непрерывные асинхронные ДАП отвергают синхронность и разрывность, но функционируют в основном аналогично дискретным версиям. Может показаться, что такие системы должны являться нестабильными. В [9] показано, что непрерывные ДАП являются стабильными (однако для них справедливы ограничения емкости, обсужденные ранее).

В работах [2-5] показано, что сигмоида является оптимальной функцией активации благодаря ее способности усиливать низкоуровневые сигналы, в то же время сжимая динамический диапазон нейронов. Непрерывная ДАП может иметь

сигмоидальную функцию с величиной  $\lambda$ , близкой к единице, образуя тем самым нейроны с плавной и непрерывной реакцией, во многом аналогичной реакции их биологических прототипов.

Непрерывная ДАП может быть реализована в виде аналоговой схемы из резисторов и усилителей. Реализация таких схем в виде СБИС кажется возможной и экономически привлекательной. Еще более обещающей является оптическая реализация, рассматриваемая в гл. 9.

## АДАПТИВНАЯ ДАП

В версиях ДАП, рассматриваемых до сих пор, весовая матрица вычисляется в виде суммы произведений пар векторов. Эти вычисления полезны, поскольку они демонстрируют функции, которые может выполнять ДАП. Однако это определенно не тот способ, посредством которого производится определение весов нейронов мозга.

Адаптивная ДАП изменяет свои веса в процессе функционирования. Это означает, что подача на вход сети обучающего набора входных векторов заставляет ее изменять энергетическое состояние до получения резонанса. Постепенно кратковременная память превращается в долговременную память, настраивая сеть в результате ее функционирования. В процессе обучения векторы подаются на слой А, а ассоциированные векторы на слой В. Один из них или оба вектора могут быть зашумленными версиями эталона; сеть обучается исходным векторам, свободным от шума. В этом случае она извлекает сущность ассоциаций, обучаясь эталонам, хотя «видела» только зашумленные аппроксимации.

Так как доказано, что непрерывная ДАП является стабильной независимо от значения весов, ожидается, что медленное изменение ее весов не должно нарушить этой стабильности. В работе [10] доказано это правило.

Простейший обучающий алгоритм использует правило Хэбба [8], в котором изменение веса пропорционально уровню активации его нейрона-источника и уровню активации нейрона-приемника. Символически это можно представить следующим образом:

$$\delta w_{ij} = \eta^*(OUT_i - OUT_j), \quad (7.5)$$

где  $\delta w_{ij}$  – изменение веса связи нейрона  $i$  с нейроном  $j$  в матрицах  $\mathbf{W}$  или  $\mathbf{W}^t$ ,  $OUT_i$  – выход нейрона  $i$  слоя 1 или 2;  $\eta$  – положительный нормирующий коэффициент обучения, меньший 1.

## КОНКУРИРУЮЩАЯ ДАП

Во многих конкурирующих нейронных системах наблюдаются некоторые виды конкуренции между нейронами. В нейронах, обрабатывающих сигналы от сетчатки, латеральное торможение приводит к увеличению выхода наиболее высокоактивных нейронов за счет соседних. Такие системы увеличивают контрастность, поднимая уровень активности нейронов, подсоединенных к яркой области сетчатки, в то же время еще более ослабляя выходы нейронов, подсоединенных к темным областям.

В ДАП конкуренция реализуется взаимным соединением нейронов внутри каждого слоя посредством дополнительных связей. Веса этих связей формируют другую весовую матрицу с положительными значениями элементов главной диагонали и отрицательными значениями остальных элементов. Теорема Кохен-Гроссберга [1] показывает, что такая сеть является безусловно стабильной, если весовые матрицы симметричны. На практике сети обычно стабильны даже в случае отсутствия симметрии весовых матриц. Однако неизвестно, какие особенности весовых матриц могут привести к неустойчивости функционирования сети.

## ЗАКЛЮЧЕНИЕ

Ограниченная емкость памяти ДАП, ложные ответы и некоторая непредсказуемость поведения привели к рассмотрению ее как устаревшей модели искусственных нейронных сетей.

Этот вывод определенно является преждевременным. ДАП имеет много преимуществ: она совместима с аналоговыми схемами и оптическими системами; для нее быстро сходятся как процесс обучения так, и процесс восстановления информации; она имеет простую и интуитивно привлекательную форму функционирования. В связи с быстрым развитием теории могут быть найдены методы, объясняющие поведение ДАП и разрешающие ее проблемы.

## Литература

12. Cohen M., Grossberg S. 1983. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. IEE Transactions on Systems, Man, and Cybernetics SMC-13:815-926.

13. Grossberg S. 1973. Contour enhancement, short term memory, and constancies in reverberating neural networks. *Studies in Applied Mathematics* 52:217-57.
14. Grossberg S. 1976. Adaptive pattern classification and universal recording, 1: Parallel development and coding of neural feature detectors. *Biological Cybernetics* 23:187-202.
15. Grossberg S. 1978. A theory of human memory: Self-organization and performance of sensory-motor codes, maps, and plans. In *Progress in theoretical biology*, vol. 5, ed. R. Rosen and F. Shell. New York: Academic Press.
16. Grossberg S. 1980. How does the brain build a cognitive code? *Psychological Review* 1:1-51.
17. Grossberg S. 1982. *Studies of mind and brain*. Boston: Reidel Press.
18. Haines K., Hecht-Nielsen R. 1988. A BAM with increased information storage capacity. *Proceedings of the IEEE International Conference on Neural Networks*, vol. 1, pp. 181-190. San Diego, CA: SOS Printing.
19. Hebb D. O. 1949. *The organization of behavior*. New York: Wiley.
20. Kosko B. (1987a). Bi-directional associative memories. *IEEE Transactions on Systems, Man and Cybernetics* 18(1):49-60.
21. Kosko B. (1987b). Competitive adaptive bi-directional associative memories. In *Proceedings of the IEEE First International Conference on Neural Networks*, eds. M. Caudill and C. Butler, vol. 2, pp. 759-66. San Diego, CA: SOS Printing.
22. Kosko B. (1987c). Constructing an associative memory. *Byte*, September, pp. 137-44.
23. Kosko B., Guest C. 1987. Optical bi-directional associative memories. *Society for Photo-optical and Instrumentation Engineers Proceedings: Image Understanding* 758:11-18.
24. McEliece R. J., Rosner E. G., Rodemich E. R., Venkatesh S. S. 1987. The capacity of Hopfield associative memory. *IEEE Transactions on Information Theory* IT-33:461-82.

## Глава 8.

### Адаптивная резонансная теория

Мозг человека выполняет трудную задачу обработки непрерывного потока сенсорной информации, получаемой из окружающего мира. Из потока тривиальной информации он должен выделить жизненно важную информацию, обработать ее и, возможно, зарегистрировать в долговременной памяти. Понимание процесса человеческой памяти представляет собой серьезную проблему; новые образы запоминаются в такой форме, что ранее запомненные не модифицируются и не забываются. Это создает дилемму: каким образом память остается пластичной, способной к восприятию новых образов, и в то же время сохраняет стабильность, гарантирующую, что образы не уничтожатся и не разрушатся в процессе функционирования?

Традиционные искусственные нейронные сети оказались не в состоянии решить проблему стабильности-пластичности. Очень часто обучение новому образу уничтожает или изменяет результаты предшествующего обучения. В некоторых случаях это не существенно. Если имеется только фиксированный набор обучающих векторов, они могут предъявляться при обучении циклически. В сетях с обратным распространением, например, обучающие векторы подаются на вход сети последовательно до тех пор, пока сеть не обучится всему входному набору. Если, однако, полностью обученная сеть должна запомнить новый обучающий вектор, он может изменить веса настолько, что потребуются полное переобучение сети.

В реальной ситуации сеть будет подвергаться постоянно изменяющимся воздействиям; она может никогда не увидеть один и тот же обучающий вектор дважды. При таких обстоятельствах сеть часто не будет обучаться; она будет непрерывно изменять свои веса, не достигая удовлетворительных результатов.

Более того, в работе [1] приведены примеры сети, в которой только четыре обучающих вектора, предъявляемых циклически, заставляют веса сети изменяться непрерывно, никогда не сходясь. Такая временная нестабильность явилась одним из главных факторов, заставивших Гроссберга и его сотрудников исследовать радикально отличные конфигурации. Адаптивная резонансная теория (АРТ) является одним из результатов исследования этой проблемы [2,4].

Сети и алгоритмы АРТ сохраняют пластичность, необходимую для изучения новых образов, в то же время предотвращая изменение ранее запомненных образов. Эта способность стимулировала большой интерес к АРТ, но многие исследователи нашли теорию трудной для понимания. Математическое описание АРТ является сложным, но основные идеи и принципы реализации достаточно просты для понимания. Мы сконцентрируемся далее на общем описании АРТ; математически более подготовленные читатели смогут найти изобилие теории в литературе, список которой приведен в конце главы. Нашей целью является обеспечение достаточно конкретной информацией, чтобы читатель мог понять основные идеи и возможности, а также провести компьютерное моделирование с целью исследования характеристик этого важного вида сетей.

### АРХИТЕКТУРА АРТ

Адаптивная резонансная теория включает две парадигмы, каждая из которых определяется формой входных данных и способом их обработки. АРТ-1 разработана для обработки двоичных входных векторов, в то время как АРТ-2, более позднее обобщение АРТ-1, может классифицировать как двоичные, так и непрерывные векторы. В данной работе рассматривается только

АРТ-1. Читателя, интересующегося АРТ-2, можно отослать к работе [3] для полного изучения этого важного направления. Для краткости АРТ-1 в дальнейшем будем обозначать как АРТ.

### Описание АРТ

Сеть АРТ представляет собой векторный классификатор. Входной вектор классифицируется в зависимости от того, на какой из множества ранее запомненных образов он похож. Свое классификационное решение сеть АРТ выражает в форме возбуждения одного из нейронов распознающего слоя. Если входной вектор не соответствует ни одному из запомненных образов, создается новая категория посредством запоминания образа, идентичного новому входному вектору. Если определено, что входной вектор похож на один из ранее запомненных векторов с точки зрения определенного критерия сходства, запомненный вектор будет изменяться (обучаться) под воздействием нового входного вектора таким образом, чтобы стать более похожим на этот входной вектор.

Запомненный образ не будет изменяться, если текущий входной вектор не окажется достаточно похожим на него. Таким образом решается дилемма стабильности-пластичности. Новый образ может создавать дополнительные классификационные категории, однако новый входной образ не может заставить измениться существующую память.

### Упрощенная архитектура АРТ

На рис. 8.1 показана упрощенная конфигурация сети АРТ, представленная в виде пяти функциональных модулей. Она включает два слоя нейронов, так называемых «слой сравнения» и «слой распознавания». Приемник 1, Приемник 2 и Сброс обеспечивают управляющие функции, необходимые для обучения и классификации.

Перед рассмотрением вопросов функционирования сети в целом необходимо рассмотреть отдельно функции модулей; далее обсуждаются функции каждого из них.

**Слой сравнения.** Слой сравнения получает двоичный входной вектор  $X$  и первоначально пропускает его неизменным для формирования выходного вектора  $C$ . На более поздней фазе в распознающем слое вырабатывается двоичный вектор  $R$ , модифицирующий вектор  $C$ , как описано ниже.

Каждый нейрон в слое сравнения (рис. 8.2) получает три двоичных входа (0 или 1): (1) компонента  $x_i$  входного вектора  $X$ ; (2) сигнал обратной связи  $R_i$  – взвешенная сумма выходов распознающего слоя; (3) вход от Приемника 1 (один и тот же сигнал подается на все нейроны этого слоя).

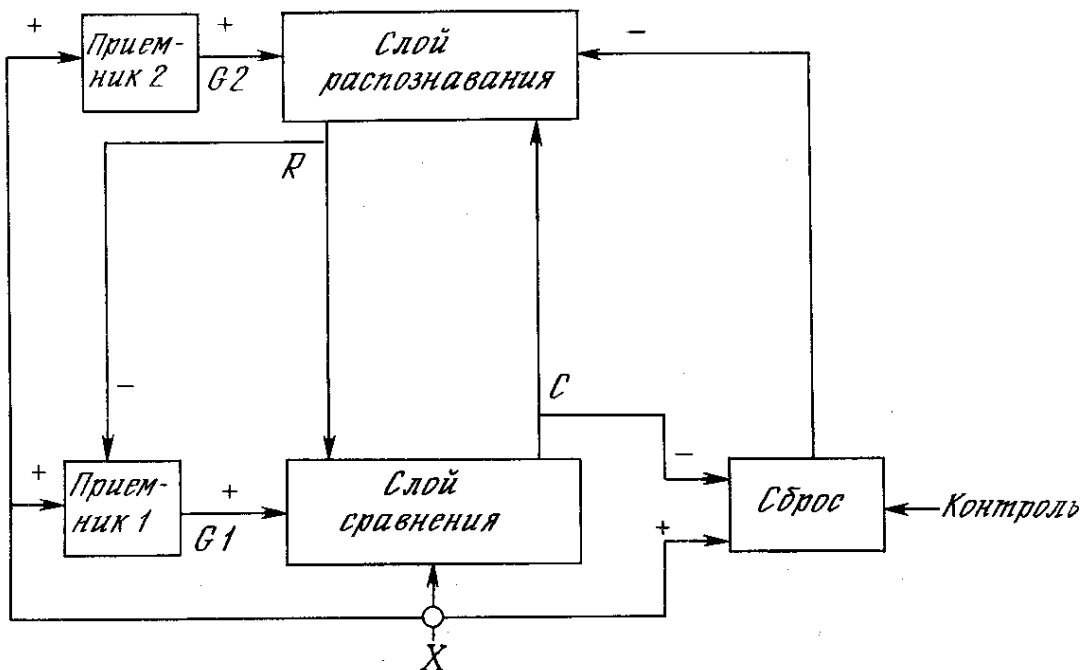


Рис. 8.1. Упрощенная сеть АРТ

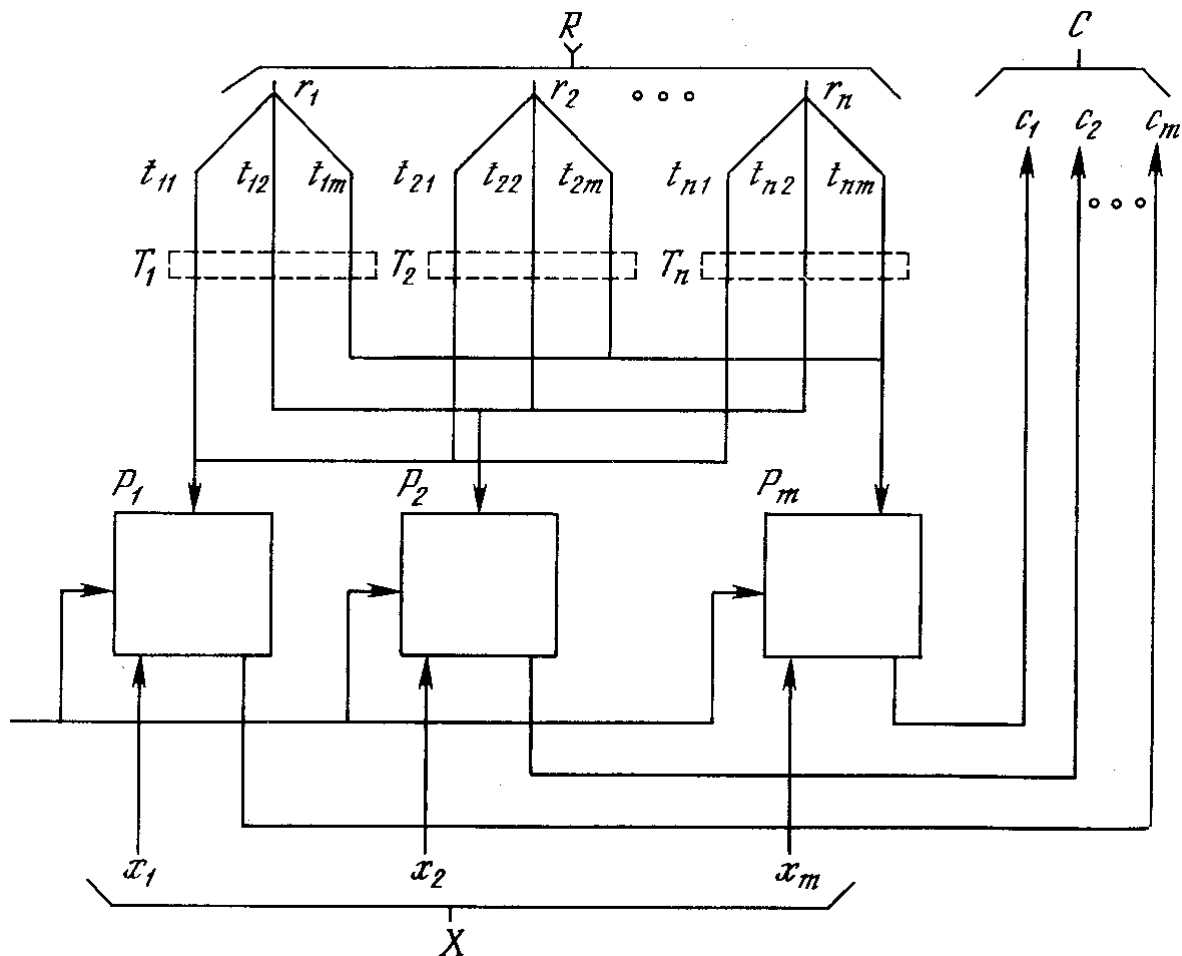


Рис. 8.2. Упрощенный слон сравнения

Чтобы получить на выходе нейрона единичное значение, как минимум два из трех его входов должны равняться единице; в противном случае его выход будет нулевым. Таким образом реализуется правило двух третей, описанное в [3]. Первоначально выходной сигнал G1 Приемника 1 установлен в единицу, обеспечивая один из необходимых для возбуждения нейронов входов, а все компоненты вектора  $\mathbf{R}$  установлены в 0; следовательно, в этот момент вектор  $\mathbf{C}$  идентичен двоичному входному вектору  $\mathbf{X}$ .

**Слой распознавания.** Слой распознавания осуществляет классификацию входных векторов. Каждый нейрон в слое распознавания имеет соответствующий вектор весов  $\mathbf{B}$ ; Только один нейрон с весовым вектором, наиболее соответствующим входному вектору, возбуждается; все остальные нейроны заторможены.

Как показано на рис. 8.3, нейрон в распознающем слое имеет, максимальную реакцию, если вектор  $\mathbf{C}$ , являющийся выходом слоя сравнения, соответствует набору его весов, следовательно, веса представляют запомненный образ или экземпляр для категории входных векторов. Эти веса являются действительными числами, а не двоичными величинами. Двоичная версия этого образа также запоминается в соответствующем наборе весов слоя сравнения (рис. 8.2); этот набор состоит из весов связей, соединяющих определенные нейроны слоя распознавания, один вес на каждый нейрон слоя сравнения.

В процессе функционирования каждый нейрон слоя распознавания вычисляет свертку вектора собственных весов и входного вектора  $\mathbf{C}$ . Нейрон, имеющий веса, наиболее близкие вектору  $\mathbf{C}$ , будет иметь самый большой выход, тем самым выигрывая соревнование и одновременно затормаживая все остальные нейроны в слое.

Как показано на рис. 8.4, нейроны внутри слоя распознавания взаимно соединены в латерально-тормозящую сеть. В простейшем случае (единственном, рассмотренном в данной работе) предусматривается, что только один нейрон в слое возбуждается в каждый момент времени (т. е. только нейрон с наивысшим уровнем активации будет иметь единичный выход; все остальные нейроны будут иметь нулевой выход). Эта конкуренция реализуется введением связей с отрицательными весами  $I_{ij}$  с выхода каждого нейрона  $r_i$  на входы остальных нейронов. Таким образом, если нейрон имеет большой выход, он тормозит все остальные нейроны в слое. Кроме того, каждый нейрон имеет связь с положительным весом со своего выхода на свой собственный вход. Если нейрон имеет единичный выходной уровень, эта обратная связь стремится усилить и поддержать его.



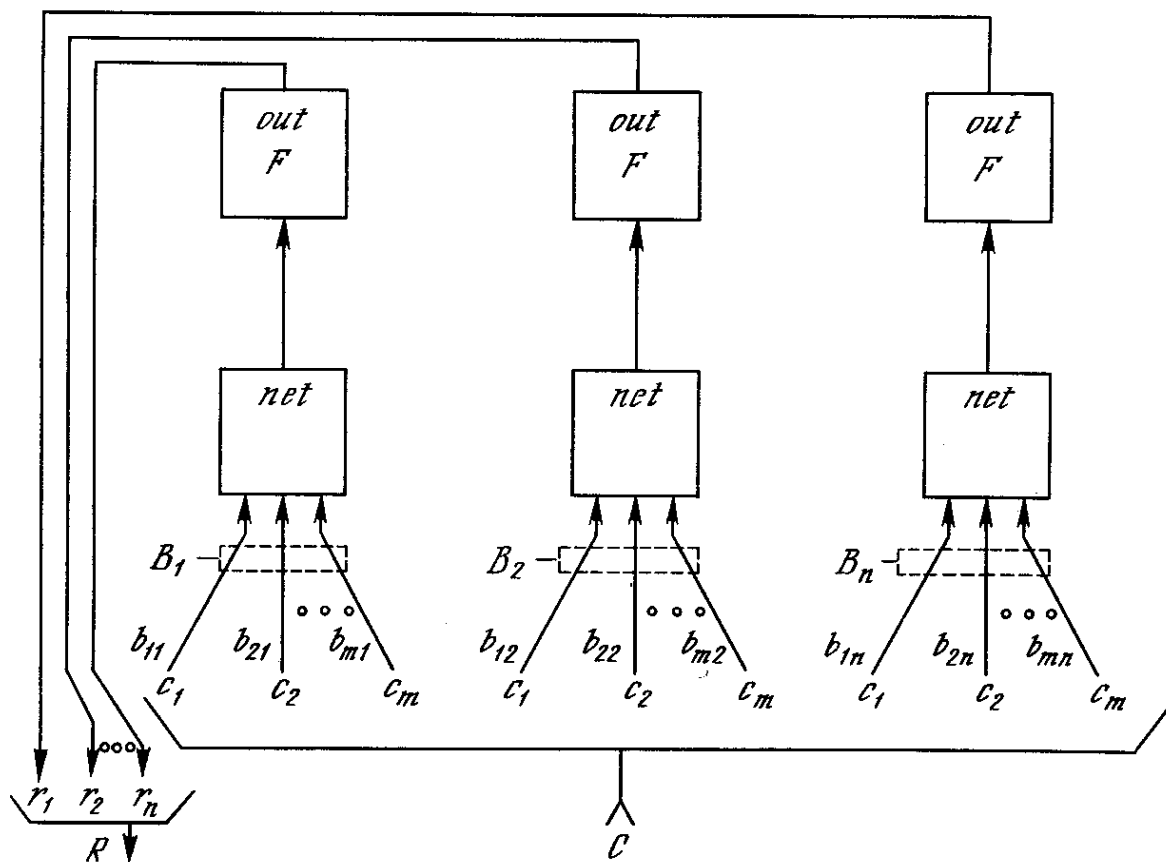


Рис. 8.3. Упрощенный слой распознавания

Приемник 2.  $G_2$ , выход Приемника 2, равен единице, если входной вектор  $X$  имеет хотя бы одну единичную компоненту. Более точно,  $G_2$  является логическим ИЛИ от компонента вектора  $X$ .

Приемник 1. Как и сигнал  $G_2$ , выходной сигнал  $G_1$  Приемника 1 равен 1, если хотя бы одна компонента двоичного входного вектора  $X$  равна единице; однако если хотя бы одна компонента вектора  $R$  равна единице,  $G_1$  устанавливается в нуль. Таблица, определяющая эти соотношения:

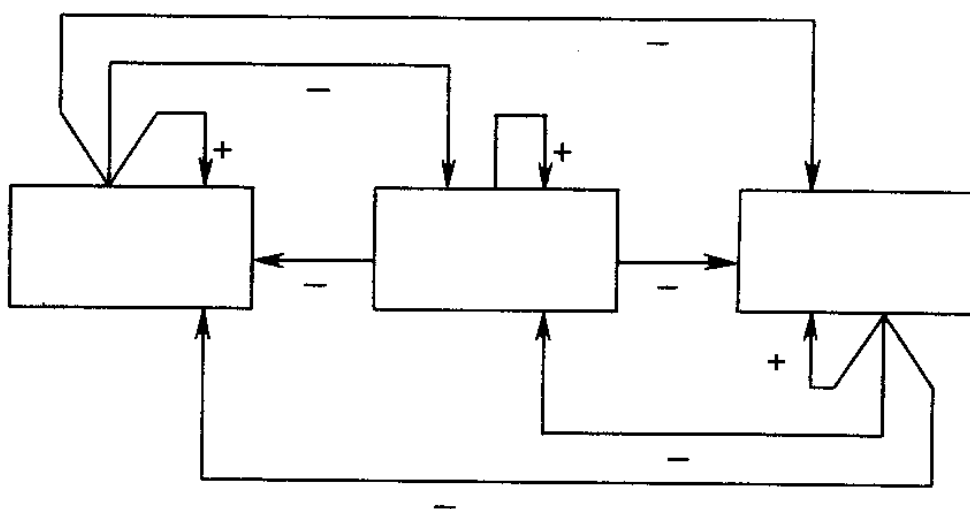


Рис. 8.4. Слой распознавания с латеральным торможением

ИЛИ от компонента вектора $\mathbf{X}$	ИЛИ от компонента вектора $\mathbf{R}$	G1
0	0	0
1	0	1
1	1	0
0	1	0

**Сброс.** Модуль сброса измеряет сходство между векторами  $\mathbf{X}$  и  $\mathbf{C}$ . Если они отличаются сильнее, чем требует параметр сходства, вырабатывается сигнал сброса возбужденного нейрона в слое распознавания.

В процессе функционирования модуль сброса вычисляет сходство как отношение количества единиц в векторе  $\mathbf{C}$  к их количеству в векторе  $\mathbf{S}$ . Если это отношение ниже значения параметра сходства, вырабатывается сигнал сброса.

### Функционирование сети АРТ в процессе классификации

Процесс классификации в АРТ состоит из трех основных фаз: распознавание, сравнение и поиск.

**Фаза распознавания.** В начальный момент времени входной вектор отсутствует на входе сети; следовательно, все компоненты входного вектора  $\mathbf{X}$  можно рассматривать как нулевые. Тем самым сигнал G2 устанавливается в 0 и, следовательно, в нуль устанавливаются выходы всех нейронов слоя распознавания. Поскольку все нейроны слоя распознавания начинают работу в одинаковом состоянии, они имеют равные шансы выиграть в последующей конкуренции.

Затем на вход сети подается входной вектор  $\mathbf{X}$ , который должен быть классифицирован. Этот вектор должен иметь одну или более компонент, отличных от нуля, в результате чего и G1, и G2 становятся равными единице. Это «подкачивает» нейроны слоя сравнения, обеспечивая один из двух единичных входов, необходимых для возбуждения нейронов в соответствии с правилом двух третей, тем самым позволяя нейрону возбуждаться, если соответствующая компонента входного вектора  $\mathbf{X}$  равна единице. Таким образом, в течение данной фазы вектор  $\mathbf{S}$  в точности дублирует вектор  $\mathbf{X}$ .

Далее для каждого нейрона в слое распознавания вычисляется свертка вектора его весов  $V_j$  и вектора  $\mathbf{C}$  (рис. 8.4). Нейрон с максимальным значением свертки имеет веса, наилучшим образом соответствующие входному вектору. Он выигрывает конкуренцию и возбуждается, одновременно затормаживая все остальные нейроны этого слоя. Таким образом, единственная компонента  $r_j$  вектора  $\mathbf{R}$  (рис. 8.1) становится равной единице, а все остальные компоненты становятся равными нулю.

В результате, сеть АРТ запоминает образы в весах нейронов слоя распознавания, один нейрон для каждой категории классификации. Нейрон слоя распознавания, веса которого наилучшим образом соответствуют входному вектору, возбуждается, его выход устанавливается в единичное значение, а выходы остальных нейронов этого слоя устанавливаются в нуль.

**Фаза сравнения.** Единственный возбужденный в слое распознавания нейрон возвращает единицу обратно в слой сравнения в виде своего выходного сигнала  $r_j$ . Эта единственная единица может быть визуально представлена в виде «всеерного» выхода, подающегося через отдельную связь с весом  $t_{ij}$  на каждый нейрон в слое сравнения, обеспечивая каждый нейрон сигналом  $p_j$ , равным величине  $t_{ij}$  (нулю или единице) (рис. 8.5).

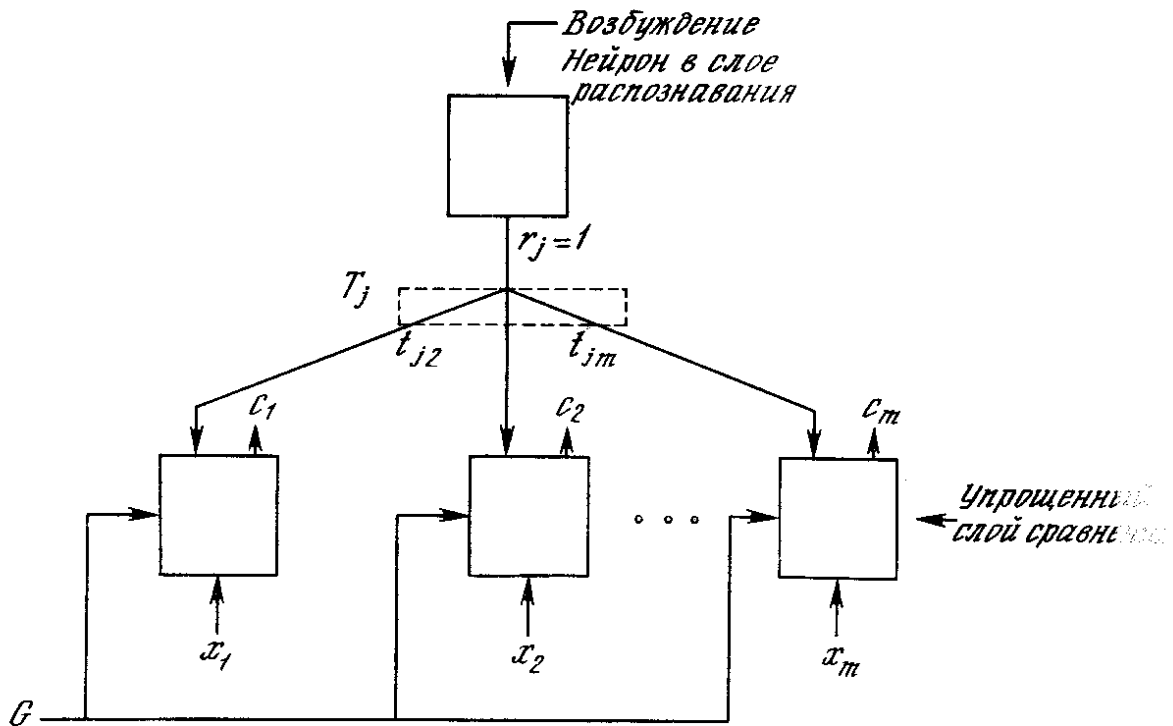


Рис. 8.5. Путь сигнала отдельного возбужденного нейрона в слое распознавания

Алгоритмы инициализации и обучения построены таким образом, что каждый весовой вектор  $T_j$  имеет двоичные значения весов; кроме того, каждый весовой вектор  $B_j$  представляет собой масштабированную версию соответствующего вектора  $T_j$ . Это означает, что все компоненты  $P$  (вектора возбуждения слоя сравнения) также являются двоичными величинами.

Так как вектор  $R$  не является больше нулевым, сигнал  $G1$  устанавливается в нуль. Таким образом, в соответствии с правилом двух третей, возбудиться могут только нейроны, получающие на входе одновременно единицы от входного вектора  $X$  и вектора  $P$ .

Другими словами, обратная связь от распознающего слоя действует таким образом, чтобы установить компоненты  $C$  в нуль в случае, если входной вектор не соответствует входному образцу, т. е. если  $X$  и  $P$  не имеют совпадающих компонент.

Если имеются существенные различия между  $X$  и  $P$  (малое количество совпадающих компонент векторов), несколько нейронов на фазе сравнения будут возбуждаться и  $C$  будет содержать много нулей, в то время как  $X$  содержит единицы. Это означает, что возвращенный вектор  $P$  не является искомым и возбужденные нейроны в слое распознавания должны быть заторможены. Это торможение производится блоком сброса (рис. 8.1), который сравнивает входной вектор  $X$  и вектор  $C$  и вырабатывает сигнал сброса, если степень сходства этих векторов меньше некоторого уровня. Влияние сигнала сброса заключается в установке выхода возбужденного нейрона в нуль, отключая его на время текущей классификации.

**Фаза поиска.** Если не выработан сигнал сброса, сходство является адекватным, и процесс классификации завершается. В противном случае другие запомненные образы должны быть исследованы с целью поиска лучшего соответствия. При этом торможение возбужденного нейрона в распознающем слое приводит к установке всех компонент вектора  $R$  в 0,  $G1$  устанавливается в 1 и входной вектор  $X$  опять прикладывается в качестве  $C$ . В результате другой нейрон выигрывает соревнование в слое распознавания и другой запомненный образ  $P$  возвращается в слой сравнения. Если  $P$  не соответствует  $X$ , возбужденный нейрон в слое распознавания снова тормозится. Этот процесс повторяется до тех пор, пока не встретится одно из двух событий:

1. Найден запомненный образ, сходство которого с вектором  $X$  выше уровня параметра сходства, т. е.  $S > \rho$ . Если это происходит, проводится обучающий цикл, в процессе которого модифицируются веса векторов  $T_j$  и  $B_j$ , связанных с возбужденным нейроном в слое распознавания.
2. Все запомненные образы проверены, определено, что они не соответствуют входному вектору, и все нейроны слоя распознавания заторможены. В этом случае предварительно не распределенный нейрон в распознающем слое выделяется этому образу и его весовые векторы  $B_j$  и  $T_j$  устанавливаются соответствующими новому входному образу.

**Проблема производительности.** Описанная сеть должна производить последовательный поиск среди всех запомненных образов. В аналоговых реализациях это будет происходить очень быстро; однако при моделировании на обычных цифровых компьютерах этот процесс может оказаться очень длительным. Если же сеть АРТ реализуется на параллельных процессорах, все свертки на распознающем уровне могут вычисляться одновременно. В этом случае поиск может быть очень быстрым.

Время, необходимое для стабилизации сети с латеральным торможением, может быть длительным при моделировании на последовательных цифровых компьютерах. Чтобы выбрать победителя в процессе латерального торможения, все нейроны в слое должны быть вовлечены в одновременные вычисления и передачу. Это может потребовать проведения большого объема вычислений перед достижением сходимости. Латеральные тормозящие сети, аналогичные используемым в неокогнитронах, могут существенно сократить это время (гл. 10).

## РЕАЛИЗАЦИЯ АРТ

### Обзор

АРТ, как это можно увидеть из литературы, представляет собой нечто большее, чем философию, но намного менее конкретное, чем программа для компьютера. Это привело к наличию широкого круга реализации, сохраняющих идеи АРТ, но сильно отличающихся в деталях. Рассматриваемая далее реализация основана на работе [5] с определенными изменениями для обеспечения совместимости с работой [2] и моделями, рассмотренными в данной работе. Эта реализация может рассматриваться в качестве типовой, но необходимо иметь в виду, что другие успешные реализации имеют большие отличия от нее.

### Функционирование сетей АРТ

Рассмотрим более детально пять фаз процесса функционирования АРТ: инициализацию, распознавание, сравнение, поиск и обучение.

**Инициализация.** Перед началом процесса обучения сети все весовые векторы  $\mathbf{V}_j$  и  $\mathbf{T}_j$ , а также параметр сходства  $\rho$ , должны быть установлены в начальные значения.

Веса векторов  $\mathbf{V}_j$  все инициализируются в одинаковые малые значения. Согласно [2], эти значения должны удовлетворять условию

$$b_{ij} < \frac{L}{L-1+m} \quad \text{для всех } i, j, \quad (8.1)$$

где  $m$  – количество компонент входного вектора,  $L$  – константа, большая 1 (обычно  $L = 2$ ).

Эта величина является критической; если она слишком большая, сеть может распределить все нейроны распознающего слоя одному входному вектору.

Веса векторов  $\mathbf{T}_j$  все инициализируются в единичные значения, так что

$$t_{ij} = 1 \quad \text{для всех } j, i. \quad (8.2)$$

Эти значения также являются критическими; в [2] показано, что слишком маленькие веса приводят к отсутствию соответствия в слое сравнения и отсутствию обучения.

Параметр сходства  $\rho$  устанавливается в диапазоне от 0 до 1 в зависимости от требуемой степени сходства между запомненным образом и входным вектором. При высоких значениях  $\rho$  сеть относит к одному классу только очень слабо отличающиеся образы. С другой стороны, малое значение  $\rho$  заставляет сеть группировать образы, которые имеют слабое сходство между собой. Может оказаться желательной возможность изменять коэффициент сходства на протяжении процесса обучения, обеспечивая только грубую классификацию в начале процесса обучения, и затем постепенно увеличивая коэффициент сходства для выработки точной классификации в конце процесса обучения.

**Распознавание.** Появление на входе сети входного вектора  $\mathbf{X}$  инициализирует фазу распознавания. Так как вначале выходной вектор слоя распознавания отсутствует, сигнал G1 устанавливается в 1 функцией ИЛИ вектора  $\mathbf{X}$ , обеспечивая все нейроны слоя сравнения одним из двух входов, необходимых для их возбуждения (как требует правило двух третей). В результате любая компонента вектора  $\mathbf{X}$ , равная единице, обеспечивает второй единичный вход, тем самым заставляя соответствующий нейрон слоя сравнения возбуждаться и устанавливая его выход в единицу. Таким образом, в этот момент времени вектор  $\mathbf{C}$  идентичен вектору  $\mathbf{X}$ .

Как обсуждалось ранее, распознавание реализуется вычислением свертки для каждого нейрона слоя распознавания, определяемой следующим выражением:

$$NET_j = (\mathbf{B}_j \cdot \mathbf{C}), \quad (8.3)$$

где  $\mathbf{B}_j$  – весовой вектор, соответствующий нейрону  $j$  в слое распознавания;  $\mathbf{C}$  – выходной вектор нейронов слоя сравнения; в этот момент  $\mathbf{C}$  равно  $\mathbf{X}$ ;  $NET_j$  – возбуждение нейрона  $j$  в слое распознавания.

$F$  является пороговой функцией, определяемой следующим образом:

$$\begin{aligned} OUT_j &= 1, \text{ если } NET_j > T, \\ OUT_j &= 0 \text{ в противном случае,} \end{aligned} \quad (8.4)$$

где  $T$  представляет собой порог.

Принято, что латеральное торможение существует, но игнорируется здесь для сохранения простоты выражения. Оно обеспечивает тот факт, что только нейрон с максимальным значением  $NET$  будет иметь выход, равный единице; все остальные нейроны будут иметь нулевой выход. Можно рассмотреть системы, в которых в распознающем слое возбуждаются несколько нейронов в каждый момент времени, однако это выходит за рамки данной работы.

**Сравнение.** На этой фазе сигнал обратной связи от слоя распознавания устанавливает  $G1$  в нуль; правило двух третей позволяет возбуждаться только тем нейронам, которые имеют равные единице соответствующие компоненты векторов  $\mathbf{P}$  и  $\mathbf{X}$ .

Блок сброса сравнивает вектор  $\mathbf{C}$  и входной вектор  $\mathbf{X}$ , вырабатывая сигнал сброса, когда их сходство  $S$  ниже порога сходства. Вычисление этого сходства упрощается тем обстоятельством, что оба вектора являются двоичными (все элементы либо 0, либо 1). Следующая процедура проводит требуемое вычисление сходства:

1. Вычислить  $D$  – количество единиц в векторе  $\mathbf{X}$ .
2. Вычислить  $N$  – количество единиц в векторе  $\mathbf{C}$ .

Затем вычислить сходство  $S$  следующим образом:

$$S = N/D. \quad (8.5)$$

Например, примем, что

$$\begin{aligned} \mathbf{X} &= 1\ 0\ 1\ 1\ 1\ 0\ 1 & D &= 5 \\ \mathbf{C} &= 0\ 0\ 1\ 1\ 1\ 0\ 1 & N &= 4 \end{aligned}$$

$$S = N/D = 0,8$$

$S$  может изменяться от 1 (наилучшее соответствие) до 0 (наихудшее соответствие).

Заметим, что правило двух третей делает  $\mathbf{C}$  логическим произведением входного вектора  $\mathbf{X}$  и вектора  $\mathbf{P}$ . Однако  $\mathbf{P}$  равен  $\mathbf{T}_j$ , весовому вектору выигравшего соревнования нейрона. Таким образом,  $D$  может быть определено как количество единиц в логическом произведении векторов  $\mathbf{T}_j$  и  $\mathbf{X}$ .

**Поиск.** Если сходство  $S$  выигравшего нейрона превышает параметр сходства, поиск не требуется. Однако если сеть предварительно была обучена, появление на входе вектора, не идентичного ни одному из предъявленных ранее, может возбудить в слое распознавания нейрон со сходством ниже требуемого уровня. В соответствии с алгоритмом обучения возможно, что другой нейрон в слое распознавания будет обеспечивать более хорошее соответствие, превышая требуемый уровень сходства несмотря на то, что свертка между его весовым вектором и входным вектором может иметь меньшее значение. Пример такой ситуации показан ниже.

Если сходство ниже требуемого уровня, запомненные образы могут быть просмотрены с целью поиска, наиболее соответствующего входному вектору образа. Если такой образ отсутствует, вводится новый несвязанный нейрон, который в дальнейшем будет обучен. Для инициализации поиска сигнал сброса тормозит возбужденный нейрон в слое распознавания на время проведения поиска, сигнал  $G1$  устанавливается в единицу и другой нейрон в слое распознавания выигрывает соревнование. Его запомненный образ затем проверяется на сходство и процесс повторяется до тех пор, пока конкуренцию не выиграет нейрон из слоя распознавания со сходством, большим требуемого уровня (успешный поиск), либо пока все связанные нейроны не будут проверены и заторможены (неудачный поиск).

Неудачный поиск будет автоматически завершаться на несвязанном нейроне, так как его веса все равны единице, своему начальному значению. Поэтому правило двух третей приведет к идентичности вектора  $\mathbf{C}$  входному вектору  $\mathbf{X}$ , сходство  $S$  примет значение единицы и критерий сходства будет удовлетворен.

**Обучение.** Обучение представляет собой процесс, в котором набор входных векторов подается последовательно на вход сети и веса сети изменяются при этом таким образом, чтобы сходные векторы активизировали соответствующие нейроны. Заметим, что это – неуправляемое обучение, нет учителя и нет целевого вектора, определяющего требуемый ответ.

В работе [2] различают два вида обучения: медленное и быстрое. При медленном обучении входной вектор предъявляется настолько кратковременно, что веса сети не имеют достаточного времени для достижения своих асимптотических значений в результате одного предъявления. В этом случае значения весов будут определяться скорее статистическими характеристиками входных векторов, чем характеристиками какого-то одного входного вектора. Динамика сети в процессе медленного обучения описывается дифференциальными уравнениями.

Быстрое обучение является специальным случаем медленного обучения, когда входной вектор прикладывается на достаточно длительный промежуток времени, чтобы позволить весам приблизиться к их окончательным значениям. В этом случае процесс обучения описывается только алгебраическими выражениями. Кроме того, компоненты весовых векторов  $T_j$  принимают двоичные значения, в отличие от непрерывного диапазона значений, требуемого в случае быстрого обучения. В данной работе рассматривается только быстрое обучение, интересующиеся читатели могут найти превосходное описание более общего случая медленного обучения в работе [2].

Рассмотренный далее обучающий алгоритм используется как в случае успешного, так и в случае неуспешного поиска.

Пусть вектор весов  $V_j$  (связанный с возбужденным нейроном  $j$  распознающего слоя) равен нормализованной величине вектора  $C$ . В [2] эти веса вычисляются следующим образом:

$$b_{ij} = \frac{Lc_i}{L-1 + \sum_k c_k} \quad (8.6)$$

где  $c_i$  –  $i$ -я компонента выходного вектора слоя сравнения;  $j$  – номер выигравшего нейрона в слое распознавания;  $b_{ij}$  – вес связи, соединяющей нейрон  $i$  в слое сравнения с нейроном  $j$  в слое распознавания;  $L$  – константа  $> 1$  (обычно 2).

Компоненты вектора весов  $T_j$ , связанного с новым запомненным вектором, изменяются таким образом, что они становятся равны соответствующим двоичным величинам вектора  $C$ :

$$t_{ij} = c_i \quad \text{для всех } i, \quad (8.7)$$

где  $t_{ij}$  является весом связи между выигравшим нейроном  $j$  в слое распознавания и нейроном  $i$  в слое сравнения.

## ПРИМЕР ОБУЧЕНИЯ СЕТИ АРТ

В общих чертах сеть обучается посредством изменения весов таким образом, что предъявление сети входного вектора заставляет сеть активизировать нейроны в слое распознавания, связанные с сходным запомненным вектором. Кроме этого, обучение проводится в форме, не разрушающей запомненные ранее образы, предотвращая тем самым временную нестабильность. Эта задача управляется на уровне выбора критерия сходства. Новый входной образ (который сеть не видела раньше) не будет соответствовать запомненным образам с точки зрения параметра сходства, тем самым формируя новый запоминаемый образ. Входной образ, в достаточной степени соответствующий одному из запомненных образов, не будет формировать нового экземпляра, он просто будет модифицировать тот, на который он похож. Таким образом при соответствующем выборе критерия сходства предотвращается запоминание ранее изученных образов и временная нестабильность.

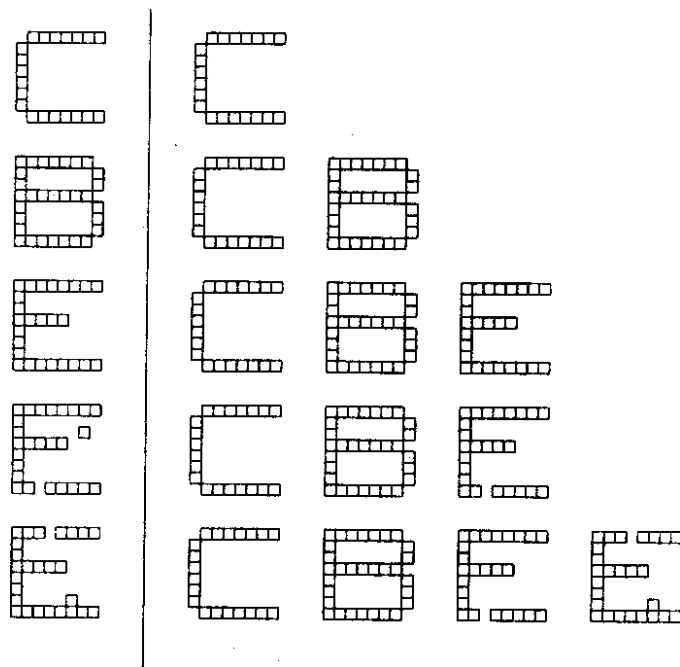


Рис. 8.6. Процесс обучения АРТ

На рис. 8.6 показан типичный сеанс обучения сети АРТ. Буквы показаны состоящими из маленьких квадратов, каждая буква размерностью 8x8. Каждый квадрат в левой части представляет компоненту вектора  $X$  с единичным значением, не показанные квадраты являются компонентами с нулевыми значениями. Буквы справа представляют запомненные образы, каждый является набором величин компонент вектора  $T_j$ .

Вначале на вход заново проиницированной системы подается буква «С». Так как отсутствуют запомненные образы, фаза поиска заканчивается неуспешно; новый нейрон выделяется в слое распознавания, и веса  $T_j$  устанавливаются равными соответствующим компонентам входного вектора, при этом веса  $V_j$  представляют масштабированную версию входного вектора.

Далее предъявляется буква «В». Она также вызывает неуспешное окончание фазы поиска и распределение нового нейрона. Аналогичный процесс повторяется для буквы «Е». Затем слабо искаженная версия буквы «Е» подается на вход сети. Она достаточно точно соответствует запомненной букве «Е», чтобы выдержать проверку на сходство, поэтому используется для обучения сети. Отсутствующий пиксель в нижней ножке буквы «Е» устанавливает в 0 соответствующую компоненту вектора  $C$ , заставляя обучающий алгоритм установить этот вес запомненного образа в нуль, тем самым воспроизводя искажения в запомненном образе. Дополнительный изолированный квадрат не изменяет запомненного образа, так как не соответствует единице в запомненном образе.

Четвертым символом является буква «Е» с двумя различными искажениями. Она не соответствует ранее запомненному образу ( $S$  меньше чем  $\rho$ ), поэтому для ее запоминания выделяется новый нейрон.

Этот пример иллюстрирует важность выбора корректного значения критерия сходства. Если значение критерия слишком велико, большинство образов не будут подтверждать сходство с ранее запомненными и сеть будет выделять новый нейрон для каждого из них. Это приводит к плохому обобщению в сети, в результате даже незначительные изменения одного образа будут создавать отдельные новые категории. Количество категорий увеличивается, все доступные нейроны распределяются, и способность системы к восприятию новых данных теряется. Наоборот, если критерий сходства слишком мал, сильно различающиеся образы будут группироваться вместе, искажая запомненный образ до тех пор, пока в результате не получится очень малое сходство с одним из них.

К сожалению, отсутствует теоретическое обоснование выбора критерия сходства, в каждом конкретном случае необходимо решить, какая степень сходства должна быть принята для отнесения образов к одной категории. Границы между категориями часто неясны, и решение задачи для большого набора входных векторов может быть чрезмерно трудным.

В работе [2] предложена процедура с использованием обратной связи для настройки коэффициента сходства, вносящая, однако, некоторые искажения в результате классификации как «наказание» за внешнее вмешательство с целью увеличения коэффициента сходства. Такие системы требуют правил определения, является ли производимая ими классификация корректной.

## ХАРАКТЕРИСТИКИ АРТ

Системы АРТ имеют ряд важных характеристик, не являющихся очевидными. Формулы и алгоритмы могут казаться произвольными, в то время как в действительности они были тщательно отобраны с целью удовлетворения требований теорем относительно производительности систем АРТ. В данном разделе описываются некоторые алгоритмы АРТ, раскрывающие отдельные вопросы инициализации и обучения.

### Инициализация весовых векторов $T$

Из ранее рассмотренного примера обучения сети можно было видеть, что правило двух третей приводит к вычислению вектора  $C$  как функции И между входным вектором  $X$  и выигравшим соревнование запомненным вектором  $T_j$ . Следовательно, любая компонента вектора  $C$  будет равна единице в том случае, если соответствующие компоненты обоих векторов равны единице. После обучения эти компоненты вектора  $T_j$  остаются единичными; все остальные устанавливаются в нуль.

Это объясняет, почему веса  $t_{ij}$  должны инициализироваться единичными значениями. Если бы они были проинициализированы нулевыми значениями, все компоненты вектора  $C$  были бы нулевыми независимо от значений компонент входного вектора, и обучающий алгоритм предохранял бы веса от изменения их нулевых значений.

Обучение может рассматриваться как процесс «сокращения» компонент запомненных векторов, которые не соответствуют входным векторам. Этот процесс необратим, если вес однажды установлен в нуль, обучающий алгоритм никогда не восстановит его единичное значение.

Это свойство имеет важное отношение к процессу обучения. Предположим, что группа точно соответствующих векторов должна быть классифицирована к одной категории, определяемой возбуждением одного нейрона в слое распознавания. Если эти вектора последовательно предъявляются сети, при предъявлении первого будет распределяться нейрон распознающего слоя, его веса будут обучены с целью соответствия входному вектору. Обучение при предъявлении остальных векторов будет приводить к обнулению весов в тех позициях, которые имеют нулевые значения в любом из входных векторов. Таким образом, запомненный вектор представляет собой логическое пересечение всех обучающих векторов и может включать существенные характеристики данной категории весов. Новый вектор, включающий только существенные характеристики, будет соответствовать этой категории. Таким образом, сеть корректно распознает образ, никогда не виденный ранее, т. е. реализуется возможность, напоминающая процесс восприятия человека.

### Настройка весовых векторов $V_j$

Выражение, описывающее процесс настройки весов (выражение (8.6) повторено здесь для справки) является центральным для описания процесса функционирования сетей АРТ.

$$b_{ij} = \frac{Lc_i}{L-1 + \sum_k c_k} \quad (8.6)$$

Сумма в знаменателе представляет собой количество единиц на выходе слоя сравнения. Эта величина может быть рассмотрена как «размер» этого вектора. В такой интерпретации «большие» векторы  $C$  производят более маленькие величины весов  $b_{ij}$ , чем «маленькие» вектора  $C$ . Это свойство самомасштабирования делает возможным разделение двух векторов в случае, когда один вектор является поднабором другого; т. е. когда набор единичных компонент одного вектора составляет подмножество единичных компонент другого.

Чтобы продемонстрировать проблему, возникающую при отсутствии масштабирования, используемого в выражении (8.6), предположим, что сеть обучена двум приведенным ниже входным векторам, при этом каждому распределен нейрон в слое распознавания.

Заметим, что  $X_1$  является поднабором  $X_2$ . В отсутствие свойства масштабирования веса  $b_{ij}$  и  $t_{ij}$  получают значения, идентичные значениям входных векторов. Если начальные значения выбраны равными 1,0, веса образов будут иметь следующие значения:

Если  $X$  прикладывается повторно, оба нейрона в слое распознавания получают одинаковые активации; следовательно, нейрон 2, ошибочный нейрон, выигрывает конкуренцию.

Кроме выполнения некорректной классификации, может быть нарушен процесс обучения. Так как  $T_2$  равно 111100, только первая единица соответствует единице входного вектора, и  $C$  устанавливается в 10000, критерий сходства удовлетворяется и алгоритм обучения устанавливает вторую и третью единицы векторов  $T_2$  и  $V_2$  в нуль, разрушая запомненный образ.



Масштабирование весов  $b_{ij}$  предотвращает это нежелательное поведение. Предположим, что в выражении (8.2) используется значение  $L=2$ , тем самым определяя следующую формулу:

$$b_{ij} = \frac{2c_i}{L-1 + \sum_k c_k}$$

Значения векторов будут тогда стремиться к величинам

Подавая на вход сети вектор  $X_1$ , получим возбуждающее воздействие 1,0 для нейрона 1 в слое распознавания и  $\frac{1}{2}$  для нейрона 2; таким образом, нейрон 1 (правильный) выиграет соревнование. Аналогично предъявление вектора  $X_2$  вызовет уровень возбуждения 1,0 для нейрона 1 и  $\frac{3}{2}$  для нейрона 2, тем самым снова правильно выбирая победителя.

### Инициализация весов $b_{ij}$

Инициализация весов  $b_{ij}$  малыми значениями является существенной для корректного функционирования систем АРТ. Если они слишком большие, входной вектора который ранее был запомнен, будет скорее активизировать несвязанный нейрон, чем ранее обученный. Выражение (8.1), определяющее начальные значения весов, повторяется здесь для справки

$$b_{ij} < \frac{L}{L-1+m} \quad \text{для всех } i, j, \quad (8.1)$$

Установка этих весов в малые величины гарантирует, что несвязанные нейроны не будут получать возбуждения большего, чем обученные нейроны в слое распознавания. Используя предыдущий пример с  $L=2$ ,  $m=5$  и  $b_{ij} < 1/3$ , произвольно установим  $b_{ij}=1/6$ . С такими весами предъявление вектора, которому сеть была ранее обучена, приведет к более высокому уровню активации для правильно обученного нейрона в слое распознавания, чем для несвязанного нейрона. Например, для несвязанного нейрона  $X_1$  будет производиться возбуждение  $1/6$ , в то время как  $X_2$  будет производить возбуждение  $\frac{1}{2}$ ; и то и другое ниже возбуждения для обученных нейронов.

**Поиск.** Может показаться, что в описанных алгоритмах отсутствует необходимость наличия фазы поиска за исключением случая, когда для входного вектора должен быть распределен новый несвязанный нейрон. Это не совсем так; предъявление входного вектора, сходного, но не абсолютно идентичного одному из запомненных образов, может при первом испытании не обеспечить выбор нейрона слоя распознавания с уровнем сходства большим  $p$ , хотя такой нейрон будет существовать.

Как и в предыдущем примере, предположим, что сеть обучается следующим двум векторам:

$$X_1 = 1\ 0\ 0\ 0\ 0$$

$$X_2 = 1\ 1\ 1\ 0\ 0$$

с векторами весов  $V_i$ , обученными следующим образом

$$V_1 = 1\ 0\ 0\ 0\ 0$$

$$V_2 = \frac{1}{2}\ \frac{1}{2}\ \frac{1}{2}\ 0\ 0$$

Теперь приложим входной вектор  $X_3 = 1\ 1\ 0\ 0\ 0$ . В этом случае возбуждение нейрона 1 в слое распознавания будет 1,0, а нейрона 2 только  $\frac{2}{3}$ . Нейрон 1 выйдет победителем (хотя он не лучшим образом соответствует входному вектору), вектор  $C$  получит значение  $1\ 1\ 0\ 0\ 0$ ,  $S$  будет равно  $\frac{1}{2}$ . Если уровень сходства установлен в  $\frac{3}{4}$ , нейрон 1 будет заторможен и нейрон 2 выиграет состязание.  $C$  станет равным  $1\ 1\ 0\ 0\ 0$ ,  $S$  станет равным 1, критерий сходства будет удовлетворен и поиск закончится.

### Теоремы АРТ

В работе [2] доказаны некоторые теоремы, показывающие характеристики сетей АРТ. Четыре результата, приведенные ниже, являются одними из наиболее важных:

1. После стабилизации процесса обучения предъявление одного из обучающих векторов (или вектора с существенными характеристиками категории) будет активизировать требуемый нейрон слоя распознавания без поиска. Эта характеристика «прямого доступа» определяет быстрый доступ к предварительно изученным образам.
2. Процесс поиска является устойчивым. После определения выигравшего нейрона в сети не будет возбуждений других нейронов в результате изменения векторов выхода слоя сравнения  $C$ ; только сигнал сброса может вызвать такие изменения.

3. Процесс обучения является устойчивым. Обучение не будет вызывать переключения с одного возбужденного нейрона слоя распознавания на другой.
4. Процесс обучения конечен. Любая последовательность произвольных входных векторов будет производить стабильный набор весов после конечного количества обучающих серий; повторяющиеся последовательности обучающих векторов не будут приводить к циклическому изменению весов.

## **ЗАКЛЮЧЕНИЕ**

Сети АРТ являются интересным и важным видом систем. Они способны решить дилемму стабильности-пластичности и хорошо работают с других точек зрения. Архитектура АРТ сконструирована по принципу биологического подобия; это означает, что ее механизмы во многом соответствуют механизмам мозга (как мы их понимаем). Однако они могут оказаться не в состоянии моделировать распределенную память, которую многие рассматривают как важную характеристику функций мозга. Экземпляры АРТ представляют собой «бабушкины узелки»; потеря одного узла разрушает всю память. Память мозга, напротив, распределена по веществу мозга, запомненные образы могут часто пережить значительные физические повреждения мозга без полной их потери.

Кажется логичным изучение архитектур, соответствующих нашему пониманию организации и функций мозга. Человеческий мозг представляет существующее доказательство того факта, что решение проблемы распознавания образов возможно. Кажется разумным эмулировать работу мозга, если мы хотим повторить его работу. Однако контраргументом является история полетов; человек не смог оторваться от земли до тех пор, пока не перестал имитировать движения крыльев и полет птиц.

## **Литература**

1. Carpenter G., Grossberg S. 1986. Neural dynamics of category learning and recognition: Attention; memory consolidation and amnesia. In Brain Structure, Learning and Memory (AAAS Symposium Series), eds. J. Davis., R. Newburgh and E. Wegman.
2. Carpenter G., Grossberg S. 1987. A massively parallel architecture for a self-organizing neural pattern recognition machine. Computing Vision, Graphics, and Image Processing 37:54-115.
3. Carpenter G., Grossberg S. 1987 ART-2: Self-organization of stable category recognition codes for analog input patterns. Applied Optics 26(23):4919-30.
4. Grossberg S. 1987. Competitive learning: From interactive activation to adaptive resonance. Cognitive Science 11:23-63.
5. Lippman R. P. 1987. An introduction to computing with neural nets. IEEE Transactions on Acoustics, Speech and Signal Processing, April, pp. 4-22.

## Глава 9.

### Оптические нейронные сети

Использование и обучение нейронных сетей требует в основном двух типов операций над данными: вычислений и передачи данных. Вычислительные функции легко и просто выполняются электронными системами. Элементы интегральных цепей работают в наносекундных интервалах. Кроме того, они имеют размеры, измеряемые в микронах, и могут иметь стоимость менее сотой цента за вентиль.

Задачи передачи данных решаются не просто. Электронные сигналы в интегральных сетях требуют наличия конденсаторов для передачи сигналов от вентиля к вентилю. Хотя конденсаторы имеют микронные размеры, занимаемое пространство (с учетом пространства, необходимого для изоляции одного конденсатора от другого) может стать настолько большим, что на пластине кремния не останется места для размещения вычислительных цепей. Несмотря на то, что существует технология реализации обыкновенных цифровых компьютеров в виде больших функциональных блоков с относительно небольшим количеством конденсаторов, эта технология не годится в случае массового параллелизма. Аналогичное решение для искусственных нейронных сетей в настоящее время неизвестно. Мощность нейронных сетей определяется большим количеством связей; взятые отдельно элементы имеют относительно малые вычислительные возможности.

Серьезную проблему представляет достижение требуемой связанности в электронных цепях. В [10] предполагается, что плотность конденсаторов в двумерной системе должна уменьшаться обратно пропорционально квадрату расстояния от нейрона-источника; в противном случае отсутствует возможность реализации системы в виде интегральных цепей. Это ограничение имеет особое значение при реализации сетей с полными взаимными связями.

Реализация нейронных сетей в виде оптических систем позволяет решить эту проблему. Взаимное соединение нейронов с помощью световых лучей не требует изоляции между сигнальными путями, световые потоки могут проходить один через другой без взаимного влияния. Более того, сигнальные пути могут быть расположены в трех измерениях. (Интегральные цепи являются существенно планарными с некоторой рельефностью, обусловленной множеством слоев.) Плотность путей передачи ограничена только размерами источников света, их дивергенцией и размерами детектора. Потенциально эти размеры могут иметь величину в несколько микрон. Наконец, все сигнальные пути могут работать одновременно, тем самым обеспечивая огромный темп передачи данных. В результате система способна обеспечить полный набор связей, работающих со скоростью света.

Оптические нейронные сети могут также обеспечить важные преимущества при проведении вычислений. Величина синаптических связей может запоминаться в голограммах с высокой степенью плотности; некоторые оценки дают теоретический предел в  $10^{12}$  бит на кубический сантиметр. Хотя такие значения на практике не достигнуты, существующий уровень плотности памяти очень высок. Кроме того, веса могут модифицироваться в процессе работы сети, образуя полностью адаптивную систему.

Учитывая эти преимущества, можно задать вопрос, почему наряду с оптическими нейронными сетями вообще рассматриваются другие способы реализации. К сожалению, возникает множество практических проблем при попытках оптической реализации нейронных сетей. Оптические устройства имеют собственные физические характеристики, часто не соответствующие требованиям искусственных нейронных сетей. Хотя они в действительности пригодны для обработки изображений, изображения от оптических нейронных сетей, полученные до настоящего времени, были разочаровывающе плохими. Однако достаточно взглянуть на первые пробы телевизионных изображений, чтобы понять, какой огромный прогресс возможен в повышении качества изображения. Несмотря на эти трудности, а также на такие проблемы, как стоимость, размеры и критичность к ориентации, потенциальные возможности оптических систем побуждали (и побуждают) попытки проведения интенсивных и широких исследований. В этой области происходят стремительные изменения и в ближайшее время ожидаются важные улучшения.

Многие изучаемые конфигурации оптических нейронных сетей можно разделить на две категории, рассмотренные в данной главе: векторно-матричные умножители и голографические корреляторы. Заметим, что детальное описание вопросов оптической физики выходит за рамки данной работы. Вместо этого приведено качественное описание работы систем и взгляд автора на достижения в этой области.

#### ВЕКТОРНО-МАТРИЧНЫЕ УМНОЖИТЕЛИ

Процесс функционирования большинства искусственных нейронных сетей может быть описан математически в виде последовательных умножений вектора на матрицу, одна операция умножения в каждом слое. Для вычисления выхода слоя входной вектор умножается на матрицу весовых коэффициентов, образуя вектор NET. К этому вектору прикладывается затем функция активации  $F$ , образуя вектор OUT, являющийся выходом слоя.

Символически

$$NET = \mathbf{XW},$$

$$OUT = F(NET),$$

где NET – вектор в виде строки, сформированный взвешенными суммами входов; OUT – выходной вектор; X – входной вектор; W – матрица весовых коэффициентов.

В биологических нейронных сетях эта операция выполняется большим количеством работающих одновременно нейронов, поэтому система работает быстро, несмотря на медленную работу отдельных нейронов.

Когда искусственные нейронные сети моделируются на универсальных компьютерах, присущая им параллельная природа вычислений теряется; каждая операция должна быть выполнена последовательно. Несмотря на большую скорость выполнения отдельных вычислений, количество операций, необходимых для выполнения умножения матриц, пропорционально квадрату размерности входного вектора (если входной и выходной векторы имеют одинаковую размерность), и время вычислений может стать слишком большим.

### Электронно-оптические матричные умножители

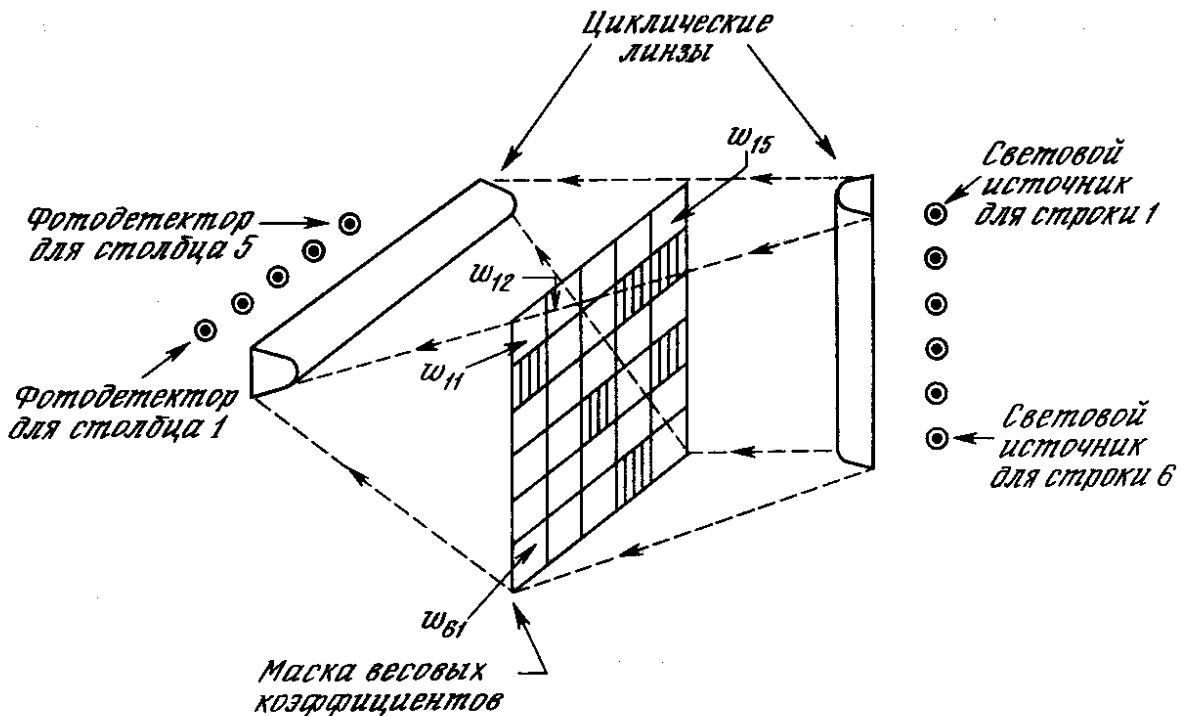
Электронно-оптические нейронные сети обеспечивают средства для выполнения параллельного умножения матриц. Рассмотренные в работах [3,6,7] такие сети работают со скоростью, ограниченной только доступными электронно-оптическими компонентами; время вычислений потенциально располагается в субнаносекундном диапазоне.

На рис. 9.1 показана система, способная выполнять умножение шестиэлементного входного вектора на матрицу размерностью 6x5, производя на выходе пятиэлементный вектор NET. Справа расположен столбец световых источников, лучи которых проходят через цилиндрические линзы; каждый луч освещает одну строку весовой маски.

Таким образом, луч 1 освещает  $w_{11}$   $w_{12}$   $w_{15}$ . В качестве маски может быть использована фотопленка, у которой прозрачность каждого квадрата пропорциональна весу. С левой стороны расположена вторая цилиндрическая линза, фокусирующая свет от каждого столбца маски на соответствующий фотодетектор. Таким образом, световой поток на фотодетекторе 1 является суммой произведений световых интенсивностей на передаточную функцию столбца 1. В символической форме

$$NET_j = \sum_i W_{ij} X_i,$$

где  $NET_j$  – выход NET нейрона  $j$  (выход фотодетектора  $j$ );  $W_{ij}$  – вес связи от нейрона  $i$  к нейрону  $j$  (величина обратно пропорциональная прозрачности весовой маски в строке  $i$ , столбце  $j$ );  $X_i$  –  $i$ -я компонента входного вектора  $i$  (выход источника света  $i$ ).



### Рис. 9.1. Электронно-оптический векторно-матричный умножитель

Выход каждого фотодетектора является сверткой между входным вектором и соответствующим столбцом весовой матрицы. Таким образом, набор выходов представляет собой вектор, равный произведению входного вектора на весовую матрицу.

Это матричное умножение выполняется параллельно. При использовании соответствующих высокоскоростных светоизлучающих диодов и фотодетекторов PIN умножение вектора на матрицу может быть выполнено менее чем за наносекунду. Более того, скорость умножения практически не зависит от размерности массива. Это позволяет наращивать сети без существенного увеличения времени вычислений. В данном простом примере веса сети фиксированы; они могут изменяться только при подстановке различных весовых масок. Для использования в адаптивных системах веса должны быть переменными. Существует многообещающий метод, основанный на использовании жидкокристаллического клапана вместо фотографического негатива. Это позволяет изменять веса электронным способом в течение микросекунд. В настоящее время жидкокристаллический клапан может использоваться для реализации двоичных весов, но имеет недостаточную стабильность и контрастность для реализации непрерывных переменных весов. Эта ситуация может измениться в ближайшем будущем.

#### Сети Хопфилда на базе электронно-оптических матричных умножителей

Если выходы фотодетекторов в сети подаются обратно для управления соответствующими световыми входами, реализуется электронно-оптическая сеть Хопфилда. Для выполнения этого должна быть обеспечена пороговая функция активации. В настоящее время функция активации наилучшим образом реализуется с помощью электронных цепей, следующих за каждым фотодетектором.

Для удовлетворения требований стабильности массив весов должен быть симметричным с нулевыми коэффициентами для квадратов главной диагонали ( $w_{11}, w_{12}, w_{mm}$ ).

**Электронно-оптическая двунаправленная ассоциативная память (ДАП).** Если две системы, показанные на рис. 9.1, соединены в каскад (выход второй системы подается на вход первой), реализуется электронно-оптическая ДАП. Для обеспечения стабильности вторая весовая маска должна являться транспозицией первой.

В [9] описана компактная система, в которой для реализации электронно-оптической ДАП требуется только одна маска и оптическая система (рис. 9.2). Здесь каждый фотодетектор и световой источник заменяются парой фотодетектор – световой источник. Функционирует данная система аналогично ранее описанному простому фотооптическому умножителю, за исключением того, что выход каждого фотодетектора управляется связанным с ним световым источником.

В процессе работы световой поток от каждого источника света справа проходит через цилиндрическую линзу, освещая соответствующую строку световой маски. Эта линза разворачивает световой поток в горизонтальном направлении, оставляя его неразвернутым в вертикальном направлении.

Каждый фотодетектор слева получает световой поток от столбца весовой маски и с помощью подключенных к нему электронных цепей реализует пороговую функцию для выработки электрического выходного сигнала NET. Сигнал NET затем управляет соответствующим данному фотодетектору световым источником, свет от которого проходит через оптическую подсистему, освещая тот же столбец. Следует отметить, что одно и то же оптическое пространство используется световыми образами, проходящими слева направо и справа налево. Так как световые потоки не взаимодействуют между собой, это не вызывает проблем.

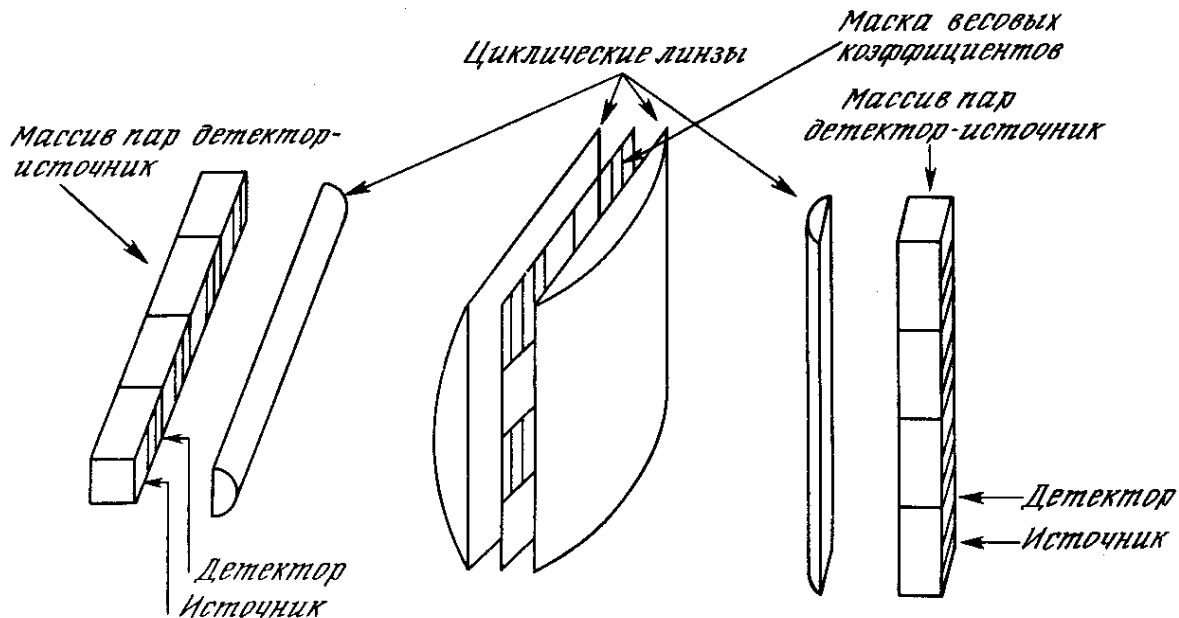


Рис. 9.2. Электронно-оптическая двунаправленная ассоциативная память

Каждый фотодетектор реагирует на световой поток от всей строки, его электронная часть реализует пороговую функцию и результирующий сигнал управляет связанным с ним световым источником. Тем самым замыкается петля обратной связи, включающая световые источники, фотодетекторы и оптическую систему. Заметим, что устойчивость ДАП гарантируется, даже если матрица не симметрична; кроме того, не требуется обязательного равенства нулю элементов главной диагонали.

**Массивы линейных модуляторов.** Линейный модулятор, недостаточно изученное до настоящего времени устройство, позволяет надеяться на существенное упрощение структуры электронно-оптических сетей. Как показано на рис. 9.3, линейный модулятор представляет собой тонкую пластину с чередующимися полосками светочувствительного материала и полосками оптических модуляторов. Прозрачность каждой полосы, соответствующей оптическому модулятору, может быть изменена электронным способом.

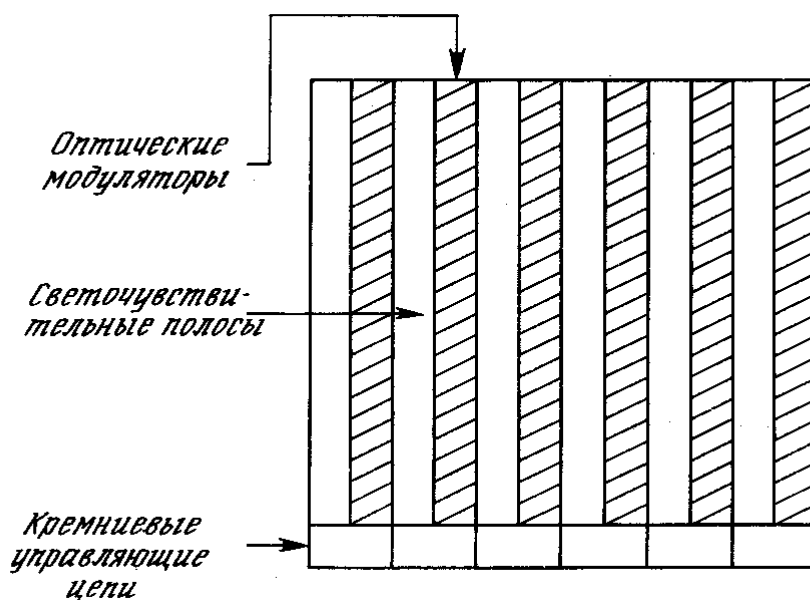


Рис. 9.3. Массив линейных пространственно-световых модуляторов

На рис. 9.4 показана упрощенная конструкция из линейных модуляторов, используемая в качестве оптического умножителя матриц. Горизонтальные полосы оптических модуляторов управляются электронным способом. Светопроводность

каждой полосе соответствует величине соответствующей компоненты входного вектора  $X$ , тем самым определяя величину светового потока через соответствующую строку весовой матрицы. В этой системе нет отдельных световых потоков для каждой световой строки; один источник света через коллиматор создает световой поток, входящий справа и проходящий через каждую полосу модулятора на весовую маску. Свет, проходящий через эту маску, попадает на вертикальные светочувствительные столбцы. Каждый столбец производит выход, пропорциональный суммарному световому потоку, проходящему через соответствующий столбец весовой маски. Таким образом, результат аналогичен описанному ранее для линзовой системы, концентрирующей свет на маленьком фотодетекторе; данная система производит умножение матриц с точно таким же результатом.

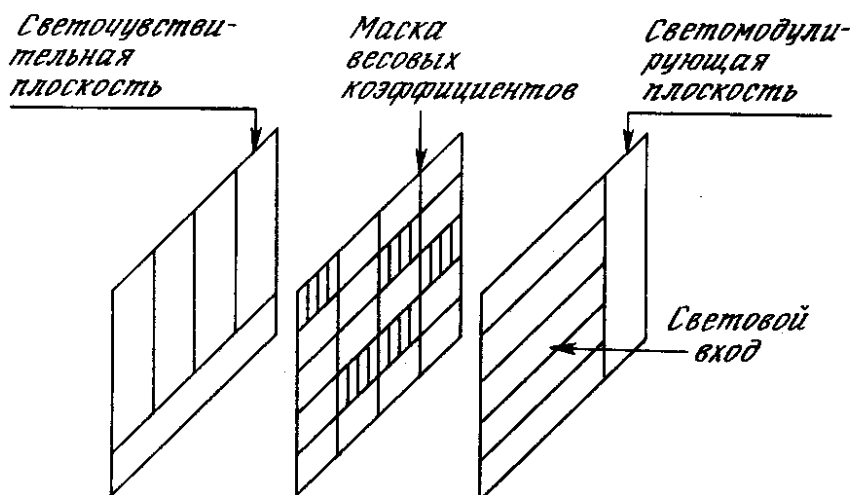
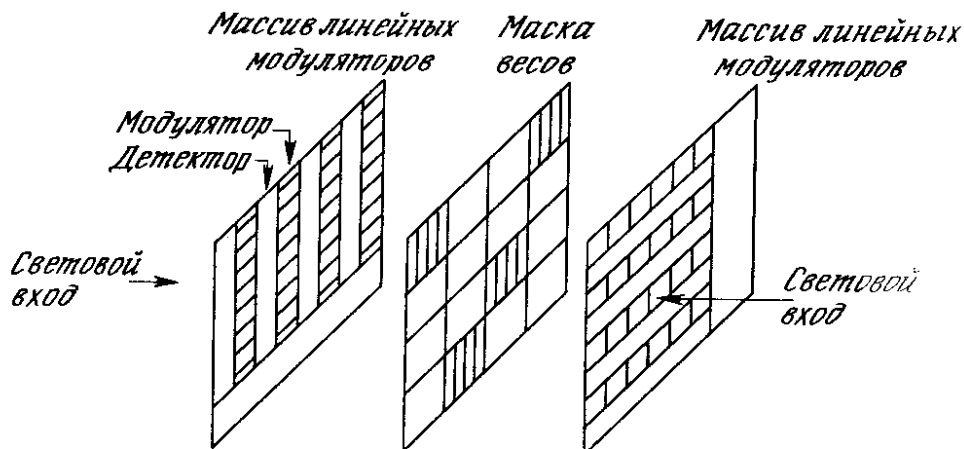


Рис. 9.4. Линейный модулятор, используемый в качестве оптического матричного умножителя.

Так как массивы линейных модуляторов передают свет, прошедший коллиматор, для данной системы не требуется цилиндрических линз. Это решает трудную проблему геометрических искажений, связанную с использовавшейся ранее оптикой. Преимущества компактной конструкции и оптической простоты, в то же время, приводят к относительно низкой скорости функционирования; современные технологии требуют десятков микросекунд для переключения световых модуляторов.

Реализация ДАП с использованием массивов линейных модуляторов. На рис. 9.5 приведена структура ДАП, сконструированной с использованием массивов линейных модуляторов. Она аналогична структуре описанного выше умножителя, за исключением того, что каждая полоса столбца светового детектора слева управляет пороговыми цепями, которые в свою очередь управляют светопроводностью связанной с ними вертикальной полосы. Таким образом, модулируется второй световой источник слева и соответствующий столбец весовой маски получает управляемый уровень освещенности. Это вырабатывает необходимый сигнал обратной связи для горизонтальных строк световых детекторов справа; их выходные сигналы обрабатываются пороговой функцией и управляют светопроводностью соответствующих горизонтальных светомодулирующих полос, тем самым замыкая петлю обратной связи ДАП.

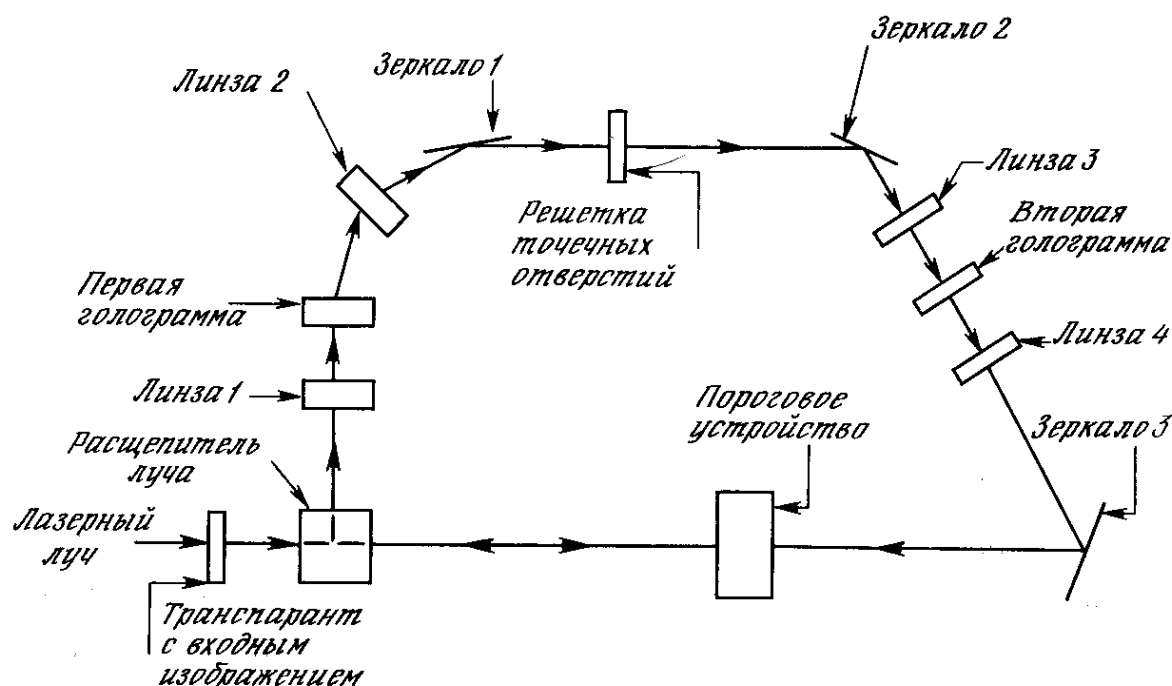


**Рис. 9.5. Оптическая двунаправленная ассоциативная память, использующая массивы линейных модуляторов.**

### ГОЛОГРАФИЧЕСКИЕ КОРРЕЛЯТОРЫ

Существует множество вариантов реализации голографических корреляторов и тем не менее их основные принципы функционирования очень схожи. Все они запоминают образцовые изображения в виде либо плоской, либо объемной голограммы и восстанавливают их при когерентном освещении в петле обратной связи. Входное изображение, которое может быть зашумленным или неполным, подается на вход системы и одновременно коррелируется оптически со всеми запомненными образцовыми изображениями. Эти корреляции обрабатываются пороговой функцией и подаются обратно на вход системы, где наиболее сильные корреляции усиливают (и, возможно, корректируют или завершают) входное изображение. Усиленное изображение проходит через систему многократно, изменяясь при каждом проходе до тех пор, пока система не стабилизируется на требуемом изображении. Заметим, что для описания распознаваемых образов использовался термин «изображение». Хотя распознавание изображений является наиболее адекватным приложением для оптических корреляторов, вход системы может рассматриваться как обобщенный вектор и система при этом становится общецелевой ассоциативной памятью.

Многие исследователи сделали большой вклад в развитие голографических корреляторов и лежащей в их основе теории. Например, в работах [2,4,8] проведены превосходные исследования. В работе [1] рассмотрена впечатляющая система, являющаяся основой следующего ниже обсуждения.



**Рис. 9.6. Оптическая система распознавания изображений**

В конфигурации, показанной на рис. 9.6, входом в систему является изображение, сформированное транспарантом, освещенным лазерным лучом. Это изображение через делитель луча передается на пороговое устройство, функции которого описаны ниже. Изображение отражается от порогового устройства, возвращается на делитель луча и затем попадает на линзу 1, которая фокусирует его на первой голограмме.

Первая голограмма содержит несколько запомненных изображений (например, изображения четырех самолетов). Входное изображение коррелируется с каждым из них, образуя световые образы. Яркость этих образов изменяется в зависимости от степени корреляции, определяющей сходство между двумя изображениями. Линза 2 и отражатель 1 проектируют изображение корреляций на микроканальный массив, где они пространственно разделяются. С микроканального массива множество световых образов передается на отражатель 2 через линзу 3 и затем прикладывается ко второй голограмме, которая имеет те же запомненные изображения, что и первая голограмма. Линза 4 и отражатель 3 затем передают суперпозицию множества коррелированных изображений на обратную сторону порогового устройства.

Пороговое устройство является ключевым для функционирования этой системы. Его передняя поверхность отражает наиболее сильно тот образ, который является самым ярким на его обратной поверхности. В данном случае на обратную поверхность проектируется набор из четырех корреляций каждого из четырех запомненных изображений с входным



изображением. Запомненное изображение, наиболее похожее на входное изображение, имеет самую высокую корреляцию, следовательно, оно будет самым ярким и наиболее сильно отражаемым от передней поверхности. Это усиленное отраженное изображение проходит через делитель луча, после чего повторно вводится в систему для дальнейшего усиления. В результате система будет сходиться к запомненному изображению, наиболее похожему на входной вектор. После этого можно убрать входной образ, и запомненный образ будет продолжать циркулировать в системе, производя выходное изображение, до сброса системы.

Записанная на видеоленту демонстрация этой системы показала ее способность восстанавливать полное изображение в случае, когда только часть изображения подается на вход системы. Это свойство имеет важное военное применение, так как распознавание цели часто должно быть выполнено в условиях частичной видимости. Кроме того, возможны многие другие промышленные применения, распознавание объектов как множества линий является задачей, решаемой на протяжении многих лет.

Несмотря на потенциальные возможности оптических корреляторов, качество изображения в существующих системах является невысоким, а их сложность и стоимость высоки. Кроме того, в настоящее время оптические корреляторы имеют большие размеры и трудны в наладке. Большие потенциальные возможности оптических корреляторов будут стимулировать проведение исследований по совершенствованию таких систем, однако в настоящее время многие вопросы остаются без ответа, несмотря на их практическое значение.

### **Объемные голограммы**

Некоторые кристаллы [8] искривляют характерный цветовой луч; величина искривления может модифицироваться лазером. Если сконструированы нейроны, способные получать и посылать свет, эти фоторефрактивные кристаллы могут использоваться для организации внутренних связей в больших сетях. В [11] исследована потенциальная плотность таких внутрисвязанных систем и приведена оценка, что практически могут быть реализованы сети с плотностью от  $10^8$  до  $10^{10}$  внутренних связей на кубический сантиметр.

Величина и направление, в котором луч искривляется фоторефрактивным кристаллом, определяется внутренней голографической решеткой, сформированной лазерным лучом высокой интенсивности. Локальный индекс рефракции кристалла является функцией локальной плотности его заряда. Лазер перераспределяет заряд путем смещения электронов, тем самым формируя области измененной силы рефракции. Если световой луч, соединяющий пару нейронов, попадает в соответствующую точку кристалла, он будет искривляться (реагировать) на соответствующий угол в направлении нейрона-приемника.

Более того, сила каждой решетки может управляться лазерным лучом, тем самым изменяя процентное соотношение рефразирующего луча. Это позволяет эффективно изменять веса внутренних связей в соответствии с обучающим алгоритмом.

### **Оптическая сеть Хопфилда, использующая объемные голограммы**

В работе [12] описана полностью оптическая рекуррентная нейронная сеть, сконструированная с использованием объемных голограмм. Сеть представляет собой оптическую реализацию сети Хопфилда, устанавливающую минимум на оптически сгенерированной энергетической поверхности. Когда предъявляется зашумленный или неполный входной образ, система сходится к наиболее похожему запомненному изображению, тем самым функционируя как оптическая ассоциативная память.

Рис. 9.7 представляет упрощенную конфигурацию системы. Резонансная петля включает массив оптических нейронов, оптическую матрицу внутренних связей и соответствующие оптические компоненты. Изображения (предъявляемые как векторы) проходят через контур с обратной связью в определяемом массивами направлении, усиливаясь в процессе обработки. Это является точной аналогией функционирования сети Хопфилда. Оптический массив нейронов суммирует входные сигналы и сигналы обратной связи и затем реализует сигмоидальную функцию активации, оптическая матрица внутренних связей выполняет векторно-матричное умножение.

Когда входной вектор (возможно представляющий собой изображение) прикладывается справа, он попадает через делитель луча  $BS_2$  на массив оптических нейронов. Здесь он усиливается, и с помощью насыщающегося двухлучевого усилителя вычисляется сигмоидальная функция. Сжатый выходной вектор частично отражается делителем луча  $BS_1$  на линзу  $L_1$  и затем вводится в оптическую матрицу внутренних связей. Часть выходного светового потока проходит через  $BS_1$  и образует выход системы.

Оптическая матрица внутренних связей состоит из двух объемных голограмм, которые хранят образцовые изображения в виде записанных лазерными лучами дифракционных образов. Они служат весами входных компонент и направляют каждую взвешенную сумму на соответствующий элемент оптического выходного вектора.

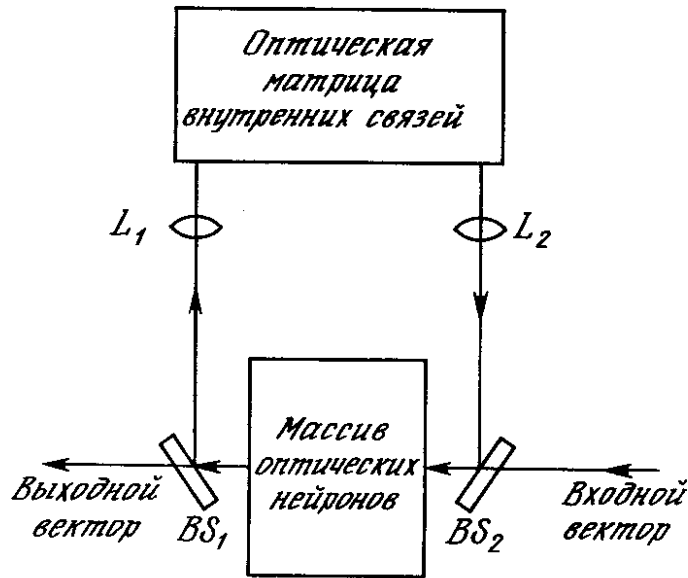


Рис. 9.7. Оптическая сеть Хопфилда

**Оптический нейрон.** На рис. 9.8 показана конструкция типичного элемента массива оптических нейронов. Он функционирует как оптически накачивающий двухлучевой насыщающий усилитель в кристалле  $\text{BaTiO}_3$ . Лазерный накачивающий луч, приложенный под углом  $\theta$ , взаимодействует с входным лучом для выработки усиленной копии входного сигнала с последующим вычислением сигмоидальной функции активации, аналогичной показанной на рис. 9.9. С использованием этой техники было достигнуто оптическое усиление приблизительно в 60 раз. Заметим, что на рис. 9.9 угол  $\varphi$  между входным лучом и линией оси кристалла  $C$  критичен для правильного функционирования этого устройства.

**Оптическая матрица внутренних связей.** В оптической матрице внутренних связей выходной сигнал массива оптических нейронов попадает в оптическую систему, содержащую две объемные голограммы. Оптическое преобразование Фурье входного сигнала производится с использованием стандартной оптической техники Фурье. Затем сигнал поступает на первую объемную голограмму, в которой хранятся образцовые векторы в фазокодированном пространстве Фурье. Выход этой голограммы поступает на вход двухлучевого оптического усилителя, аналогичного усилителю оптического нейрона, но работающему в ненасыщенном режиме. В результате усиление поднимается до уровня, в котором возможна циклическая регенерация. Затем оптически выполняется обратное преобразование Фурье усиленного сигнала и результат подается на вторую объемную голограмму, в которой хранятся те же образцовые изображения, но на этот раз в объектном пространстве (а не в фазокодированном пространстве Фурье). Выходом системы является суперпозиция векторно-матричных произведений входного вектора и запомненных образцовых векторов. Этот оптический образ вырабатывается оптической матрицей внутренних связей и прикладывается к массиву оптических нейронов, замыкая контур обратной связи.

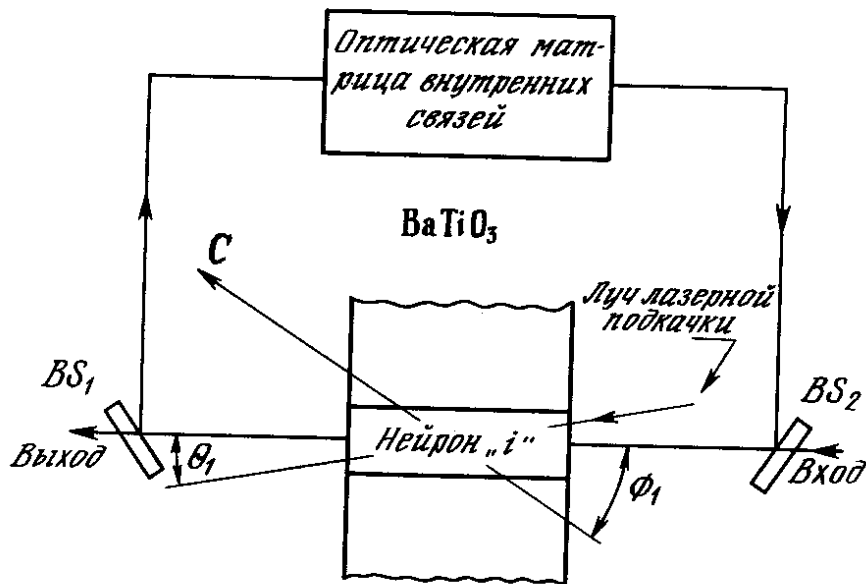


Рис. 9.8. Массив оптических нейронов

Обсуждение оптической матрицы внутренних связей здесь сильно упрощено; детали реализации включают сложную оптическую технику функционирования, выходящую за рамки данной работы. Интересующиеся читатели отсылаются к работам [12] и [5].

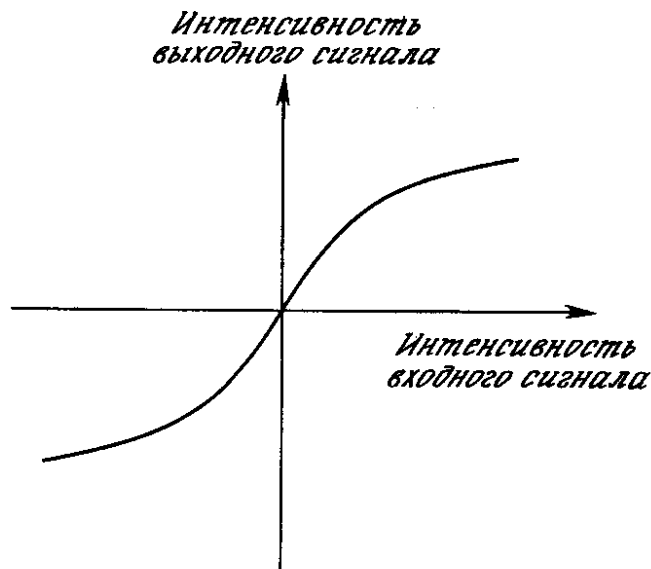


Рис. 9.9. Сигмоидальная функция активации

## ЗАКЛЮЧЕНИЕ

Оптические нейронные сети предлагают огромные выгоды с точки зрения скорости и плотности внутренних связей. Они могут быть использованы (в той или иной форме) для реализации сетей фактически с любой архитектурой.

В настоящее время ограничения электронно-оптических устройств создают множество серьезных проблем, которые должны быть решены прежде, чем оптические нейронные сети получат широкое применение. Однако учитывая, что большое количество превосходных исследователей работает над этой проблемой, а также большую поддержку со стороны военных, можно надеяться на быстрый прогресс в этой области.

## Литература

1. Abu-Mostafa Y. S., Psaltis D. 1987. Optical neural computers. Scientific American, March, pp. 88-95.
2. Anderson D. Z. 1985. Coherent optical Eigenstate memory. Proceeding of the Optical Society of America 1985 Annual Meeting.
3. Athale R. A., Friedlander C. B., Kushner C. B. 1986. Attentive associative architectures and their implications to optical computing. Proceedings of the Society of Photo-Optical Instrumentation Engineering 625:179-88
4. Dunning G. J., Marom E., Owechko Y., Soffer B. N. 1985. All-optical associative holographic memory with feedback using phase conjugate mirrors. Proceedings of the Society of Photo-Optical Instrumentation Engineering 625:179-188.
5. Fainman N. H., Klancnik E., Lee S. H. 1968. Optical Engineering 25:228.
6. Farhat N. H., Psaltis D., Prata A., Paek E. 1985. Optical implementation of the Hopfield model. Applied optics 24:1469-75
7. Fisher A. D., Giles C. L., Lee J. N. 1985. An adaptive optical computing element. Proceedings of the Optical Society of America Topical Meeting.
8. Jansson T., Karagaleff C., Stoll K. M. 1986. Photo-refractive LiNbO<sub>3</sub> as a storage medium for high-density optical neural networks. 1986 Optical Society of America Annual Meeting.
9. Kosko B. 1987. Optical bidirectional associative memories. Proceedings of the Society of Photo-Optical Instrumentation Engineering: Image Understanding and the Man-Machine Interface 758:11-18.
10. Mead C. 1988. Paper presented during plenary session. IEEE Second International Conference on Neural Networks. San Diego, June.
11. Psaltis D., Wagner K., Brady D. 1987 Learning in optical neural computers. In Proceedings of IEEE First International Conference on Neural Networks, edc. M. Caudill and C. Butler. San Diego, CA:SOS Printing.
12. Stoll H. M., Lee L. S. 1988. Continuous time optical neural networks. Proceedings of IEEE International Conference on Neural Networks. San Diego, CA:SOS Printing.

## Глава 10.

### Когнитрон и неокогнитрон

Люди решают сложные задачи распознавания образов с обескураживающей легкостью. Двухлетний ребенок без видимых усилий различает тысячи лиц и других объектов, составляющих его окружение, несмотря на изменение расстояния, поворота, перспективы и освещения.

Может показаться, что изучение этих врожденных способностей должно сделать простой задачей разработку компьютера, повторяющего способности человека к распознаванию. Ничто не может быть более далеким от истины. Сходство и различия образов, являющиеся очевидными для человека, пока ставят в тупик даже наиболее сложные компьютерные системы распознавания. Таким образом, бесчисленное количество важных приложений, в которых компьютеры могут заменить людей в опасных, скучных или неприятных работах, остаются за пределами их текущих возможностей.

Компьютерное распознавание образов является больше искусством; наука ограничена наличием нескольких методик, имеющих относительно небольшое использование на практике. Инженер, конструирующий типовую систему распознавания образов, обычно начинает с распознавания печатного текста. Эти методы часто являются неадекватными задаче, и старания разработчиков быстро сводятся к разработке алгоритмов, узко специфичных для данной задачи.

Обычно целью конструирования систем распознавания образов является оптимизация ее функционирования над выборочным набором образов. Очень часто разработчик завершает эту задачу нахождением нового, приблизительно похожего образа, что приводит к неудачному завершению алгоритмов. Этот процесс может продолжаться неопределенно долго, никогда не приводя к устойчивому решению, достаточному для повторения процесса восприятия человека, оценивающего качество функционирования системы.

К счастью, мы имеем существующее доказательство того, что задача может быть решена: это система восприятия человека. Учитывая ограниченность успехов, достигнутых в результате стремления к собственным изобретениям, кажется вполне логичным вернуться к биологическим моделям и попытаться определить, каким образом они функционируют так хорошо. Очевидно, что это трудно сделать по нескольким причинам. Во-первых, сверхвысокая сложность человеческого мозга затрудняет понимание принципов его функционирования. Трудно понять общие принципы функционирования и взаимодействия его приблизительно  $10^{11}$  нейронов и  $10^{14}$  синаптических связей. Кроме того, существует множество проблем при проведении экспериментальных исследований. Микроскопические исследования требуют тщательно подготовленных образцов (заморозка, срезы, окраска) для получения маленького двумерного взгляда на большую трехмерную структуру. Техника микропроб позволяет провести исследования внутренней электрохимии узлов, однако трудно контролировать одновременно большое количество узлов и наблюдать их взаимодействие. Наконец, этические соображения запрещают многие важные эксперименты, которые могут быть выполнены только на людях. Большое значение имели эксперименты над животными, однако животные не обладают способностями человека описывать свои впечатления.

Несмотря на эти ограничения, многое было изучено благодаря блестяще задуманным экспериментам. Например, в [1] описан эксперимент, в котором котят выращивали в визуальном окружении, состоящем только из горизонтальных черных и белых полос. Известно, что определенные области коры чувствительны к углу ориентации, поэтому у этих котят не развились нейроны, чувствительные к вертикальным полосам. Этот результат наводит на мысль, что мозг млекопитающих не является полностью «предустановленным» даже на примитивном уровне распознавания ориентации линий. Напротив, он постоянно самоорганизуется, основываясь на опыте.

На микроскопическом уровне обнаружено, что нейроны обладают как возбуждающими, так и тормозящими синапсами. Первые стремятся к возбуждению нейрона; последние подавляют его возбуждение (см. приложение А). Это наводит на мысль, что мозг адаптируется либо изменением воздействия этих синапсов, либо созданием или разрушением синапсов в результате воздействия окружающей среды. Данное предположение остается пока гипотезой с ограниченным физиологическим подтверждением. Однако исследования, проведенные в рамках этой гипотезы, привели к созданию цифровых моделей, некоторые из которых показывают замечательные способности к адаптивному распознаванию образов.

## КОГНИТРОН

Основываясь на текущих знаниях анатомии и физиологии мозга, в работе [2] разработан когнитрон, гипотетическая модель системы восприятия человека. Компьютерные модели, исследованные в [2], продемонстрировали впечатляющие способности адаптивного распознавания образов, побуждая физиологов исследовать соответствующие механизмы мозга. Это взаимно усиливающее взаимодействие между искусственными нейронными сетями, физиологией и психологией может оказаться средством, посредством которого будет со временем достигнуто понимание механизмов мозга.

### Структура

Когнитрон конструируется в виде слоев нейронов, соединенных синапсами. Как показано на рис. 10.1, пресинаптический нейрон в одном слое связан с постсинаптическим нейроном в следующем слое. Имеются два типа нейронов: возбуждающие узлы, которые стремятся вызвать возбуждение постсинаптического узла, и тормозящие узлы, которые тормозят это возбуждение. Возбуждение нейрона определяется взвешенной суммой его возбуждающих и тормозящих входов, однако в действительности механизм является более сложным, чем простое суммирование.

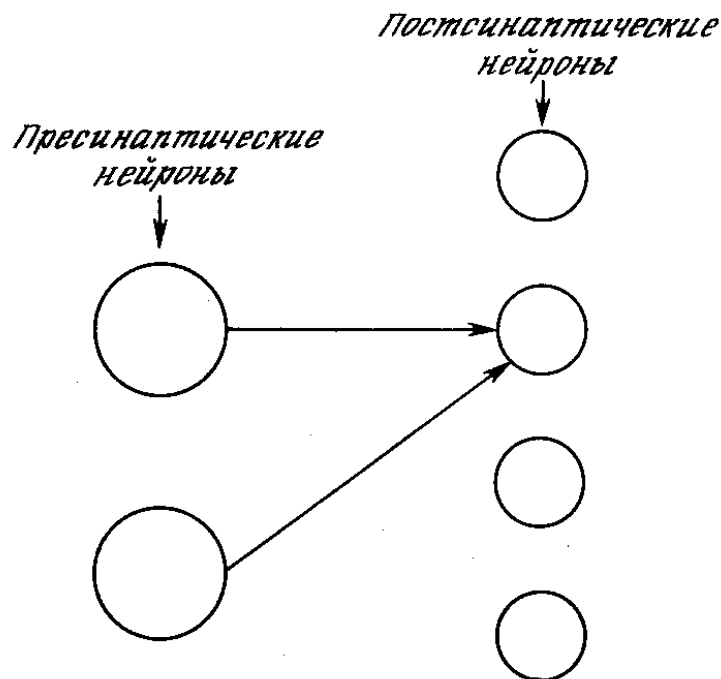


Рис. 10.1. Пресинаптические и постсинаптические нейроны

На рис. 10.2 показано, что каждый нейрон связан только с нейронами в соседней области, называемой *областью связи*. Это ограничение области связи согласуется с анатомией зрительной коры, в которой редко соединяются между собой нейроны, располагающиеся друг от друга на расстоянии более одного миллиметра. В рассматриваемой модели нейроны упорядочены в виде

слоев со связями от одного слоя к следующему. Это также аналогично послойной структуре зрительной коры и других частей головного мозга.

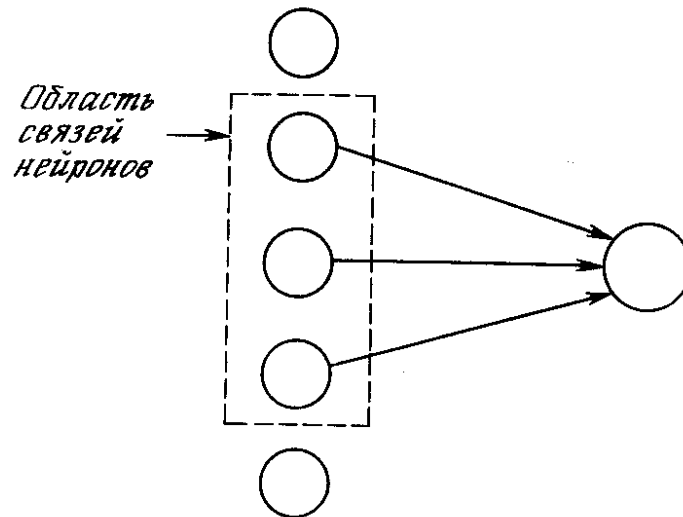


Рис. 10.2. Область связей нейрона

### Обучение

Так как когнитрон реализован в виде многослойной сети, возникают сложные проблемы обучения, связанные с выбранной структурой. Автор отверг управляемое обучение, как биологически неправдоподобное, используя взамен этого обучение без учителя. Получая обучающий набор входных образов, сеть самоорганизуется посредством изменения силы синаптических связей. При этом отсутствуют предварительно определенные выходные образы, представляющие требуемую реакцию сети, однако сеть самоорганизуется с целью распознавания входных образов с замечательной точностью.

Алгоритм обучения когнитрона является концептуально привлекательным. В заданной области слоя обучается только наиболее сильно возбужденный нейрон. Автор сравнивает это с «элитным обучением», при котором обучаются только «умные» элементы. Те нейроны, которые уже хорошо обучены, что выражается силой их возбуждения, получают приращение силы своих синапсов с целью дальнейшего усиления своего возбуждения.

На рис. 10.3 показано, что области связи соседних узлов значительно перекрываются. Это расточительное дублирование функций оправдывается взаимной конкуренцией между ближайшими узлами. Даже если узлы в начальный момент имеют абсолютно идентичный выход, небольшие отклонения всегда имеют место; один из узлов всегда будет иметь более сильную реакцию на входной образ, чем соседние. Его сильное возбуждение будет оказывать сдерживающее воздействие на возбуждение соседних узлов, и только его синапсы будут усиливаться; синапсы соседних узлов останутся неизменными.

**Возбуждающий нейрон.** Грубо говоря, выход возбуждающего нейрона в когнитроне определяется отношением его возбуждающих входов к тормозящим входам. Эта необычная функция имеет важные преимущества, как практические, так и теоретические.

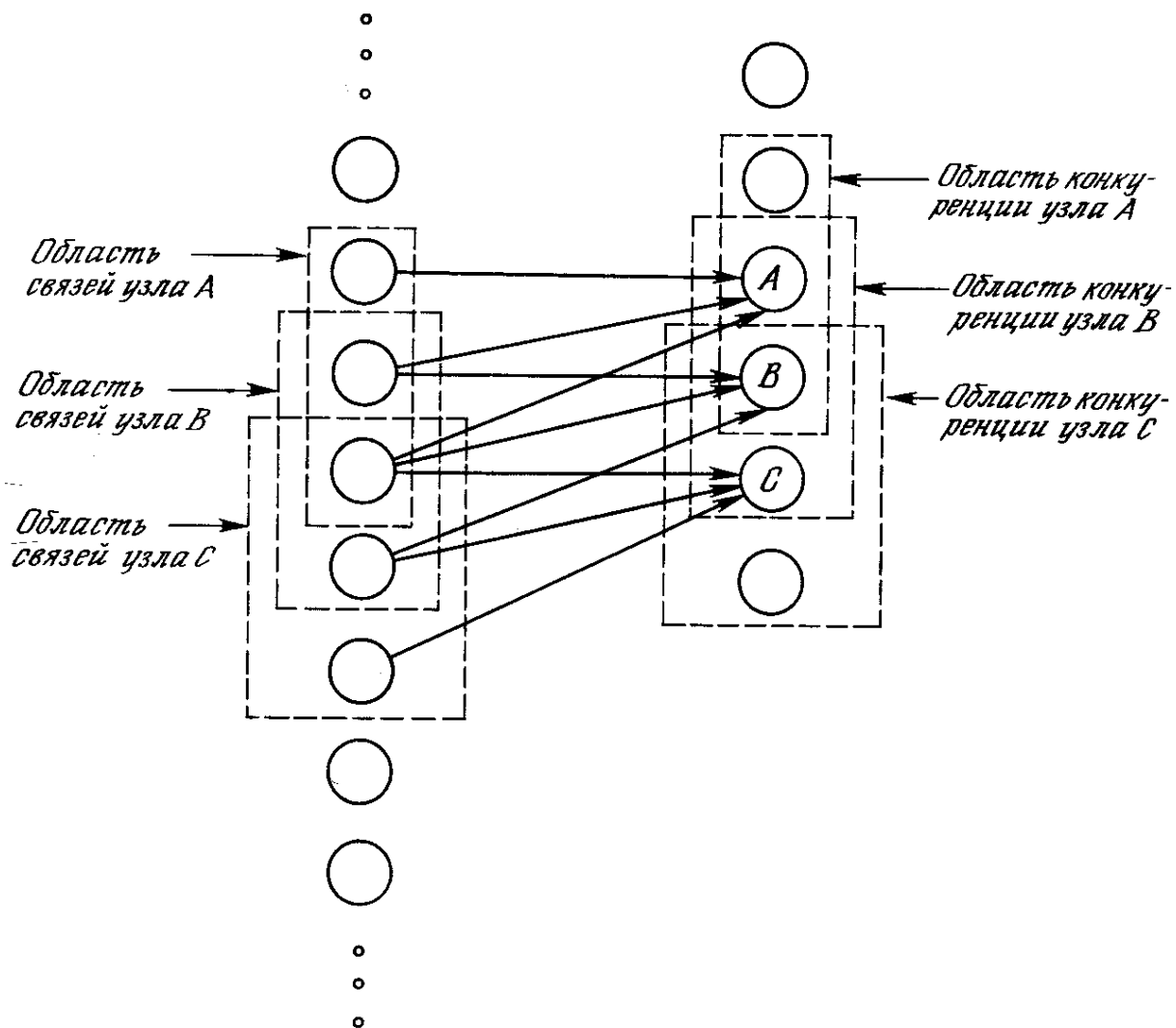


Рис. 10.3. Область связи с областью конкуренции

Суммарный возбуждающий вход в нейрон взвешенной суммой входов от возбуждающих предшествующем слое. Аналогично суммарный вход / является взвешенной суммой входов от всех тормозящих нейронов. В символическом виде

$$E = \sum_i a_i u_i, \quad I = \sum_j b_j v_j,$$

где  $a_i$  – вес  $i$ -го возбуждающего синапса,  $u_i$  – выход  $i$ -го возбуждающего нейрона,  $b_j$  – вес  $j$ -го тормозящего синапса,  $v_j$  – выход  $j$ -го тормозящего нейрона.

Заметим, что веса имеют только положительные значения. Выход нейрона затем вычисляется следующим образом:

$$\text{NET} = \frac{1 + E}{1 + I} - 1$$

$$\text{OUT} = \text{NET}, \quad \text{при } \text{NET} \geq 0,$$

$$\text{OUT} = 0, \quad \text{при } \text{NET} < 0.$$

Предполагая, что NET имеет положительное значение, это можно записать следующим образом:

$$\text{OUT} = \frac{E - I}{1 + I}$$

Когда тормозящий вход мал ( $I \ll 1$ ), OUT может быть аппроксимировано как

$$\text{OUT} = E - I,$$

что соответствует выражению для обычного линейного порогового элемента (с нулевым порогом).

Алгоритм обучения когнитрона позволяет весам синапсов возрастать без ограничений. Благодаря отсутствию механизма уменьшения весов они просто возрастают в процессе обучения. В обычных линейных пороговых элементах это привело бы к произвольно большому выходу элемента. В когнитроне большие возбуждающие и тормозящие входы resultируются в ограничивающей формуле вида:

$$\text{OUT} = \frac{E}{I} - 1, \text{ если } E \gg 1 \text{ и } I \gg 1.$$

В данном случае OUT определяется отношением возбуждающих входов к тормозящим входам, а не их разностью. Таким образом, величина OUT ограничивается, если оба входа возрастают в одном и том же диапазоне  $X$ . Предположив, что это так,  $E$  и  $I$  можно выразить следующим образом:

$$E = pX, I = qX, p, q - \text{константы,}$$

и после некоторых преобразований

$$\text{OUT} = \frac{p - q}{2q} \cdot \left[ 1 + \text{th} \left( \frac{\log(pq)}{2} \right) \right].$$

Эта функция возрастает по закону Вебера-Фехнера, который часто используется в нейрофизиологии для аппроксимации нелинейных соотношений входа/выхода сенсорных нейронов. При использовании этого соотношения нейрон когнитрона в точности эмулирует реакцию биологических нейронов. Это делает его как мощным вычислительным элементом, так и точной моделью для физиологического моделирования.

**Тормозящие нейроны.** В когнитроне слой состоит из возбуждающих и тормозящих узлов. Как показано на рис. 10.4, нейрон слоя 2 имеет область связи, для которой он имеет синаптические соединения с набором выходов нейронов в слое 1. Аналогично в слое 1 существует тормозящий нейрон, имеющий ту же область связи. Синаптические веса тормозящих узлов не изменяются в процессе обучения; их веса заранее установлены таким образом, что сумма весов в любом из тормозящих нейронов равна единице. В соответствии с этими ограничениями, выход тормозящего узла INHIB является взвешенной суммой его входов, которые в данном случае представляют собой среднее арифметическое выходов возбуждающих нейронов, к которым он подсоединен. Таким образом,



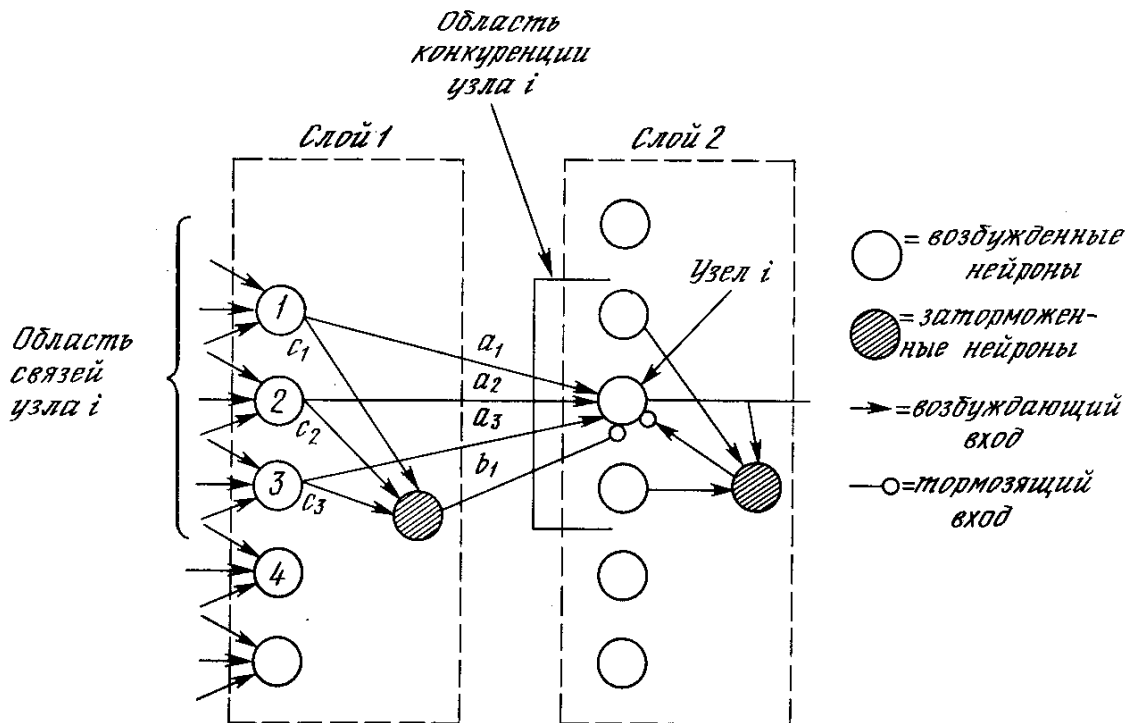


Рис. 10.4. Слой когнитрона

$$\text{INHIB} = \sum_i c_i \text{OUT}_i,$$

где  $\sum_i c_i = 1$ ,  $c_i$  – возбуждающий вес  $i$ .

**Процедура обучения.** Как объяснялось ранее, веса возбуждающих нейронов изменяются только тогда, когда нейрон возбужден сильнее, чем любой из узлов в области конкуренции. Если это так, изменение в процессе обучения любого из его весов может быть определено следующим образом:

$$\delta a_i = q c_j u_j,$$

где  $c_j$  – тормозящий вес связи нейрона  $j$  в слое 1 с тормозящим нейроном  $i$ ,  $u_j$  – выход нейрона  $j$  в слое 1,  $a_i$  – возбуждающий вес  $i$ ,  $q$  – нормирующий коэффициент обучения.

Изменение тормозящих весов нейрона  $i$  в слое 2 пропорционально отношению взвешенной суммы возбуждающих входов к удвоенному тормозящему входу. Вычисления проводятся по формуле

$$\delta b_i = \frac{q \sum_j a_j u_j}{2 \cdot \text{INHIB}_i}.$$

Когда возбужденных нейронов в области конкуренции нет, для изменения весов используются другие выражения. Это необходимо, поскольку процесс обучения начинается с нулевыми значениями весов; поэтому первоначально нет возбужденных нейронов ни в одной области конкуренции, и обучение производится не может. Во всех случаях, когда победителя в области конкуренции нейронов нет, изменение весов нейронов вычисляется следующим образом:

$$\Delta a_i = q^* c_j u_j, \quad \delta b_i = q^* \text{INHIB}_i,$$

где  $q^*$  – положительный обучающий коэффициент меньший, чем  $q$ .

Приведенная стратегия настройки гарантирует, что узлы с большой реакцией заставляют возбуждающие синапсы, которыми они управляют, увеличиваться сильнее, чем тормозящие синапсы. И наоборот, узлы, имеющие малую реакцию, вызывают малое возрастание возбуждающих синапсов, но большее возрастание тормозящих синапсов. Таким образом, если узел 1 в слое 1 имеет большой выход, синапс  $a_1$  возрастет больше, чем синапс  $b_1$ . И наоборот, узлы, имеющие малый выход, обеспечат малую величину для приращения  $a_i$ . Однако другие узлы в области связи будут возбуждаться, тем самым увеличивая сигнал INHIB и значения  $b_i$ .

В процессе обучения веса каждого узла в слое 2 настраиваются таким образом, что вместе они составляют шаблон, соответствующий образам, которые часто предъявляются в процессе обучения. При предъявлении сходного образа шаблон соответствует ему и узел вырабатывает большой выходной сигнал. Сильно отличающийся образ вырабатывает малый выход и обычно подавляется конкуренцией.

**Латеральное торможение.** На рис. 10.4 показано, что каждый нейрон слоя 2 получает латеральное торможение от нейронов, расположенных в его области конкуренции. Тормозящий нейрон суммирует входы от всех нейронов в области конкуренции и вырабатывает сигнал, стремящийся к торможению целевого нейрона. Этот метод является эффективным, но с вычислительной точки зрения медленным. Он охватывает большую систему с обратной связью, включающую каждый нейрон в слое; для его стабилизации может потребоваться большое количество вычислительных итераций.

Для ускорения вычислений в работе [2] используется остроумный метод ускоренного латерального торможения (рис. 10.5). Здесь дополнительный узел латерального торможения обрабатывает выход каждого возбуждающего узла для моделирования требуемого латерального торможения. Сначала он определяет сигнал, равный суммарному тормозящему влиянию в области конкуренции:

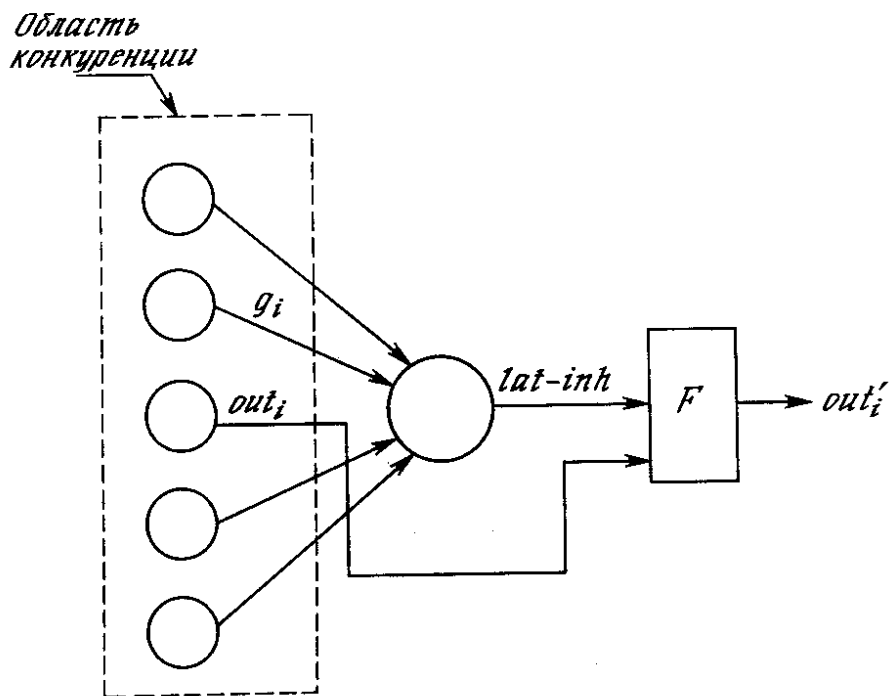


Рис. 10.5. Ускоренное торможение

$$\text{LAT\_INHIB} = \sum_i g_i \text{OUT}_i,$$

где  $\text{OUT}_i$  – выход  $i$ -го нейрона в области конкуренции,  $g_i$  – вес связи от этого нейрона к латерально-тормозящему нейрону;  $g_i$  выбраны таким образом, что  $\sum_i g_i = 1$ .

Выход тормозящего нейрона  $\text{OUT}'$  затем вычисляется следующим образом:

$$OUT' = \frac{1 + OUT_i}{1 + LAT\_INHIB} - 1$$

Благодаря тому что все вычисления, связанные с таким типом латерального торможения, являются нерекурсивными, они могут быть проведены за один проход для слоя, тем самым определяя эффект в виде большой экономии в вычислениях.

Этот метод латерального торможения решает и другую сложную проблему. Предположим, что узел в слое 2 возбуждается сильно, но возбуждение соседних узлов уменьшается постепенно с увеличением расстояния. При использовании обычного латерального торможения будет обучаться только центральный узел. Другие узлы определяют, что центральный узел в их области конкуренции имеет более высокий выход. С предлагаемой системой латерального торможения такой ситуации случиться не может. Множество узлов может обучаться одновременно и процесс обучения является более достоверным.

**Рецептивная область.** Анализ, проводимый до этого момента, был упрощен рассмотрением только одномерных слоев. В действительности когнитрон конструировался как каскад двумерных слоев, причем в данном слое каждый нейрон получает входы от набора нейронов на части двумерного плана, составляющей его область связи в предыдущем слое.

С этой точки зрения когнитрон организован подобно зрительной коре человека, представляющей собой трехмерную структуру, состоящую из нескольких различных слоев. Оказывается, что каждый слой мозга реализует различные уровни обобщения; входной слой чувствителен к простым образам, таким, как линии, и их ориентации в определенных областях визуальной области, в то время как реакция других слоев является более сложной, абстрактной и независимой от позиции образа.

Аналогичные функции реализованы в когнитроне путем моделирования организации зрительной коры. На рис. 10.6 показано, что нейроны когнитрона в слое 2 реагируют на определенную небольшую область входного слоя 1. Нейрон в слое 3 связан с набором нейронов слоя 2, тем самым реагируя косвенно на более широкий набор нейронов слоя 1. Подобным образом нейроны в последующих слоях чувствительны к более широким областям входного образа до тех пор, пока в выходном слое каждый нейрон не станет реагировать на все входное поле.

Если область связи нейронов имеет постоянный размер во всех слоях, требуется большое количество слоев для перекрытия всего входного поля выходными нейронами. Количество слоев может быть уменьшено путем расширения области связи в последующих слоях. К сожалению, результатом этого может явиться настолько большое перекрытие областей связи, что нейроны выходного слоя будут иметь одинаковую реакцию. Для решения этой проблемы может быть использовано расширение области конкуренции. Так как в данной области конкуренции может возбудиться только один узел, влияние малой разницы в реакциях нейронов выходного слоя усиливается.

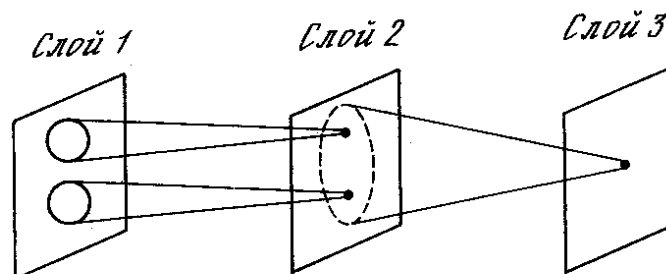


Рис. 10.6. Области связей когнитрона

В альтернативном варианте связи с предыдущим слоем могут быть распределены вероятностно с большинством синаптических связей в ограниченной области и с более длинными соединениями, встречающимися намного реже. Это отражает вероятностное распределение нейронов, обнаруженное в мозге. В когнитроне это позволяет каждому нейрону выходного слоя реагировать на полное входное поле при наличии ограниченного количества слоев.

**Результаты моделирования.** В [4] описываются результаты компьютерного моделирования четырехслойного когнитрона, предназначенного для целей распознавания образов. Каждый слой состоит из массива 12x12 возбуждающих нейронов и такого же количества тормозящих нейронов. Область связи представляет собой квадрат, включающий 5x5 нейронов. Область конкуренции имеет форму ромба высотой и шириной в пять нейронов. Латеральное торможение охватывает область 7x7 нейронов. Нормирующие параметры обучения установлены таким образом, что  $q=16,0$  и  $q'=2,0$ . Веса синапсов проинициализированы в 0.

Сеть обучалась путем предъявления пяти стимулирующих образов, представляющих собой изображения арабских цифр от 0 до 4, на входном слое. Веса сети настраивались после предъявления каждой цифры, входной набор подавался на вход сети циклически до тех пор, пока каждый образ не был предъявлен суммарно 20 раз.

Эффективность процесса обучения оценивалась путем запуска сети в реверсивном режиме; выходные образы, являющиеся реакцией сети, подавались на выходные нейроны и распространялись обратно к входному слою. Образы, полученные во входном слое, затем сравнивались с исходным входным образом. Чтобы сделать это, обычные однонаправленные связи принимались проводящими в обратном направлении и латеральное торможение отключалось. На рис. 10.7 показаны типичные результаты тестирования. В столбце 2 показаны образы, произведенные каждой цифрой на выходе сети. Эти образы возвращались обратно, вырабатывая на входе сети образ, близкий к точной копии исходного входного образа. Для столбца 4 на выход сети подавался только выход нейрона, имеющего максимальное возбуждение. Результирующие образы в точности те же, что и в случае подачи полного выходного образа, за исключением цифры 0, для которой узел с максимальным выходом располагался на периферии и не покрывал полностью входного поля.

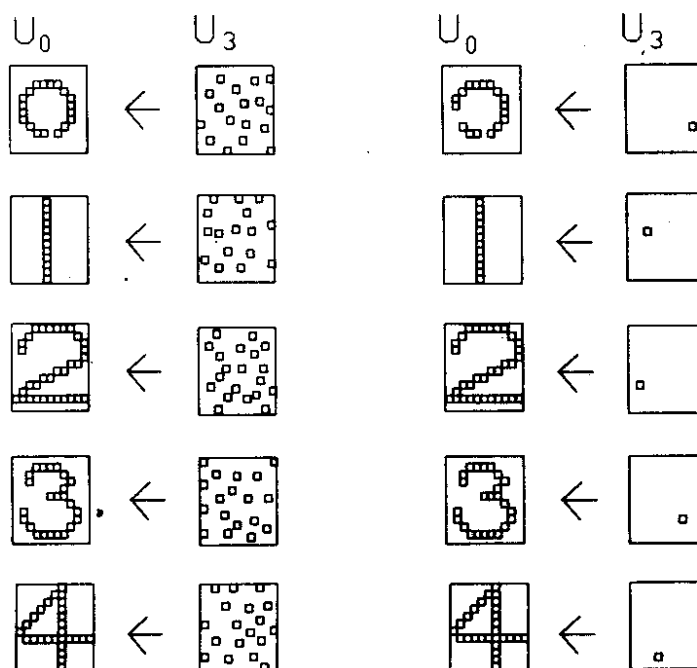


Рис. 10.7. Результаты экспериментов с когнитроном

## НЕОКОГНИТРОН

В попытках улучшить когнитрон была разработана мощная парадигма, названная *неокогнитрон* [5–7]. В то время как когнитрон и неокогнитрон имеют определенное сходство, между ними также существуют фундаментальные различия, связанные с эволюцией исследований авторов. Оба образца являются многоуровневыми иерархическими сетями, организованными аналогично зрительной коре. В то же время неокогнитрон более соответствует модели зрительной системы, предложенной в работах [10–12]. В результате неокогнитрон является намного более мощной парадигмой с точки зрения способности распознавать образы независимо от их преобразований, вращении, искажений и изменений масштаба. Как и когнитрон, неокогнитрон использует самоорганизацию в процессе обучения, хотя была описана версия [9], в которой вместо этого использовалось управляемое обучение.

Неокогнитрон ориентирован на моделирование зрительной системы человека. Он получает на входе двумерные образы, аналогичные изображениям на сетчатой оболочке глаза, и обрабатывает их в последующих слоях аналогично тому, как это было обнаружено в зрительной коре человека. Конечно, в неокогнитроне нет ничего, ограничивающего его использование только для обработки визуальных данных, он достаточно универсален и может найти широкое применение как обобщенная система распознавания образов.

В зрительной коре были обнаружены узлы, реагирующие на такие элементы, как линии и углы определенной ориентации. На более высоких уровнях узлы реагируют на более сложные и абстрактные образы такие, как окружности, треугольники и прямоугольники. На еще более высоких уровнях степень абстракции возрастает до тех пор, пока не определяются узлы, реагирующие на лица и сложные формы. В общем случае узлы на более высоких уровнях получают вход от группы низкоуровневых узлов и, следовательно, реагируют на более широкую область визуального поля. Реакции узлов более высокого уровня менее зависят от позиции и более устойчивы к искажениям.

## Структура

Неокогнитрон имеет иерархическую структуру, ориентированную на моделирование зрительной системы человека. Он состоит из последовательности обрабатывающих слоев, организованных в иерархическую структуру (рис. 10.8). Входной образ подается на первый слой и передается через плоскости, соответствующие последующим слоям, до тех пор, пока не достигнет выходного слоя, в котором идентифицируется распознаваемый образ.

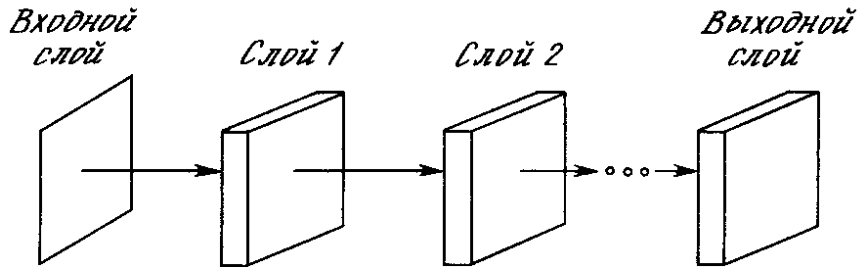


Рис. 10.8. Структура слоев неокогнитрона

Структура неокогнитрона трудна для представления в виде диаграммы, но концептуально проста. Чтобы подчеркнуть его многоуровневость (с целью упрощения графического представления), используется анализ верхнего уровня. Неокогнитрон показан состоящим из слоев, слои состоят из набора плоскостей и плоскости состоят из узлов.

**Слои.** Каждый слой неокогнитрона состоит из двух массивов плоскостей (рис. 10.9). Массив плоскостей, содержащих простые узлы, получает выходы предыдущего слоя, выделяет определенные образы и затем передает их в массив плоскостей, содержащих комплексные узлы, где они обрабатываются таким образом, чтобы сделать выделенные образы менее позиционно зависимыми.

**Плоскости.** Внутри слоя плоскости простых и комплексных узлов существуют парами, т. е. для плоскости простых узлов существует одна плоскость комплексных узлов, обрабатывающая ее выходы. Каждая плоскость может быть визуально представлена как двумерный массив узлов.

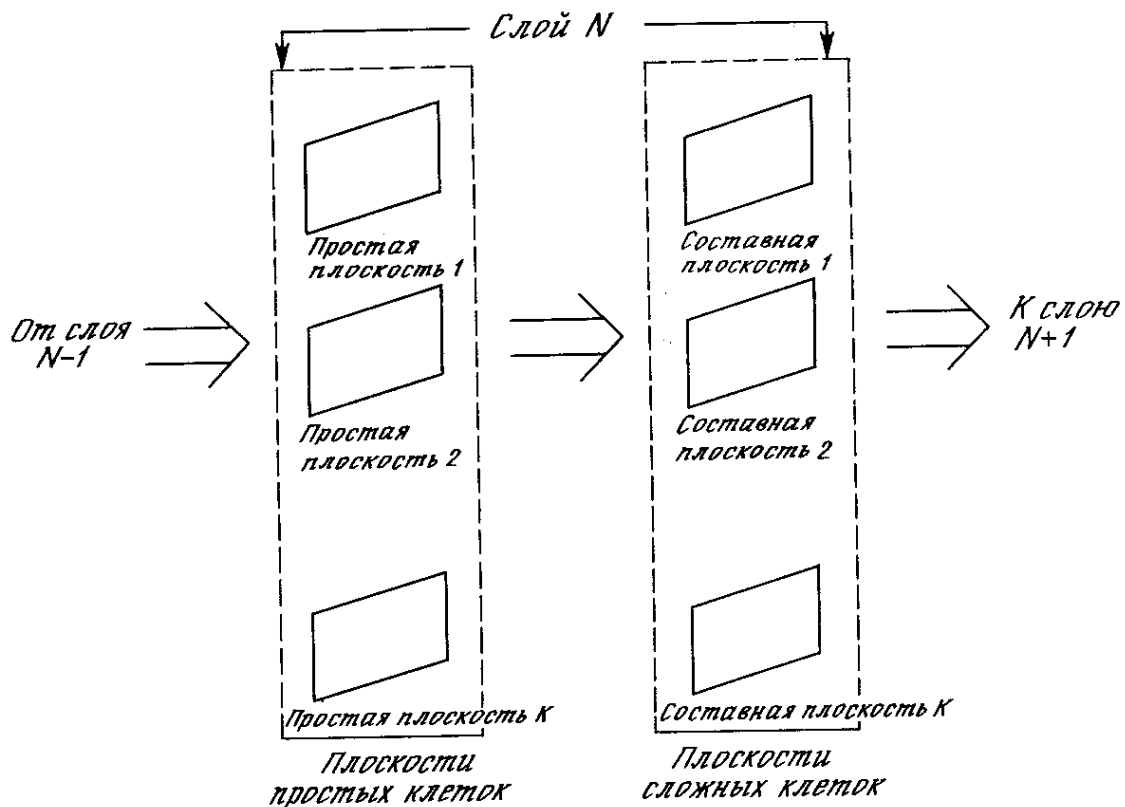
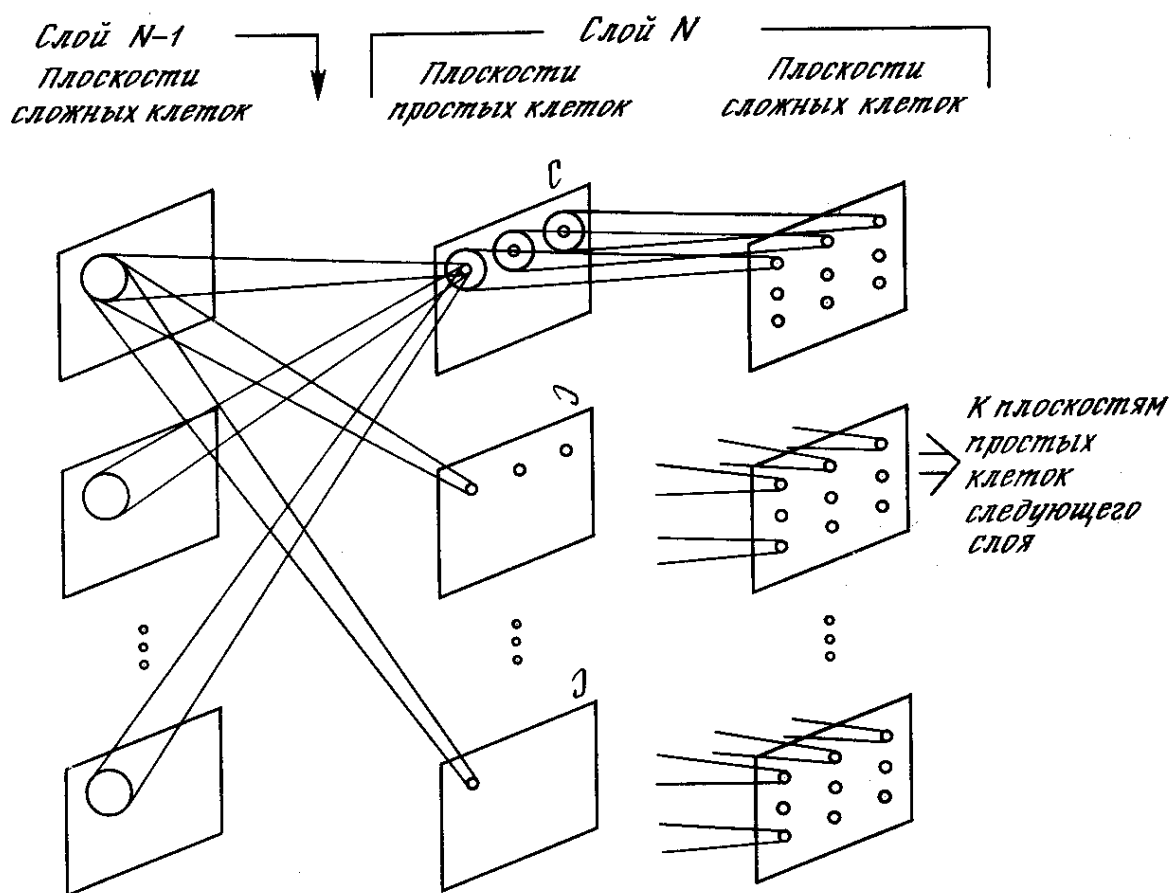


Рис. 10.9. Структура плоскостей неокогнитрона

**Простые узлы.** Все узлы в данной плоскости простых узлов реагируют на один и тот же образ. Как показано на рис. 10.10, плоскость простых узлов представляет массив узлов, каждый из которых «настраивается» на один специфический входной образ. Каждый простой узел чувствителен к ограниченной области входного образа, называемой его рецептивной областью. Например, все узлы в верхней плоскости простых узлов на рис. 10.10 реагируют на «С». Узел реагирует, если «С» встречается во входном образе и если «С» обнаружено в его рецептивной области.

На рис. 10.10 показано, что другие плоскости простых узлов в этом слое могут реагировать на поворот «С» на 90°, другие на поворот на 180° и т. д. Если должны быть выделены другие буквы (и их искаженные версии), дополнительные плоскости требуются для каждой из них.

Рецептивные области узлов в каждой плоскости простых узлов перекрываются с целью покрытия всего входного образа этого слоя. Каждый узел получает входы от соответствующих областей всех плоскостей комплексных узлов в предыдущем слое. Следовательно, простой узел реагирует на появление своего образа в любой сложной плоскости предыдущего слоя, если он окажется внутри его рецептивной области.



**Рис. 10.10. Система неокогнитрона**

**Комплексные узлы.** Задачей комплексных узлов является уменьшение зависимости реакции системы от позиции образов во входном поле. Для достижения этого каждый комплексный узел получает в качестве входного образа выходы набора простых узлов из соответствующей плоскости того же слоя. Эти простые узлы покрывают непрерывную область простой плоскости, называемую рецептивной областью комплексного узла. Возбуждение любого простого узла в этой области является достаточным для возбуждения данного комплексного узла. Таким образом, комплексный узел реагирует на тот же образ, что и простые узлы в соответствующей ему плоскости, но он менее чувствителен к позиции образа, чем любой из них.

Таким образом, каждый слой комплексных узлов реагирует на более широкую область входного образа, чем это делалось в предшествующих слоях. Эта прогрессия возрастает линейно от слоя к слою, приводя к требуемому уменьшению позиционной чувствительности системы в целом.

## Обобщение

Каждый нейрон в слое, близком к входному, реагирует на определенные образы в определенном месте, такие, как угол с определенной ориентацией в заданной позиции. Каждый слой в результате этого имеет более абстрактную, менее специфичную реакцию по сравнению с предшествующим; выходной слой реагирует на полные образы, показывая высокую степень независимости от их положения, размера и ориентации во входном поле. При использовании в качестве классификатора комплексный узел выходного слоя с наибольшей реакцией реализует выделение соответствующего образа во входном поле. В идеальном случае это выделение нечувствительно к позиции, ориентации, размерам или другим искажениям.

## Вычисления

Простые узлы в неокогнитроне имеют точно такие же характеристики, что и описанные для когнитрона, и используют те же формулы для определения их выхода. Здесь они не повторяются.

Тормозящий узел вырабатывает выход, пропорциональный квадратному корню из взвешенной суммы квадратов его входов. Заметим, что входы в тормозящий узел идентичны входам соответствующего простого узла и область включает область ответа во всех комплексных плоскостях. В символическом виде

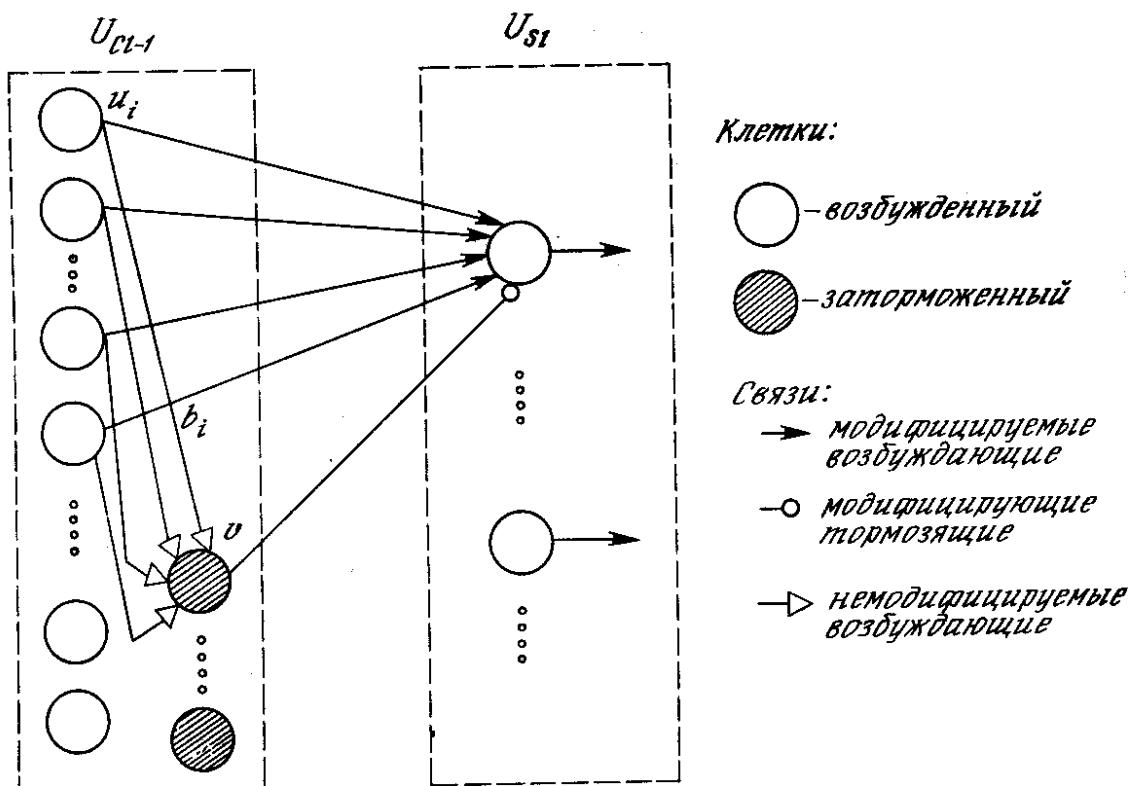
$$v = \sqrt{\sum_i (b_i u_i)^2},$$

где  $v$  – выход тормозящего узла;  $i$  – область над всеми комплексными узлами, с которыми связан тормозящий узел;  $b_i$  – вес  $i$ -й синаптической связи от комплексного узла к тормозящему узлу;  $u_i$  – выход  $i$ -го комплексного узла.

Веса  $b_i$  выбираются монотонно уменьшающимися с увеличением расстояния от центра области реакции, при этом сумма их значений должна быть равна единице.

## Обучение

Только простые узлы имеют настраиваемые веса. Это веса связей, соединяющих узел с комплексными узлами в предыдущем слое и имеющих изменяемую силу синапсов, настраиваемую таким образом, чтобы выработать максимальную реакцию на определенные стимулирующие свойства. Некоторые из этих синапсов являются возбуждающими и стремятся увеличить выход узлов, в то время как другие являются тормозящими и уменьшают выход узла.



### Рис. 10.11. Связи от сложных клеток одного уровня к простым клеткам следующего уровня

На рис. 10.11 показана полная структура синаптических связей между простым узлом и комплексными узлами в предшествующем слое. Каждый простой узел реагирует только на набор комплексных узлов внутри своей рецептивной области. Кроме того, существует тормозящий узел, реагирующий на те же самые комплексные узлы. Веса синапсов тормозящего узла не обучаются, – они выбираются таким образом, чтобы узел реагировал на среднюю величину выходов всех узлов, к которым он подключен. Единственный тормозящий синапс от тормозящего узла к простому узлу обучается, как и другие синапсы.

**Обучение без учителя.** Для обучения неокогнитрона на вход сети подается образ, который необходимо распознать, и веса синапсов настраиваются слой за слоем, начиная с набора простых узлов, ближайших ко входу. Величина синаптической связи от каждого комплексного узла к данному простому узлу увеличивается тогда и только тогда, когда удовлетворяются следующие два условия:

- 1) комплексный узел реагирует;
- 2) простой узел реагирует более сильно, чем любой из его соседних (внутри его области конкуренции).

Таким образом, простой узел обучается реагировать более сильно на образы, появляющиеся наиболее часто в его рецептивной области, что соответствует результатам исследований, полученных в экспериментах с котятами. Если распознаваемый образ отсутствует на входе, тормозящий узел предохраняет от случайного возбуждения.

Математическое описание процесса обучения и метод реализации латерального торможения аналогичны описанным для когнитрона, поэтому здесь они не повторяются. Необходимо отметить, что выходы простых и комплексных узлов являются аналоговыми, непрерывными и линейными и что алгоритм обучения предполагает их неотрицательность.

Когда выбирается простой узел, веса синапсов которого должны быть увеличены, он рассматривается как представитель всех узлов в плоскости, вызывая увеличение их синаптических связей на том же самом образе. Таким образом, все узлы в плоскости обучаются распознавать одни и те же свойства, и после обучения будут делать это независимо от позиции образа в поле комплексных узлов в предшествующем слое.

Эта система имеет ценную способность к самовосстановлению. Если данный узел выйдет из строя, будет найден другой узел, реагирующий более сильно, и этот узел будет обучен распознаванию входного образа, тем самым перекрывая действия своего отказавшего товарища.

**Обучение с учителем.** В работах [3] и [8] описано самоорганизующееся неуправляемое обучение. Наряду с этими впечатляющими результатами, были опубликованы отчеты о других экспериментах, использующих обучение с учителем [9]. Здесь требуемая реакция каждого слоя заранее определяется экспериментатором. Затем веса настраиваются с использованием обычных методов для выработки требуемой реакции. Например, входной слой настраивался для распознавания отрезков линий в различных ориентациях во многом аналогично первому слою обработки зрительной коры. Последующие слои обучались реагировать на более сложные и абстрактные свойства до тех пор, пока в выходном слое требуемый образ не будет выделен. При обработке сети, превосходно распознающей рукописные арабские цифры, экспериментаторы отказались от достижения биологического правдоподобия, обращая внимание только на достижение максимальной точности результатов системы.

**Реализация обучения.** В обычных конфигурациях рецептивное поле каждого нейрона возрастает при переходе к следующему слою. Однако количество нейронов в слое будет уменьшаться при переходе от входных к выходным слоям. Наконец, выходной слой имеет только один нейрон в плоскости сложных узлов. Каждый такой нейрон представляет определенный входной образ, которому сеть была обучена. В процессе классификации входной образ подается на вход неокогнитрона и вычисляются выходы слоев за слоем, начиная с входного слоя. Так как только небольшая часть входного образа подается на вход каждого простого узла входного слоя, некоторые простые узлы регистрируют наличие характеристик, которым они обучены, и возбуждаются. В следующем слое выделяются более сложные характеристики как определенные комбинации выходов комплексных узлов. Слои за слоем свойства комбинируются во все возрастающем диапазоне; выделяются более общие характеристики и уменьшается позиционная чувствительность.

В идеальном случае только один нейрон выходного слоя должен возбудиться. В действительности обычно будет возбуждаться несколько нейронов с различной силой, и входной образ должен быть определен с учетом соотношения их выходов. Если используется сила латерального торможения, возбуждаться будет только нейрон с максимальным выходом. Однако это часто является не лучшим вариантом. На практике простая функция от небольшой группы наиболее сильно возбужденных нейронов будет часто улучшать точность классификации.

## ЗАКЛЮЧЕНИЕ

Как когнитрон, так и неокогнитрон производят большое впечатление с точки зрения точности, с которой они моделируют биологическую нервную систему. Тот факт, что эти системы показывают результаты, имитирующие некоторые аспекты способностей человека к обучению и познанию, наводит на мысль, что наше понимание функций мозга приближается к уровню, способному принести практическую пользу.



Неокогнитрон является сложной системой и требует существенных вычислительных ресурсов. По этим причинам кажется маловероятным, что такие системы реализуют оптимальное инженерное решение сегодняшних проблем распознавания образов. Однако с 1960 г. стоимость вычислений уменьшалась в два раза каждые два-три года, тенденция, которая, по всей вероятности, сохранится в течение как минимум ближайших десяти лет. Несмотря на то, что многие подходы, казавшиеся нереализуемыми несколько лет назад, являются общепринятыми сегодня и могут оказаться тривиальными через несколько лет, реализация моделей неокогнитрона на универсальных компьютерах является бесперспективной. Необходимо достигнуть тысячекратных улучшений стоимости и производительности компьютеров за счет специализации архитектуры и внедрения технологии СБИС, чтобы сделать неокогнитрон практической системой для решения сложных проблем распознавания образов, однако ни эта, ни какая-либо другая модель искусственных нейронных сетей не должны отвергаться только на основании их высоких вычислительных требований.

#### Литература

1. Blakemore C., Cooper G. F. 1970. Development of the brain depends on the visual environment. *Nature* 228(5270):477–78.
2. Fukushima K. 1975. Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics* 20:121–36.
3. Fukushima K. 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics* 36(4):193–202.
4. Fukushima K. 1981. Cognitron: A self-organizing multilayer neural network model. *NHK Technical Monograph No. 30*, pp. 1–25. Available from Nippon Hoso Kyokai (Japanese Broadcasting Corp.), Technical Research Labs, Tokyo, Japan.
5. Fukushima K. 1984. A hierarchical neural network model for associative memory. *Biological Cybernetics* 50:105–113.
6. Fukushima K. 1986. A neural network model for selective attention in visual pattern recognition. *Biological Cybernetics* 55(1):5–15.
7. Fukushima K. 1987. A neural network for selective attention. In *Proceedings of the IEEE First International Conference on Neural Networks*, eds. M. Caudill and C. Butler, vol. 2, pp. 11–18. San Diego, CA: SOS Printing.
8. Fukushima K., Miyake S. 1982. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern recognition* 15(6): 455–69.
9. Fukushima K., Miyake S., Takayuki I. 1983. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Transaction on Systems, Man and Cybernetics* SMC-13(5):826–34.
10. Hubel D. H., Wiesel T. N. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology* 160:106–54.
11. Hubel D. H., Wiesel T. N. 1965. Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cats. *Journal of Neurophysiology* 28:229–89.
12. Hubel D. H., Wiesel T. N. 1977. Functional architecture of macaque monkey visual cortex. *Proceedings of the Royal Society, London. Ser. B* 198, pp. 1–59.

## Приложение А.

### Биологические нейронные сети

#### ЧЕЛОВЕЧЕСКИЙ МОЗГ: БИОЛОГИЧЕСКАЯ МОДЕЛЬ ДЛЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Структура искусственных нейронных сетей была смоделирована как результат изучения человеческого мозга. Как мы отмечали выше, сходство между ними в действительности очень незначительно, однако даже эта скромная эмуляция мозга приносит ощутимые результаты. Например, искусственные нейронные сети имеют такие аналогичные мозгу свойства, как способность обучаться на опыте, основанном на знаниях, делать абстрактные умозаключения и совершать ошибки, что является более характерным для человеческой мысли, чем для созданных человеком компьютеров.

Учитывая успехи, достигнутые при использовании грубой модели мозга, кажется естественным ожидать дальнейшего продвижения вперед при использовании более точной модели. Разработка такой модели требует детального понимания структуры и функций мозга. Это в свою очередь требует определения точных характеристик нейронов, включая их вычислительные элементы и элементы связи. К сожалению, информация не является полной;

большая часть мозга остается тайной для понимания. Основные исследования проведены в области идентификации функций мозга, однако и здесь отсутствуют подходы, отличающиеся от чисто «схематических». Биохимия нейронов, фундаментальных строительных блоков мозга, очень неохотно раскрывает свои секреты. Каждый год приносит новую информацию относительно электрохимического поведения нейронов, причем всегда в направлении раскрытия новых уровней сложности. Ясно одно: нейрон является намного более сложным, чем представлялось несколько лет назад, и нет полного понимания процесса его функционирования.

Однако, несмотря на наши ограниченные познания, мозг может быть использован в качестве ценной модели в вопросах развития искусственных нейронных сетей. Используя метод проб и ошибок, эволюция, вероятно, привела к структурам, оптимальным образом пригодным для решения проблем, более характерных для человека. Кажется маловероятным, что мы получим более хорошее решение. Тщательно моделируя мозг, мы продвигаемся в исследовании природы и в будущем будем, вероятно, воспроизводить больше возможностей мозга.

Данное приложение содержит штриховые наброски современных знаний относительно структуры и функций мозга. Хотя изложение этих сведений очень краткое, мы пытались сохранить точность. Следующие разделы иллюстрируют текст данной работы и, возможно, будут стимулировать интерес к биологическим системам, что приведет к развитию искусственных нейронных сетей.

#### ОРГАНИЗАЦИЯ ЧЕЛОВЕЧЕСКОГО МОЗГА

Человеческий мозг содержит свыше тысячи миллиардов вычислительных элементов, называемых *нейронами*. Превышая по количеству число звезд в Млечном Пути галактики, эти нейроны связаны сотнями триллионов нервных нитей, называемых *синапсами*. Эта сеть нейронов отвечает за все явления, которые мы называем мыслями, эмоциями, познанием, а также и за совершение мириадов сенсомоторных и автономных функций. Пока мало понятно, каким образом все это происходит, но уже исследовано много вопросов физиологической структуры и определенные функциональные области постепенно изучаются исследователями.

Мозг также содержит густую сеть кровеносных сосудов, которые обеспечивают кислородом и питательными веществами нейроны и другие ткани. Эта система кровоснабжения связана с главной системой кровообращения посредством высокоэффективной фильтрующей системы, называемой *гематоэнцефалическим барьером*, этот барьер является механизмом защиты, который предохраняет мозг от возможных токсичных веществ, находящихся в крови. Защита обеспечивается низкой проницаемостью кровеносных сосудов мозга, а также плотным перекрытием глиальных клеток, окружающих нейроны. Кроме этого, глиальные клетки обеспечивают структурную основу мозга. Фактически весь объем мозга, не занятый нейронами и кровеносными сосудами, заполнен глиальными клетками.

Гематоэнцефалический барьер является основой для обеспечения сохранности мозга, но он значительно осложняет лечение терапевтическими лекарствами. Он также мешает исследованиям, изучающим влияние различных химических веществ на функции мозга. Лишь небольшая часть лекарств, созданных с целью влияния на мозг, может преодолевать этот барьер. Лекарства состоят из небольших молекул, способных проникать через крошечные поры в кровеносных сосудах. Чтобы воздействовать на функции мозга, они должны затем пройти через глиальные клетки или раствориться в их мембране. Лишь некоторые молекулы интересующих нас лекарств удовлетворяют этим требованиям; молекулы многих терапевтических лекарств задерживаются этим барьером.

Мозг является основным потребителем энергии тела. Включая в себя лишь 2% массы тела, в состоянии покоя он использует приблизительно 20% кислорода тела. Даже когда мы спим, расходование энергии продолжается. В действительности существуют доказательства возможности увеличения расходования энергии во время фазы сна, сопровождаемой движением глаз. Потребляя только 20 Вт, мозг с энергетической точки зрения невероятно эффективен. Компьютеры с одной крошечной долей вычислительных возможностей мозга потребляют много тысяч ватт и требуют сложных средств для охлаждения, предохраняющего их от температурного саморазрушения.

### Нейрон

Нейрон является основным строительным блоком нервной системы. Он является клеткой, подобной всем другим клеткам тела; однако определенные существенные отличия позволяют ему выполнять все вычислительные функции и функции связи внутри мозга.

Как показано на рис. А.1, нейрон состоит из трех частей: тела клетки, дендритов и аксона, каждая часть со своими, но взаимосвязанными функциями.

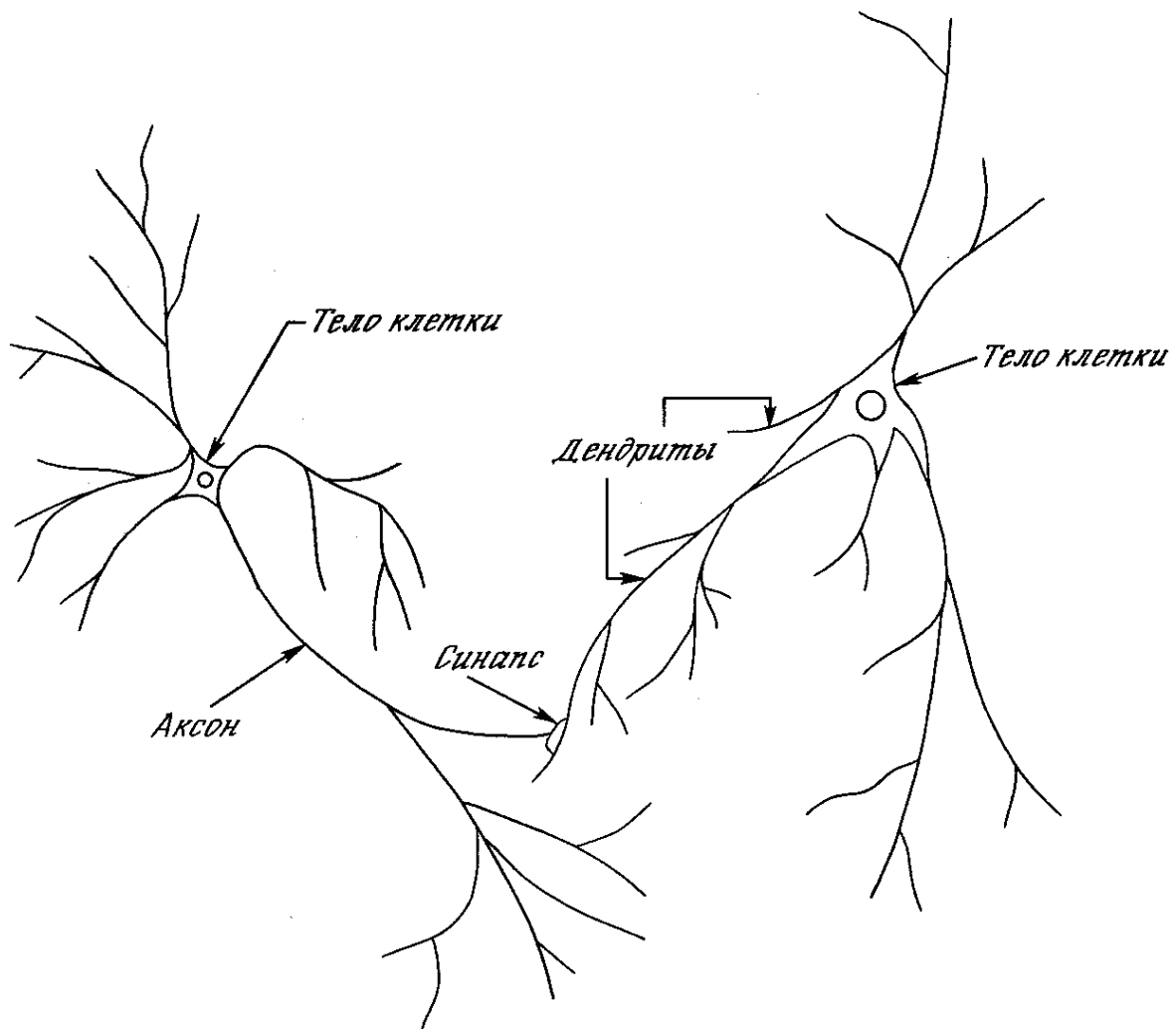


Рис. А.1. Компоненты нейрона

Функционально дендриты получают сигналы от других клеток через контакты, называемые синапсами. Отсюда сигналы проходят в тело клетки, где они суммируются с другими такими же сигналами. Если суммарный сигнал в течение короткого промежутка времени является достаточно большим, клетка возбуждается, вырабатывая в аксоне импульс, который передается на следующие клетки. Несмотря на очевидное упрощение, эта схема функционирования объясняет большинство известных процессов мозга.

**Тело ячейки.** Нейроны в мозгу взрослого человека не восстанавливаются; они отмирают. Это означает, что все компоненты должны непрерывно заменяться, а материалы обновляться по мере необходимости. Большинство этих процессов происходит в теле клетки, где изменение химических факторов приводит к большим изменениям сложных молекул. Кроме этого, тело клетки управляет расходом энергии нейрона и регулирует множество других процессов в клетке. Внешняя мембрана тела клетки нейрона имеет уникальную способность генерировать нервные импульсы (потенциалы действия), являющиеся жизненными функциями нервной системы и центром ее вычислительных способностей.

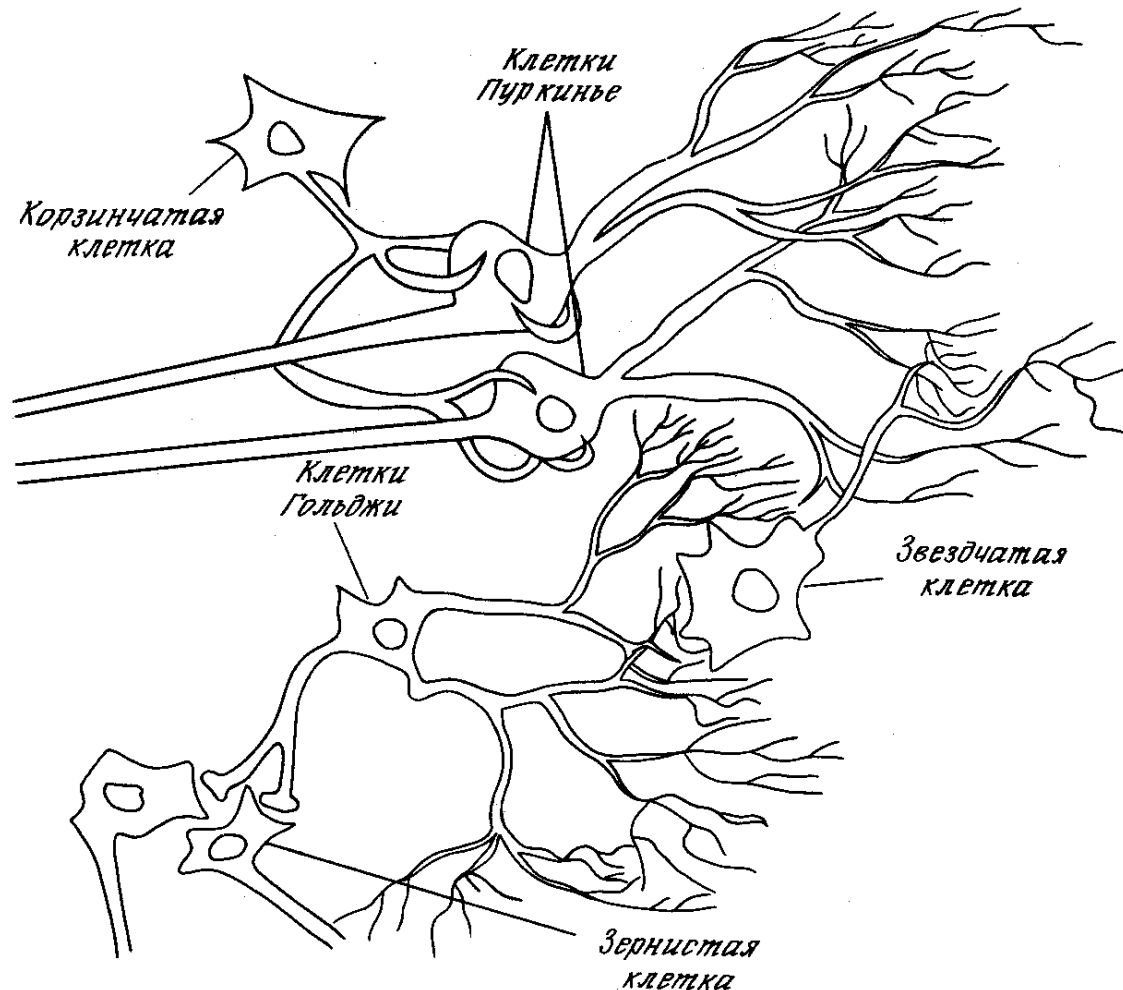


Рис. А.2. Типы нейронов

Были идентифицированы сотни типов нейронов, каждый со своей характерной формой тела клетки (рис. А.2), имеющей обычно от 5 до 100 мкм в диаметре. В настоящее время этот факт рассматривается как проявление случайности, однако могут быть найдены различные морфологические конфигурации, отражающие важную функциональную специализацию. Определение функций различных типов клеток является в настоящее время предметом интенсивных исследований и основой понимания обрабатывающих механизмов мозга.

**Дендриты.** Большинство входных сигналов от других нейронов попадают в клетку через дендриты, представляющие собой густо ветвящуюся структуру, исходящую от тела клетки. На дендритах располагаются синаптические соединения, которые получают сигналы от других аксонов. Кроме этого, существует огромное количество синаптических связей от аксона к аксону, от аксона к телу клетки и от дендрита к дендриту; их функции не очень ясны, но они слишком широко распространены, чтобы не считаться с ними.

В отличие от электрических цепей, синаптические контакты обычно не являются физическими или электрическими соединениями. Вместо этого имеется узкое пространство, называемое *синаптической щелью*, отделяющее дендрит от передающего аксона. Специальные химические вещества, выбрасываемые аксоном в синаптическую щель, диффундируют к дендриту. Эти химические вещества, называемые *нейротрансмиттерами*, улавливаются специальными рецепторами на дендрите и внедряются в тело клетки.

Определено более 30 видов нейротрансмиттеров. Некоторые из них являются возбуждающими и стремятся вызвать возбуждение клетки и выработать выходной импульс. Другие являются тормозящими и стремятся подавить такой импульс. Тело клетки суммирует сигналы, полученные от дендритов, и если их результирующий сигнал выше порогового значения, вырабатывается импульс, проходящий по аксону к другим нейронам.

**Аксон.** Аксон может быть как коротким (0,1 мм), так и превышать длину 1 м, распространяясь в другую часть тела человека. На конце аксон имеет множество ветвей, каждая из которых завершается синапсом, откуда сигнал передается в другие нейроны через дендриты, а в некоторых случаях прямо в тело клетки. Таким образом, всего один нейрон может генерировать импульс, который возбуждает или затормаживает сотни или тысячи других нейронов, каждый из которых, в свою очередь, через свои дендриты может воздействовать на сотни или тысячи других нейронов. Таким образом, эта высокая степень связанности, а не функциональная сложность самого нейрона, обеспечивает нейрону его вычислительную мощност.

Синаптическая связь, завершающая ветвь аксона, представляет собой маленькие утолщения, содержащие сферические структуры, называемые *синаптическими пузырьками*, каждый из которых содержит большое число нейротрансмиттерных молекул. Когда нервный импульс приходит в аксон, некоторые из этих пузырьков высвобождают свое содержимое в синаптическую щель, тем самым инициализируя процесс взаимодействия нейронов (рис. А.3).

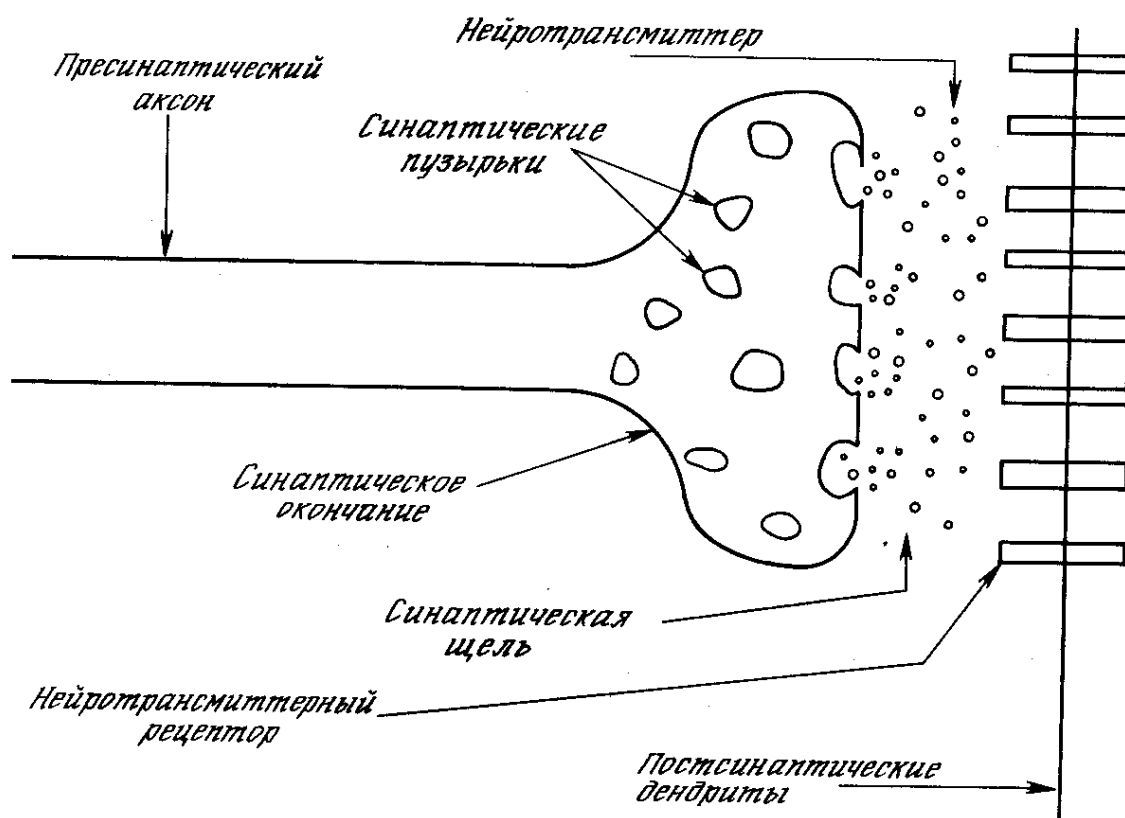


Рис. А.3. Синапс

Кроме распространения такого бинарного сигнала, обеспечиваемого возбуждением первого импульса, в нейронах при слабой стимуляции могут также распространяться электрохимические сигналы с последовательной реакцией. Локальные по своей природе, эти сигналы быстро затухают с удалением от места возбуждения, если не будут усилены. Природа использует это свойство первых клеток путем создания вокруг аксонов изолирующей оболочки из шванговских клеток. Эта оболочка, называемая миелиновой, прерывается приблизительно через каждый миллиметр вдоль аксона узкими разрывами, называемыми *узлами*, или *перехватами Ранвье*. Нервные импульсы, проходящие в аксон, передаются скачкообразно от узла к узлу. Таким образом, аксону нет нужды расходовать энергию для поддержания своего химического градиента по всей своей длине. Только оставшиеся неизолрованными перехваты Ранвье являются объектом генерации первого импульса; для передачи сигнала от узла к узлу более эффективными являются градуальные реакции. Кроме этого свойства оболочки, обеспечивающего сохранение энергии, известны ее другие свойства. Например, миелинизированные нервные окончания передают сигналы значительно быстрее немиелинизированных. Обнаружено, что некоторые болезни приводят к ухудшению этой изоляции, что, по-видимому, является причиной других болезней.

## Мембрана клетки

В мозгу существует 2 типа связей: передача химических сигналов через синапсы и передача электрических сигналов внутри нейрона. Великолепное сложное действие мембраны создает способность клетки вырабатывать и передавать оба типа этих сигналов.

Мембрана клетки имеет около 5 нм толщины и состоит из двух слоев липидных молекул. Встроенные в мембрану различные специальные протеины можно разделить на пять классов: насосы, каналы, рецепторы, энзимы и структурные протеины.

Насосы активно перемещают ионы через мембрану клетки для поддержания градиентов концентрации. Каналы пропускают ионы выборочно и управляют их прохождением через мембрану. Некоторые каналы открываются или закрываются распространяющимся через мембрану электрическим потенциалом, тем самым обеспечивая быстрое и чувствительное средство изменения ионных градиентов. Другие типы каналов управляются химически, изменяя свою проницаемость при получении химических носителей.

Рецепторами являются протеины, которые распознают и присоединяют многие типы молекул из окружения клетки с большой точностью. Энзимы оболочки ускоряют разнообразные химические реакции внутри или около клеточной мембраны. Структурные протеины соединяют клетки и помогают поддерживать структуру самой клетки.

Внутренняя концентрация натрия в клетке в 10 раз ниже, чем в ее окружении, а концентрация калия в 10 раз выше. Эти концентрации стремятся к выравниванию с помощью утечки через поры в мембране клетки. Чтобы сохранить необходимую концентрацию, протеиновые молекулы мембраны, называемые натриевыми насосами, постоянно отсасывают натрий из клетки и подкачивают калий в клетку. Каждый насос перемещает приблизительно две сотни ионов натрия и около ста тридцати ионов калия в секунду. Нейрон может иметь миллионы таких насосов, перемещающих сотни миллионов ионов калия и натрия через мембрану клетки в каждую секунду. На концентрацию калия внутри ячейки влияет также наличие большого числа постоянно открытых калиевых каналов, т. е. протеиновых молекул, которые хорошо пропускают ионы калия в клетку, но препятствуют прохождению натрия. Комбинация этих двух механизмов отвечает за создание и поддержание динамического равновесия, соответствующего состоянию нейрона в покое.

Градиент ионной концентрации в мембране клетки вырабатывает внутри клетки электрический потенциал  $-70$  мВ относительно ее окружения. Чтобы возбудить клетку (стимулировать возникновение потенциала действия) синаптические входы должны уменьшить этот уровень до приблизительно  $-50$  мВ. При этом потоки натрия и калия сразу направляются в обратную сторону; в течение миллисекунд внутренний потенциал клетки становится  $+50$  мВ относительно внешнего окружения. Это изменение полярности быстро распространится через клетку, заставляя нервный импульс распространиться по всему аксону до его пресинаптических окончаний. Когда импульс достигнет окончания аксона, открываются управляемые напряжением кальциевые каналы. Это вызывает освобождение нейротрансмиттерных молекул в синаптическую щель и процесс распространяется на другие нейроны. После генерации потенциала действия клетка войдет в рефракторный период на несколько миллисекунд, в течении которого она восстановит свой первоначальный потенциал для подготовки к генерации следующего импульса.

Рассмотрим этот процесс более детально. Первоначальное получение нейротрансмиттерных молекул снижает внутренний потенциал клетки с  $-70$  до  $-50$  мВ. При этом зависимые от потенциала натриевые каналы открываются, позволяя натрию проникнуть в клетку. Это еще более уменьшает потенциал, увеличивая приток натрия в клетку, и создает самоусиливающийся процесс, который быстро распространяется в соседние области, изменяя локальный потенциал клетки с отрицательного до положительного.

Через некоторое время после открытия натриевые каналы закрываются, а калиевые каналы открываются. Это создает усиленный поток ионов калия из клетки, что восстанавливает внутренний потенциал  $-70$  мВ. Это быстрое изменение напряжения образует потенциал действия, который быстро распространяется по всей длине аксона подобно лавине.

Натриевые и калиевые каналы реагируют на потенциал клетки и, следовательно, можно сказать, что они управляют напряжением. Другой тип каналов является химически управляемым. Эти каналы открываются только тогда, когда специальная нейротрансмиттерная молекула попадает на рецептор, и они совсем не чувствительны к напряжению. Такие каналы обнаруживаются в постсинаптических мембранах на дендритах и ответственны за реакцию нейронов на воздействие различных нейротрансмиттерных молекул. Чувствительный к ацетилхолину белок (ацетилхолиновый рецептор) является одним из таких химических каналов. Когда молекулы ацетилхолина выделяются в синаптическую щель, они диффундируют к ацетилхолиновым рецепторам, входящим в постсинаптическую мембрану. Эти рецепторы (которые также являются каналами) затем открываются, обеспечивая свободный проход как калия, так и натрия через мембрану. Это приводит к кратковременному локальному уменьшению отрицательного внутреннего потенциала клетки (формируя положительный импульс). Так как импульсы являются короткими и слабыми, то чтобы заставить клетку выработать необходимый электрический потенциал, требуется открытие многих таких каналов.

Ацетилхолиновые рецепторы-каналы пропускают и натрий, и калий, вырабатывая тем самым положительные импульсы. Такие импульсы являются возбуждающими, поскольку они способствуют появлению необходимого потенциала. Другие химически

управляемые каналы пропускают только калиевые ионы из клетки, производя отрицательный импульс; эти импульсы являются тормозящими, поскольку они препятствуют возбуждению клетки.

Гамма-аминоасляная кислота (ГАМК) является одним из более общих тормозных нейротрансмиттеров. Обнаруженная почти исключительно в головном и спинном мозге, она попадает на рецептор канала, который выборочно пропускает ионы хлора. После входа эти ионы увеличивают отрицательный потенциал клетки и тем самым препятствуют ее возбуждению. Дефицит ГАМК связан с хореей Хантингтона, имеющей неврологический синдром, вызывающий неконтрольное движение мускулатуры. К несчастью, гематоэнцефалический барьер препятствует увеличению снабжения ГАМК, и как выйти из этого положения, пока неизвестно. Вероятно, что и другие неврологические и умственные расстройства будут наблюдаться при подобных нарушениях в нейротрансмиттерах или других химических носителях. Уровень возбуждения нейрона определяется кумулятивным эффектом большого числа возбуждающих и тормозящих входов, суммируемых телом клетки в течение короткого временного интервала. Получение возбуждающей нейротрансмиттерной молекулы будет увеличивать уровень возбуждения нейрона; их меньшее количество или смесь тормозящих молекул уменьшает уровень возбуждения. Таким образом, нейронный сигнал является импульсным или частотно-модулируемым (ЧМ). Этот метод модуляции, широко используемый в технике (например, ЧМ-радио), имеет значительные преимущества при наличии помех по сравнению с другими способами. Исследования показали изумляющую сложность биохимических процессов в мозге. Например, предполагается наличие свыше 30 веществ, являющихся нейротрансмиттерами, и большое количество рецепторов с различными ответными реакциями. Более того, действие определенных нейротрансмиттерных молекул зависит от типа рецептора в постсинаптической мембране, некоторые нейротрансмиттеры могут быть возбуждающими для одного синапса и тормозящими для другого. Кроме того, внутри клетки существует система «вторичного переносчика», которая включается при получении нейротрансмиттера, что приводит к выработке большого количества молекул циклического аденозинтрифосфата, тем самым производя значительное усиление физиологических реакций.

Исследователи всегда надеются найти простые образы для унификации сложных и многообразных наблюдений. Для нейробиологических исследований такие простые образы до сих пор не найдены. Большинство результатов исследований подвергаются большому сомнению прежде, чем ими воспользуются. Одним из таких результатов в изучении мозга явилось открытие множества видов электрохимической деятельности, обнаруженных в работе мозга; задачей является их объединение в связанную функциональную модель.

## **КОМПЬЮТЕРЫ И ЧЕЛОВЕЧЕСКИЙ МОЗГ**

Существует подобие между мозгом и цифровым компьютером: оба оперируют электронными сигналами, оба состоят из большого количества простых элементов, оба выполняют функции, являющиеся, грубо говоря, вычислительными. Тем не менее существуют и фундаментальные отличия. По сравнению с микросекундными и даже наносекундными интервалами вычислений современных компьютеров нервные импульсы являются слишком медленными. Хотя каждый нейрон требует наличия миллисекундного интервала между передаваемыми сигналами, высокая скорость вычислений мозга обеспечивается огромным числом параллельных вычислительных блоков, причем количество их намного превышает доступное современным ЭВМ. Диапазон ошибок представляет другое фундаментальное отличие: ЭВМ присуща свобода от ошибок, если входные сигналы безупречно точны и ее аппаратное и программное обеспечение не повреждены. Мозг же часто производит лучшее угадывание и приближение при частично незавершенных и неточных входных сигналах. Часто он ошибается, но величина ошибки должна гарантировать наше выживание в течение миллионов лет.

Первые цифровые вычислители часто рассматривались как «электронный мозг». С точки зрения наших текущих знаний о сложности мозга, такое заявление оптимистично, да и просто не соответствует истине. Эти две системы явно различаются в каждой своей части. Они оптимизированы для решения различных типов проблем, имеют существенные различия в структуре и их работа оценивается различными критериями.

Некоторые говорят, что искусственные нейронные сети когда-нибудь будут дублировать функции человеческого мозга. Прежде чем добиться этого, необходимо понять организацию и функции мозга. Эта задача, вероятно, не будет решена в ближайшем будущем. Надо отметить то, что современные нейросети базируются на очень упрощенной модели, игнорирующей большинство тех знаний, которые мы имеем о детальном функционировании мозга. Поэтому необходимо разработать более точную модель, которая могла бы качественнее моделировать работу мозга.

Прорыв в области искусственных нейронных сетей будет требовать развития их теоретического фундамента. Теоретические выкладки, в свою очередь, должны предваряться улучшением математических методов, поскольку исследования серьезно тормозятся нашей неспособностью иметь дело с такими системами. Успокаивает тот факт, что современный уровень математического обеспечения был достигнут под влиянием нескольких превосходных исследователей. В действительности аналитические проблемы являются сверхтрудными, так как рассматриваемые системы являются очень сложными нелинейными динамическими системами. Возможно, для описания систем, имеющих сложность головного мозга, необходимы совершенно новые математические методы. Может быть и так, что разработать полностью удовлетворяющий всем требованиям аппарат невозможно.

Несмотря на существующие проблемы, желание смоделировать человеческий мозг не угасает, а получение зачаровывающих результатов вдохновляет на дальнейшие усилия. Успешные модели, основанные на предположениях о структуре мозга, разрабатываются нейроанатомами и нейрофизиологами с целью их изучения для согласования структуры и функций этих моделей. С другой стороны, успехи в биологической науке ведут к модификации и тщательной разработке искусственных моделей. Аналогично инженеры применяют искусственные модели для реализации мировых проблем и получают положительные результаты, несмотря на отсутствие полного взаимопонимания.

Объединение научных дисциплин для изучения проблем искусственных нейросетей принесет эффективные результаты, которые могут стать беспримерными в истории науки. Биологи, анатомы, физиологи, инженеры, математики и даже философы активно включились в процесс исследований. Проблемы являются сложными, но цель высока: познается сама человеческая мысль.



## Приложение Б.

### Алгоритмы обучения

Искусственные нейронные сети обучаются самыми разнообразными методами. К счастью, большинство методов обучения исходят из общих предпосылок и имеет много идентичных характеристик. Целью данного приложения является обзор некоторых фундаментальных алгоритмов, как с точки зрения их текущей применимости, так и с точки зрения их исторической важности. После ознакомления с этими фундаментальными алгоритмами другие, основанные на них, алгоритмы будут достаточно легки для понимания и новые разработки также могут быть лучше поняты и развиты.

#### ОБУЧЕНИЕ С УЧИТЕЛЕМ И БЕЗ УЧИТЕЛЯ

Обучающие алгоритмы могут быть классифицированы как алгоритмы обучения с учителем и без учителя. В первом случае существует учитель, который предъявляет входные образы сети, сравнивает результирующие выходы с требуемыми, а затем настраивает веса сети таким образом, чтобы уменьшить различия. Трудно представить такой обучающий механизм в биологических системах; следовательно, хотя данный подход привел к большим успехам при решении прикладных задач, он отвергается исследователями, полагающими, что искусственные нейронные сети обязательно должны использовать те же механизмы, что и человеческий мозг.

Во втором случае обучение проводится без учителя, при предъявлении входных образов сеть самоорганизуется посредством настройки своих весов согласно определенному алгоритму. Вследствие отсутствия указания требуемого выхода в процессе обучения результаты непредсказуемы с точки зрения определения возбуждающих образов для конкретных нейронов. При этом, однако, сеть организуется в форме, отражающей существенные характеристики обучающего набора. Например, входные образы могут быть классифицированы согласно степени их сходства так, что образы одного класса активизируют один и тот же выходной нейрон.

#### МЕТОД ОБУЧЕНИЯ ХЭББА

Работа [2] обеспечила основу для большинства алгоритмов обучения, которые были разработаны после ее выхода. В предшествующих этой работе трудах в общем виде определялось, что обучение в биологических системах происходит посредством некоторых физических изменений в нейронах, однако отсутствовали идеи о том, каким образом это в действительности может иметь место. Основываясь на физиологических и психологических исследованиях, Хэбб в [2] интуитивно выдвинул гипотезу о том, каким образом может обучаться набор биологических нейронов. Его теория предполагает только локальное взаимодействие между нейронами при отсутствии глобального учителя; следовательно, обучение является неуправляемым. Несмотря на то что его работа не включает математического анализа, идеи, изложенные в ней, настолько ясны и непринужденны, что получили статус универсальных допущений. Его книга стала классической и широко изучается специалистами, имеющими серьезный интерес в этой области.

#### Алгоритм обучения Хэбба

По существу Хэбб предположил, что синаптическое соединение двух нейронов усиливается, если оба эти нейрона возбуждены. Это можно представить как усиление синапса в соответствии с корреляцией уровней возбужденных нейронов, соединяемых данным синапсом. По этой причине алгоритм обучения Хэбба иногда называется корреляционным алгоритмом.

Идея алгоритма выражается следующим равенством:

$$w_{ij}(t+1) = w_{ij}(t) + \text{NET}_i \text{NET}_j,$$

где  $w_{ij}(t)$  – сила синапса от нейрона  $i$  к нейрону  $j$  в момент времени  $t$ ;  $\text{NET}_i$  – уровень возбуждения пресинаптического нейрона;  $\text{NET}_j$  – уровень возбуждения постсинаптического нейрона.

Концепция Хэбба отвечает на сложный вопрос, каким образом обучение может проводиться без учителя. В методе Хэбба обучение является исключительно локальным явлением, охватывающим только два нейрона и соединяющий их синапс; не требуется глобальной системы обратной связи для развития нейронных образований.

Последующее использование метода Хэбба для обучения нейронных сетей привело к большим успехам, но наряду с этим показало ограниченность метода; некоторые образы просто не могут использоваться для обучения этим методом. В результате появилось большое количество расширений и нововведений, большинство из которых в значительной степени основано на работе Хэбба.

### Метод сигнального обучения Хэбба

Как мы видели, выход NET простого искусственного нейрона является взвешенной суммой его входов. Это может быть выражено следующим образом:

$$\text{NET}_j = \sum_i \text{OUT}_i w_{ij}$$

где  $\text{NET}_j$  – выход NET нейрона  $j$ ;  $\text{OUT}_i$  – выход нейрона  $i$ ;  $w_{ij}$  – вес связи нейрона  $i$  с нейроном  $j$ .

Можно показать, что в этом случае линейная многослойная сеть не является более мощной, чем однослойная сеть; рассматриваемые возможности сети могут быть улучшены только введением нелинейности в передаточную функцию нейрона. Говорят, что сеть, использующая сигмоидальную функцию активации и метод обучения Хэбба, обучается по сигнальному методу Хэбба. В этом случае уравнение Хэбба модифицируется следующим образом:

$$\text{OUT}_i = \frac{1}{1 + \exp(-\text{NET}_i)} = F(\text{NET}_i)$$

$$w_{ij}(t+1) = w_{ij}(t) + \text{OUT}_i \text{OUT}_j$$

где  $w_{ij}(t)$  – сила синапса от нейрона  $i$  к нейрону  $j$  в момент времени  $t$ ;  $\text{OUT}_i$  – выходной уровень пресинаптического нейрона равный  $F(\text{NET}_i)$ ;  $\text{OUT}_j$  – выходной уровень постсинаптического нейрона равный  $F(\text{NET}_j)$ .

### Метод дифференциального обучения Хэбба

Метод сигнального обучения Хэбба предполагает вычисление свертки предыдущих изменений выходов для определения изменения весов. Настоящий метод, называемый методом дифференциального обучения Хэбба, использует следующее равенство:

$$w_{ij}(t+1) = w_{ij}(t) + [\text{OUT}_i(t) - \text{OUT}_i(t-1)][\text{OUT}_j(t) - \text{OUT}_j(t-1)],$$

где  $w_{ij}(t)$  – сила синапса от нейрона  $i$  к нейрону  $j$  в момент времени  $t$ ;  $\text{OUT}_i(t)$  – выходной уровень пресинаптического нейрона в момент времени  $t$ ;  $\text{OUT}_j(t)$  – выходной уровень постсинаптического нейрона в момент времени  $t$ .

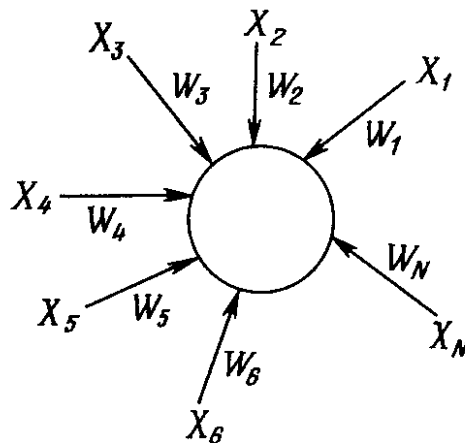


Рис. Б.1. Сеть «Инстар» Гроссберга

### ВХОДНЫЕ И ВЫХОДНЫЕ ЗВЕЗДЫ

Много общих идей, используемых в искусственных нейронных сетях, прослеживаются в работах Гроссберга; в качестве примера можно указать конфигурации входных и выходных звезд [1], используемые во многих сетевых парадигмах. Входная звезда, как показано на рис. Б.1, состоит из нейрона, на который подается группа входов через синаптические веса. Выходная звезда, показанная на рис. Б.2, является нейроном, управляющим группой весов. Входные и выходные звезды могут быть взаимно соединены в сети любой сложности; Гроссберг рассматривает их как модель определенных биологических функций. Вид звезды определяет ее название, однако звезды обычно изображаются в сети иначе.

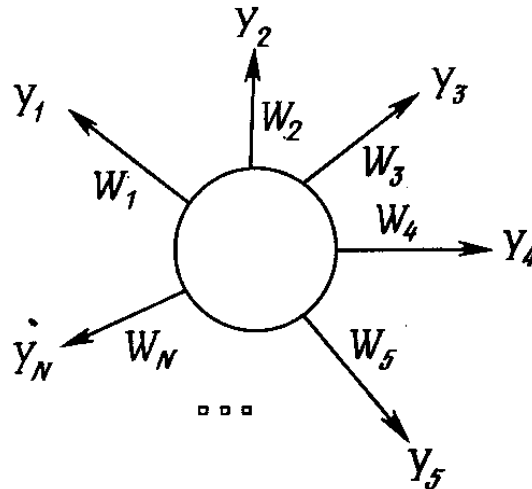


Рис. Б.2. Сеть «Аутстар» Гроссберга

### Обучение входной звезды

Входная звезда выполняет распознавание образов, т. е. она обучается реагировать на определенный входной вектор  $X$  и ни на какой другой. Это обучение реализуется путем настройки весов таким образом, чтобы они соответствовали входному вектору. Выход входной звезды определяется как взвешенная сумма ее входов, как это описано в предыдущих разделах. С другой точки зрения, выход можно рассматривать как свертку входного вектора с весовым вектором, меру сходства нормализованных векторов. Следовательно, нейрон должен реагировать наиболее сильно на входной образ, которому был обучен.

Процесс обучения выражается следующим образом:

$$w_i(t+1) = w_i(t) + \alpha[x_i - w_i(t)],$$

где  $w_i$  – вес входа  $x_i$ ;  $x_i$  –  $i$ -й вход;  $\alpha$  – нормирующий коэффициент обучения, который имеет начальное значение 0,1 и постепенно уменьшается в процессе обучения.

После завершения обучения предъявление входного вектора  $X$  будет активизировать обученный входной нейрон. Это можно рассматривать как единый обучающий цикл, если  $\alpha$  установлен в 1, однако в этом случае исключается способность входной звезды к обобщению. Хорошо обученная входная звезда будет реагировать не только на определенный единичный вектор, но также и на незначительные изменения этого вектора. Это достигается постепенной настройкой нейронных весов при предъявлении в процессе обучения векторов, представляющих нормальные вариации входного вектора. Веса настраиваются таким образом, чтобы усреднить величины обучающих векторов, и нейроны получают способность реагировать на любой вектор этого класса.

### Обучение выходной звезды

В то время как входная звезда возбуждается всякий раз при появлении определенного входного вектора, выходная звезда имеет дополнительную функцию; она вырабатывает требуемый возбуждающий сигнал для других нейронов всякий раз, когда возбуждается.

Для того чтобы обучить нейрон выходной звезды, его веса настраиваются в соответствии с требуемым целевым вектором. Алгоритм обучения может быть представлен символически следующим образом:

$$w_i(t+1) = w_i(t) + \beta[y_i - w_i(t)],$$

где  $\beta$  представляет собой нормирующий коэффициент обучения, который в начале приблизительно равен единице и постепенно уменьшается до нуля в процессе обучения.

Как и в случае входной звезды, веса выходной звезды, постепенно настраиваются над множеством векторов, представляющих собой обычные вариации идеального вектора. В этом случае выходной сигнал нейронов представляет собой статистическую характеристику обучающего набора и может в действительности сходиться в процессе обучения к идеальному вектору при предъявлении только искаженных версий вектора.

## ОБУЧЕНИЕ ПЕРСЕПТРОНА

В 1957 г. Розенблатт [4] разработал модель, которая вызвала большой интерес у исследователей. Несмотря на некоторые ограничения ее исходной формы, она стала основой для многих современных, наиболее сложных алгоритмов обучения с учителем. Перцептрон является настолько важным, что вся гл. 2 посвящена его описанию; однако это описание является кратким и приводится в формате, несколько отличном от используемого в [4].

Перцептрон является двухуровневой, нерекurrentной сетью, вид которой показан на рис. Б.3. Она использует алгоритм обучения с учителем; другими словами, обучающая выборка состоит из множества входных векторов, для каждого из которых указан свой требуемый вектор цели. Компоненты входного вектора представлены непрерывным диапазоном значений; компоненты вектора цели являются двоичными величинами (0 или 1). После обучения сеть получает на входе набор непрерывных входов и выработывает требуемый выход в виде вектора с бинарными компонентами.

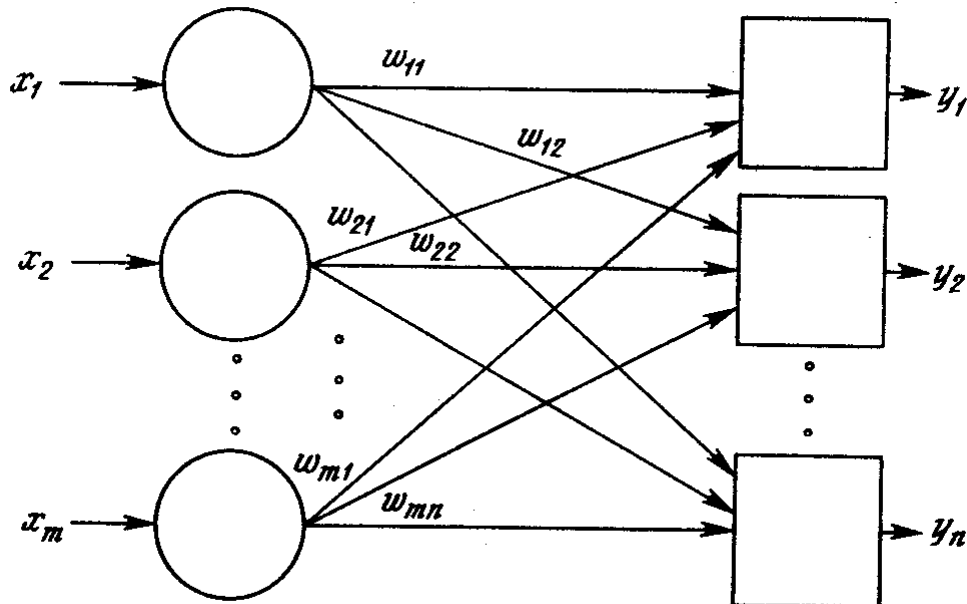


Рис. Б.3. Однослойная нейронная сеть

Обучение осуществляется следующим образом:

1. Рандомизируются все веса сети в малые величины.
2. На вход сети подается входной обучающий вектор  $X$  и вычисляется сигнал NET от каждого нейрона, используя стандартное выражение

$$NET_j = \sum_i x_i w_{ij}.$$

3. Вычисляется значение пороговой функции активации для сигнала NET от каждого нейрона следующим образом:

$$\begin{aligned} OUT_j &= 1, && \text{если } NET_j \text{ больше чем порог } \theta_j, \\ OUT_j &= 0 && \text{в противном случае.} \end{aligned}$$

Здесь  $\theta_j$  представляет собой порог, соответствующий нейрону  $j$  (в простейшем случае, все нейроны имеют один и тот же порог).

4. Вычисляется ошибка для каждого нейрона посредством вычитания полученного выхода из требуемого выхода:

$$error_j = target_j - OUT_j.$$

5. Каждый вес модифицируется следующим образом:

$$W_{ij}(t+1) = w_{ij}(t) + \alpha x_i error_j.$$

6. Повторяются шаги со второго по пятый до тех пор, пока ошибка не станет достаточно малой.

## МЕТОД ОБУЧЕНИЯ УИДРОУ-ХОФФА

Как мы видели, персептрон ограничивается бинарными выходами. Уидроу вместе со студентом университета Хоффом расширили алгоритм обучения персептрона на случай непрерывных выходов, используя сигмоидальную функцию [5,6]. Кроме того, они разработали математическое доказательство того, что сеть при определенных условиях будет сходиться к любой функции, которую она может представить. Их первая модель – Адалин – имеет один выходной нейрон, более поздняя модель – Мадалин – расширяет ее на случай с многими выходными нейронами.

Выражения, описывающие процесс обучения Адалина, очень схожи с персептронными. Существенные отличия имеются в четвертом шаге, где используются непрерывные сигналы NET вместо бинарных OUT. Модифицированный шаг 4 в этом случае реализуется следующим образом:

4. Вычисляется ошибка для каждого нейрона посредством вычитания полученного выхода из требуемого выхода:

$$\text{error}_j = \text{target}_j - \text{NET}_j.$$

## МЕТОДЫ СТАТИСТИЧЕСКОГО ОБУЧЕНИЯ

В гл. 5 детально описаны статистические методы обучения, поэтому здесь приводится лишь обзор этих методов.

Однослойные сети несколько ограничены с точки зрения проблем, которые они могут решать; однако в течение многих лет отсутствовали методы обучения многослойных сетей. Статистическое обучение обеспечивает путь решения этих проблем.

По аналогии обучение сети статистическими способами подобно процессу отжига металла. В процессе отжига температура металла вначале повышается, пока атомы металла не начнут перемещаться почти свободно. Затем температура постепенно уменьшается и атомы непрерывно стремятся к минимальной энергетической конфигурации. При некоторой низкой температуре атомы переходят на низший энергетический уровень.

В искусственных нейронных сетях полная величина энергии сети определяется как функция определенного множества сетевых переменных. Искусственная переменная температуры иницируется в большую величину, тем самым позволяя сетевым переменным претерпевать большие случайные изменения. Изменения, приводящие к уменьшению полной энергии сети, сохраняются; изменения, приводящие к увеличению энергии, сохраняются в соответствии с вероятностной функцией. Искусственная температура постепенно уменьшается с течением времени и сеть конвергирует в состояние минимума полной энергии.

Существует много вариаций на тему статистического обучения. Например, глобальная энергия может быть определена как средняя квадратичная ошибка между полученным и желаемым выходным вектором из обучаемого множества, а переменными могут быть веса сети. В этом случае сеть может быть обучена, начиная с высокой искусственной температуры, путем выполнения следующих шагов:

1. Подать обучающий вектор на вход сети и вычислить выход согласно соответствующим сетевым правилам.
2. Вычислить значение средней квадратичной ошибки между желаемым и полученным выходными векторами.
3. Изменить сетевые веса случайным образом, затем вычислить новый выход и результирующую ошибку. Если ошибка уменьшилась, оставить измененный вес; если ошибка увеличилась, оставить измененный вес с вероятностью, определяемой распределением Больцмана. Если изменения весов не производится, то вернуть вес к его предыдущему значению.
4. Повторить шаги с 1 по 3, постепенно уменьшая искусственную температуру.

Если величина случайного изменения весов определяется в соответствии с распределением Больцмана, сходимость к глобальному минимуму будет осуществляться только в том случае, если температура изменяется обратно пропорционально логарифму прошедшего времени обучения. Это может привести к невероятной длительности процесса обучения, поэтому большое внимание уделялось поиску более быстрых методов обучения. Выбором размера шага в соответствии с распределением Коши может быть достигнуто уменьшение температуры, обратно пропорциональное обучающему времени, что существенно уменьшает время, требуемое для сходимости.

Заметим, что существует класс статистических методов для нейронных сетей, в которых переменными сети являются выходы нейронов, а не веса. В гл. 5 эти алгоритмы рассматривались подробно.

## САМООРГАНИЗАЦИЯ

В работе [3] описывались интересные и полезные результаты исследований Кохонена на самоорганизующихся структурах, используемых для задач распознавания образов. Вообще эти структуры классифицируют образы, представленные векторными величинами, в которых каждый компонент вектора соответствует элементу образа. Алгоритмы Кохонена основываются на технике обучения без учителя. После обучения подача входного вектора из данного класса будет приводить к выработке возбуждающего уровня в каждом выходном нейроне; нейрон с максимальным возбуждением представляет

классификацию. Так как обучение проводится без указания целевого вектора, то нет возможности определять заранее, какой нейрон будет соответствовать данному классу входных векторов. Тем не менее это планирование легко проводится путем тестирования сети после обучения.

Алгоритм трактует набор из  $n$  входных весов нейрона как вектор в  $n$ -мерном пространстве. Перед обучением каждый компонент этого вектора весов инициализируется в случайную величину. Затем каждый вектор нормализуется в вектор с единичной длиной в пространстве весов. Это делается делением каждого случайного веса на квадратный корень из суммы квадратов компонент этого весового вектора.

Все входные вектора обучающего набора также нормализуются и сеть обучается согласно следующему алгоритму:

1. Вектор  $\mathbf{X}$  подается на вход сети.
2. Определяются расстояния  $D_j$  (в  $n$ -мерном пространстве) между  $\mathbf{X}$  и весовыми векторами  $\mathbf{W}_j$  каждого нейрона. В евклидовом пространстве это расстояние вычисляется по следующей формуле

$$D_j = \sqrt{\sum_i (x_i - w_{ij})^2},$$

где  $x_i$  – компонента  $i$  входного вектора  $\mathbf{X}$ ,  $w_{ij}$  – вес входа  $i$  нейрона  $j$ .

3. Нейрон, который имеет весовой вектор, самый близкий к  $\mathbf{X}$ , объявляется победителем. Этот весовой вектор, называемый  $\mathbf{W}_c$ , становится основным в группе весовых векторов, которые лежат в пределах расстояния  $D$  от  $\mathbf{W}_c$ .
4. Группа весовых векторов настраивается в соответствии со следующим выражением:

$$\mathbf{W}_j(t+1) = \mathbf{W}_j(t) + \alpha[\mathbf{X} - \mathbf{W}_j(t)]$$

для всех весовых векторов в пределах расстояния  $D$  от  $\mathbf{W}_c$

5. Повторяются шаги с 1 по 4 для каждого входного вектора.

В процессе обучения нейронной сети значения  $D$  и  $\alpha$  постепенно уменьшаются. Автор [3] рекомендовал, чтобы коэффициент  $\alpha$  в начале обучения устанавливался приблизительно равным 1 и уменьшался в процессе обучения до 0, в то время как  $D$  может в начале обучения равняться максимальному расстоянию между весовыми векторами и в конце обучения стать настолько маленьким, что будет обучаться только один нейрон.

В соответствии с существующей точкой зрения, точность классификации будет улучшаться при дополнительном обучении. Согласно рекомендации Кохонена, для получения хорошей статистической точности количество обучающих циклов должно быть, по крайней мере, в 500 раз больше количества выходных нейронов.

Обучающий алгоритм настраивает весовые векторы в окрестности возбужденного нейрона таким образом, чтобы они были более похожими на входной вектор. Так как все векторы нормализуются в векторы с единичной длиной, они могут рассматриваться как точки на поверхности единичной гиперболы. В процессе обучения группа соседних весовых точек перемещается ближе к точке входного вектора. Предполагается, что входные векторы фактически группируются в классы в соответствии с их положением в векторном пространстве. Определенный класс будет ассоциироваться с определенным нейроном, перемещая его весовой вектор в направлении центра класса и способствуя его возбуждению при появлении на входе любого вектора данного класса.

После обучения классификация выполняется посредством подачи на вход сети испытуемого вектора, вычисления возбуждения для каждого нейрона с последующим выбором нейрона с наивысшим возбуждением как индикатора правильной классификации.

## Литература

1. Grossberg S. 1974. Classical and instrumental learning by neural networks. Progress in theoretical biology, vol. 3, pp. 51–141. New York: Academic Press.
2. Hebb D. O. 1949. Organization of behavior. New York: Science Editions.
3. Kohonen T. 1984. Self-organization and associative memory. Series in Information Sciences, vol. 8. Berlin: Springer verlag.
4. Rosenblatt R. 1959. Principles of neurodynamics. New York: Spartan Books.
5. Widrow B. 1959 Adaptive sampled-data systems, a statistical theory of adaptation. 1959. IRE WESCON Convention Record, part 4. New York: Institute of Radio Engineers.
6. Widrow B., Hoff M. 1960. Adaptive switching circuits. 1960. IRE WESCON Convention Record. New York: Institute of Radio Engineers.