

Л. Н. Ясницкий

Э Л Е К Т И В Н Ы Й К У Р С

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

Учебное пособие



ИЗДАТЕЛЬСТВО

БИНОМ

УДК 62.506.222.001.57+539.3
ББК 32.81
Я82

Ясницкий Л. Н.

Я82 Искусственный интеллект. Элективный курс : учебное пособие / Л. Н. Ясницкий. — М. : БИНОМ. Лаборатория знаний, 2011. — 197 с. : ил.

ISBN 978-5-9963-0234-5

Учебное пособие является составной частью учебно-методического комплекса, включающего также методическое пособие «Искусственный интеллект» и лабораторный практикум, размещенный на сайте <http://www.LbAI.ru>. Задача курса — познакомить школьников с историей искусственного интеллекта, с основными стратегиями, применяемыми при создании интеллектуальных информационных систем, научить пользоваться нейросетевыми технологиями и применять их для решения практических проблем.

Для учащихся старших классов информационно-технологического, физико-математического, естественно-научного, социально-гуманитарного профилей.

УДК 62.506.222.001.57+539.3
ББК 32.81

По вопросам приобретения обращаться:
«БИНОМ. Лаборатория знаний»
Телефон: (499) 157-5272
e-mail: binom@Lbz.ru, <http://www.Lbz.ru>

ISBN 978-5-9963-0234-5

© БИНОМ. Лаборатория знаний,
2011

ОГЛАВЛЕНИЕ

Предисловие	5
Глава 1. Прошлые и настоящее искусственного интеллекта	9
1.1. История искусственного интеллекта	9
1.2. Основные стратегии искусственного интеллекта	18
1.3. Искусственный интеллект как лидирующая научная отрасль	21
1.4. Основные сферы применения искусственного интеллекта	22
Глава 2. Экспертные системы	29
2.1. Данные и знания	29
2.2. Способы представления знаний	31
2.3. Технология экспертных систем	37
Глава 3. Нейронные сети	48
3.1. Персептрон и его развитие	48
3.2. Возможности и сферы применения персептронов	92
3.3. Проблемы проектирования и применения персептронов	133
3.4. Заключительные замечания и выводы	151
Глава 4. Интеллектуальные игры	155
4.1. Понятие игры и дерева возможностей	155
4.2. Методы подрезки дерева возможностей	157
4.3. Идеи обучения игровых программ	163

Глава 5. Моделирование творчества	169
5.1. Философские аспекты творчества	169
5.2. Музыкальное творчество	173
5.3. Поэтическое творчество	177
Глава 6. Настоящее и будущее искусственного	
интеллекта	181
6.1. Искусственный интеллект	
и интеллектуальные информационные	
системы	181
6.2. Нейронные сети и экспертные системы	183
6.3. Нейросетевые технологии и метод	
математического моделирования	185
6.5. Философские проблемы искусственного	
интеллекта	187
Терминологический словарь	191
Литература	195

ПРЕДИСЛОВИЕ

«Искусственный интеллект». Трудно назвать иной термин, имеющий к себе столь противоречивое отношение. Этот термин может вызывать уважение, крайнюю степень заинтересованности и любопытства. А может, наоборот, служить поводом для иронии, скептицизма, шуток и насмешек.

Тем не менее события последних лет убедительно показывают: компьютеры «умнеют» буквально на глазах! Интеллектуальные компьютерные технологии проникают практически во все области деятельности человека, и этот процесс постоянно расширяется и усиливается. Поэтому теперешним школьникам — завтрашним выпускникам — крайне необходимо овладеть определенным минимумом информационно-компьютерной компетентности, в том числе фундаментальными знаниями в области искусственного интеллекта.

Искусственный интеллект — это некий «запретный плод», который манит к себе молодых творческих людей, желающих создавать новое, познавать мир и самих себя. Он — как линия горизонта, которая видна, но к которой невозможно приблизиться. Это неиссякаемый источник философских споров, неисчерпаемая тема для писателей-фантастов, никогда не теряющая актуальности область научных исследований.

Автор этой книги уже много лет занимается разработкой интеллектуальных информационных систем, читает лекции по тематике искусственного интеллекта в нескольких университетах, проводит экспериментальные занятия в школах и лицеях. По его личным наблюдениям существу-

ют две категории людей, интересующихся и занимающихся искусственным интеллектом.

Первая категория таких людей ставит перед собой задачу — разобраться в самом смысле человеческого бытия и в принципах мироздания, ответить на вопросы: как устроен человеческий мозг, в чем смысл жизни человека, каково его предназначение, как и откуда он появился на планете Земля и что с ним будет дальше?

Для второй категории, наоборот, характерен прагматизм. Они применяют методы искусственного интеллекта в самых что ни на есть практических целях, поняв, что добиться высокой эффективности компьютерных программ можно путем копирования, моделирования или имитации принципов деятельности человеческого мозга. Такие специалисты изучают мозг, выдвигают новые идеи и гипотезы, реализуют и проверяют их в работе при решении конкретных производственных задач. Термин «искусственный интеллект» они стараются не использовать, оставляя его философам и фантастам. Их прагматическая терминология звучит более приземленно: «интеллектуальная информационная система» или «система искусственного интеллекта».

Поместив в название книги более традиционный термин «Искусственный интеллект», автор отдает дань легендарному прошлому этой замечательной научной области, понимая, что именно из этого прошлого произрастают воистину грандиозное настоящее и уже отчетливо просматривающееся будущее искусственного интеллекта. Это будущее интуитивно чувствует современная «продвинутая» молодежь, в том числе школьного возраста. Не дожидаясь обновления школьных учебных программ, многие старшеклассники-вундеркинды уже сейчас самостоятельно делают свои первые «заплывы» в «море» искусственного интеллекта, черпая информацию из сети Интернет и научно-популярных книг. Именно для таких школьников в первую очередь предназначен данный элективный учебный курс.

Особенность этого курса состоит в том, что в нем затрагиваются весьма серьезные теоретические вопросы философского плана, но в то же время он является популярным

руководством по применению идей и методов искусственного интеллекта для решения конкретных практически важных задач.

Для вас, уважаемые школьники, это своего рода трамплин, стартовая площадка, с которой вы можете начать самореализацию в жизни, расширить свой кругозор, выработать собственный взгляд, свое понимание жизни и жизненных процессов и в то же время освоить и научиться применять самые современные компьютерные технологии.

Основная цель курса — показать, что искусственный интеллект — это вовсе не что-то фантастическое и нереальное, а вполне конкретный и нужный предмет, и что моделировать человеческий мозг совсем не сложно, это может делать каждый школьник для решения реальных практических задач, встречающихся в его жизни:

- какую профессию выбрать и куда пойти учиться?
- сколько процентов билетов выучить, чтобы получить хорошую оценку на экзамене?
- кого выбрать себе в спутники (спутницы) жизни?
- когда обратиться к родителям с просьбой, чтобы не получить отказ?
- как определить, правду ли говорит собеседник?
- какая команда победит в спортивных состязаниях?
- какие шаги предпринять, чтобы улучшить свои спортивные результаты?
- каким завтра (или в ближайшую неделю) будет курс валюты?
- кто победит на выборах и как можно повлиять на их результат?

и многое другое.

Таким образом, цель данного курса — научить вас, уважаемые школьники, как сделать из вашего компьютера доброго советчика, помогающего принимать правильные решения и строить прогнозы развития различных событий и ситуаций.

Глубокому изучению предмета способствует выполнение *лабораторных работ*. В их основу заложена идея азартной компьютерной игры, так что их выполнение будет проходить практически без вмешательства учителя или лаборанта. В ходе выполнения этих лабораторных работ вы освоите специально созданный для вас программный инструмент — *симулятор нейронных сетей*. С его помощью вы сможете создавать нейронные сети, подбирать их параметры, обучать эти сети решению множества задач из областей медицины, политологии, социологии, спорта, туризма, криминалистики, педагогики, экономики, промышленности и др. Вам будет предоставлена и возможность самим придумывать задачи и решать их методами искусственного интеллекта. А самым любознательным и нетерпеливым мы рекомендуем, не дожидаясь начала уроков, посетить сайт <http://www.LbAI.ru>, скачать на свой компьютер лабораторные работы и попробовать выполнить их, самостоятельно освоив этот, в общем-то, не очень сложный для современных молодых людей предмет.

ГЛАВА 1

ПРОШЛОЕ И НАСТОЯЩЕЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

1.1. История искусственного интеллекта

Человек — это самый сложный из доступных для нашего восприятия объектов, а способность мышления — его главное свойство. *Искусственный интеллект* — это наука, ставящая своей целью изучение и моделирование главного свойства человека — *мышления*.

Какова природа мышления? Какие процессы происходят в нашем мозге, когда мы думаем, чувствуем, видим, понимаем? Что такое творчество и можно ли его моделировать? Возможно ли в принципе понять, как работает наш мозг, и заставить мыслить неживую материю? На протяжении тысячелетий человек задавался этими вопросами, но до сих пор, а возможно, и никогда мы не сможем на них ответить с полной определенностью.

История попыток создания искусственного подобия человеческого разума насчитывает более 700 лет. Первую попытку создания машины, моделирующей человеческий разум, связывают с именем испанского рыцаря, поэта, философа, богослова, алхимика и изобретателя Раймунда Луллия (рис. 1.1).

Огромный интерес представляет сама личность этого человека. Любимец короля, дуэлянт и повеса, как о нем пишут историки, рыцарь Раймунд Луллий вдруг отказался от светских развлечений и ушел в монастырь, чтобы стать мудрецом. Согласно легенде, это его решение было продиктовано благородной целью — постичь науки и с их помощью излечить от тяжелой болезни свою возлюбленную.

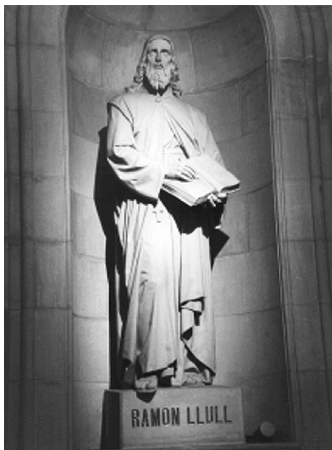


Рис. 1.1. Статуя Раймунда Луллия в Барселонском университете

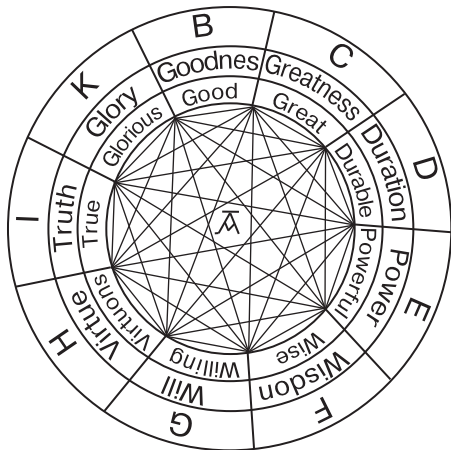


Рис. 1.2. Один из кругов логической машины Р. Луллия

К сожалению, истории неизвестно, удалось ли Луллию достичь своей высокой цели. Известно только, что в возрасте 80 лет его во время спора насмерть забили камнями его оппоненты — те, кто были не согласны с учением Луллия...

Раймунд Луллий родился в 1235 и умер в 1315 г. В его времена ученые были заняты поиском неких универсальных понятий и истин, которые, будучи связанными между собой, давали бы общую картину мироздания, а значит и ответы на все интересующие человечество вопросы. Это был век философов-мудрецов, астрологов и алхимиков, занятых поисками философского камня.

Развивая традиции ученых своего времени, Луллий сконструировал механическое устройство, состоящее из системы вращающихся кругов. Каждый такой круг (рис. 1.2) был поделен на секторы, окрашенные в разные цвета и помеченные латинскими буквами. Круги соединялись друг с другом, и, приводя их во вращение, можно было получать различные сочетания символов и цветов — так называемую «формулу истины».

Машины Луллия могли давать ответы на всевозможные вопросы, составлять гороскопы, ставить диагнозы бо-

лезней, делать прогнозы на урожай. В наиболее позднем варианте машина Луллия состояла из 14 кругов, размеченных буквами и раскрашенных в различные цвета, которые символизировали различные понятия, элементы, стихии, субъекты и объекты знания. Круги приводились в движение системой рычагов. Поворачиваясь, они могли образовывать около 18 квадриллионов (18×10^{15}) разнообразных сочетаний буквенных и цветовых «истин». Запросы в эту машину вводились с помощью поворота внутреннего круга, на котором было надписано девять вариантов вопросов: «Что?», «Почему?», «Из чего?», «Сколько?», «Каким образом?», «Где?», «Когда?», «Какое?», «Которое из двух?».

Говоря современным языком, можно сказать, что машина Луллия представляла собой механическую экспертную систему, наделенную базой знаний, устройствами ввода/вывода и естественным языком общения. Свести к логическим операциям если не все знания о мире, то хотя бы часть из них, а затем поручить не человеческому мозгу, а механическому устройству процедуру вывода «формул знания», следующих из накопленной базы знаний, — эта идея искусственного интеллекта, впервые высказанная и реализованная средневековым рыцарем Раймундом Луллем, прожила семь веков и достигла в наши дни своего расцвета.

В 1940-х гг. искусственный интеллект был выделен в самостоятельное научное направление. А в 1956 г. произошло событие, благодаря которому этот год называют «второй датой рождения» искусственного интеллекта. В этом году в Стэнфордском университете (рис. 1.3) состоялся семинар под названием «Artifical intelligence», что в переводе означает «Искусственный интеллект». Именно тогда был впервые предложен этот термин.

С тех пор история искусственного интеллекта представляет собой постоянные споры и метания между двумя крайностями — оптимизмом и пессимизмом. Интересны знаменитые предсказания американского экономиста и социолога, исследователя в области теории управления и моделирования социальных процессов, нобелевского лауреа-



Рис. 1.3. Стэнфордский университет (США), где в 1956 г. прошел семинар под названием «Artificial intelligence» — «Искусственный интеллект»

та Герберта Саймона, сделанные им в 1957 г. Приведем несколько из этих предсказаний:

- в ближайшее десятилетие ЭВМ завоюет титул чемпиона мира по шахматам;
- в пределах десяти лет ЭВМ откроет и сумеет доказать важную новую математическую теорему;
- в десятилетний срок большинство теорий в области психологии примет вид программ для вычислительной машины.

Сегодня, спустя полвека, мы видим, что предсказания Г. Саймона постепенно сбываются, что он ошибался только в сроках. Мы также можем отметить, что эйфория вокруг искусственного интеллекта имела как положительные, так и отрицательные последствия. С одной стороны, она стимулировала интерес общественности к новому научному направлению, выразившийся в выделении ученым крупных грантов от правительства США. С другой стороны, искусственный интеллект стал объектом весьма резкой критики

более «трезвомыслящих» ученых. Мы также знаем, к каким тяжелым последствиям привела эта критика в СССР, когда за решение проблем научных дискуссий взялся государственный репрессивный аппарат: кибернетика была объявлена «буржуазной лженаукой», а те, кто ей занимался — «врагами народа».

Вскоре после признания искусственного интеллекта самостоятельной отраслью науки произошло ее разделение на два основных стратегических направления: *нейрокибернетику* и *кибернетику «черного ящика»*. Подход к созданию систем искусственного интеллекта, используемый нейрокибернетикой, иногда называют *низкоуровневым*, или *восходящим*, а подход, используемый кибернетикой «черного ящика», — *высокоуровневым*, или *нисходящим*. Названия этих стратегических направлений соответствуют их содержанию, что станет ясно в дальнейшем.

Основную идею нейрокибернетики можно сформулировать следующим образом: «Единственный объект в природе, способный мыслить, — это человеческий мозг, поэтому любое мыслящее устройство должно быть обязательно выполнено по образу и подобию человеческого мозга, воспроизводить его структуру и принцип действия». Таким образом, нейрокибернетика занимается моделированием самой структуры мозга и его деятельности.

Как известно, мозг человека состоит из большого количества взаимосвязанных нервных клеток — *нейронов*, поэтому усилия нейрокибернетиков сосредоточены на разработке элементов, подобных нейронам, и на объединении этих элементов в системы — *нейросети* и *нейрокомпьютеры*. Первые нейросети и нейрокомпьютеры были созданы американскими учеными — Уорреном Мак-Каллоком, Вальтером Питтсом и Фредериком Розенблаттом в конце 1950-х гг. Это были устройства, моделирующие человеческий глаз и его взаимодействие с мозгом, которые умели распознавать буквы латинского алфавита.

В отличие от нейрокибернетики, кибернетика «черного ящика» не придает значения структуре и принципу дей-

ствия мыслящего устройства. Главное, чтобы оно адекватно моделировало самый высокий уровень деятельности мозга — уровень его интеллектуальных функций. Это направление искусственного интеллекта ориентировано на поиск алгоритмов решения интеллектуальных задач с использованием существующих компьютеров независимо от их аппаратной базы.

Поставив перед собой задачу моделирования функций мозга, представители этого научного направления столкнулись с серьезной проблемой. Оказалось, что несмотря на многовековую историю исследований ни одна из существующих наук (философия, психология, лингвистика и др.) не смогла предложить сколько-нибудь конкретный алгоритм человеческого мышления, поэтому программистам пришлось создавать собственные модели мышления.

В конце 1950-х гг. появилась *модель лабиринтного поиска*. Согласно этому подходу, решение интеллектуальной задачи выполнялось путем перебора огромного количества вариантов, который представлялся в виде движения по некоему лабиринту. В настоящее время модель лабиринтного поиска признается тупиковой и ограниченно используется в игровых компьютерных программах.

В начале 1960-х гг. началась эпоха *эвристического программирования*. Как писал автор этого термина, американский математик П. Пойа, «цель эвристики — исследовать методы и правила того, как делать открытия и изобретения».

Это очень сложная проблема. К сожалению, Архимед, выпрыгнувший из ванны с криком «Эврика!», никому не объяснял, как он догадался, что тело, погруженное в жидкость, теряет в своем весе ровно столько, сколько весит вытесненный им объем воды. Ньютон открыл свой закон всемирного тяготения, наблюдая за падением яблока. Менделеев увидел принцип построения периодической таблицы во сне. Поэтов, художников и музыкантов вдохновляют на творческие поиски возвышенные чувства, алгоритмизировать которые невозможно в принципе.

Чтобы понять механизмы творческого мышления, авторы эвристического подхода провели следующий экспе-

римент. Была отобрана группа студентов, не знакомых с математической логикой. Каждый студент должен был самостоятельно доказать одну или несколько теорем из учебника, не заглядывая в него. При этом он должен был рассуждать вслух, делать любые записи, прекращать работу, если становилось ясно, что выбран неверный путь, и начинать все сначала. Обработав полученные магнитофонные записи, выкладки и черновики студентов, программисты нашли в них *эвристики* — некие правила, которыми пользовались студенты, доказывая теоремы. Эти правила не имели строгих математических обоснований, а их использование не всегда приводило к желаемому результату. Однако программа, в которую заложили эвристики студентов, смогла доказать все теоремы, которые имелись в учебнике, и дополнительно сформулировать еще две теоремы, которых не хватало до полной логической завершенности курса! Теперь эту программу, известную под названием «Логик-теоретик», принято считать родоначальницей *эвристического программирования*.

Ясно, что успехи нисходящей стратегии связаны с развитием возможностей ЭВМ и искусства программирования, т. е. с успехами целого комплекса научных исследований, который называют «компьютерными науками». В противоположность этому успехи восходящей стратегии (от низкого уровня к высокому) связаны с успехами наук о человеке.

Серьезный прорыв в практических приложениях нисходящего подхода к созданию систем искусственного интеллекта произошел в середине 1970-х гг., когда, отказавшись от поисков универсального алгоритма мышления, программисты начали моделировать конкретные знания специалистов-экспертов. Тем самым появилось новое направление искусственного интеллекта — *экспертные системы*.

С появлением экспертных систем бизнес в сфере интеллектуальных информационных технологий впервые начал становиться рентабельным. С середины 1980-х гг. искусственный интеллект превратился в одно из наиболее привлекательных в коммерческом отношении направлений

компьютерной индустрии. Постоянно растут ежегодные капиталовложения, создаются промышленные и военные экспертные системы.

Однако победа, одержанная приверженцами нисходящего подхода в 1980-х гг., оказалась неокончательной. Разобравшись в причинах своих неудач (подробности об этом см. в главе 3) и выполнив ряд серьезных фундаментальных исследований, ученые-нейрокибернетики вышли на рынок программной продукции с *нейронными сетями*. В результате 1990-е гг. ознаменовались приходом искусственных нейронных сетей в бизнес, где они показали свою реальную эффективность при решении множества задач — от анализа платежеспособности клиентов банка до прогнозирования курсов валют и предсказания результатов президентских выборов.

В последнее время, наряду с указанными выше двумя стратегическими подходами к проблеме моделирования мышления, активно развивается третий подход, называемый *эволюционным*. Смысл его состоит в том, что процесс моделирования человеческого мозга заменяется моделированием процесса его эволюции. Природные эволюционные процессы, описываемые теорией Чарлза Дарвина, кладутся в основу поведения простейших существ — *интеллектуальных агентов*, не имеющих тел, но способных чувствовать, действовать, бороться за существование, размножаться, развиваться и жить, как это происходит в реальном физическом мире.

В заключение отметим, что исследования по искусственному интеллекту всегда находились на переднем крае информатики. Теории, идеи и методы искусственного интеллекта всегда привлекали и, вероятно, еще долго будут привлекать внимание тех, кто стремится расширить область применения и возможности компьютеров, сделать их более «дружественными», более похожими на разумных существ, с которыми приятно общаться и работать.



Коротко о главном

Первая попытка создания устройства, моделирующего человеческий интеллект, была произведена в XIII в. испанским ученым Р. Луллием. Термин «искусственный интеллект» был предложен в 1956 г. на семинаре, проводимом в Стэнфордском университете (США). Позже искусственный интеллект распался на два направления: кибернетику «черного ящика» и нейрокибернетику. Первый нейрокомпьютер был предложен и создан американскими учеными У. Мак-Каллоком, В. Питтсом и Ф. Розенблаттом в конце 1950-х гг.



Вопросы и задания

1. Попробуйте самостоятельно сформулировать определения интеллекта и искусственного интеллекта.
2. Когда и кем была произведена первая попытка моделирования интеллекта человека?
3. Опишите назначение и принцип действия машины Р. Луллия.
4. Когда и где впервые был использован термин «искусственный интеллект»?
5. В чем состоит принципиальное различие между кибернетикой «черного ящика» и нейрокибернетикой?
6. Каков смысл терминов: «низкоуровневое», «восходящее», «высокоуровневое» и «нисходящее» направление искусственного интеллекта?
7. Какие задачи решал первый нейрокомпьютер?
8. Кто был создателем первого нейрокомпьютера?

1.2. Основные стратегии искусственного интеллекта

Как уже было сказано, искусственный интеллект — это научная отрасль, занимающаяся исследованием и моделированием естественного интеллекта человека. Однако сам естественный интеллект человека является очень сложным объектом исследований, и его моделирование осуществляется на разных уровнях абстрагирования. Можно выделить три таких уровня, которым соответствуют три основных стратегических направления (в дальнейшем мы будем называть их *стратегиями*) искусственного интеллекта (рис. 1.4):

- технологии экспертных систем (высокоуровневая стратегия);
- технологии эволюционного моделирования;
- нейроинформационные технологии (низкоуровневая стратегия).

Самому высокому уровню абстрагирования соответствуют *технологии экспертных систем*, основанных на явных знаниях о предметной области. Согласно этой стратегии, знания о предметной области могут быть получены от специалистов-экспертов или из других информационных источников. Именно человек-эксперт, обладающий знаниями о предметной области, способен действовать на

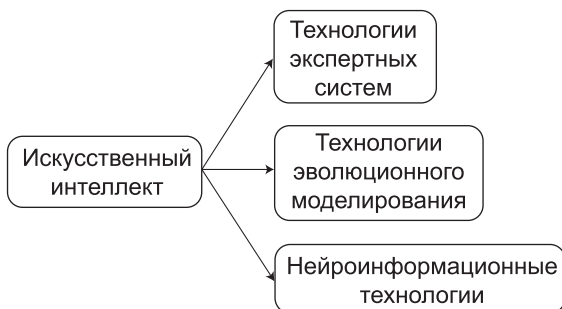


Рис. 1.4. Основные стратегии искусственного интеллекта

самом высоком уровне. Его знания формализуются, т. е. представляются в удобном для использования виде и закладываются в программный комплекс — *экспертную систему*, которая на основе этих знаний делает затем логически обоснованные полезные выводы, тем самым моделируя интеллект человека-эксперта. (Кстати, именно эту стратегию впервые применил в XIII в. испанский ученый Раймунд Луллий и именно с нее берет начало область науки, называемая искусственным интеллектом.)

На самом нижнем уровне абстрагирования находятся нейросетевые технологии. Согласно этой стратегии строится модель, учитывающая структуру мозга, который состоит из множества нейронов, соединенных нервными волокнами. Знания поступают в такую модель в ходе ее *обучения* на специально подобранных примерах, характеризующих конкретную предметную область. Так же, как и в мозге, знания хранятся здесь в неявной форме — кодируются в виде множества *сил синаптических связей* (об этом пойдет речь в главе 3), имитирующих электропроводность межнейронных соединений. Основоположниками этой стратегии, появившейся в середине XX в., являются американские ученые У. Мак-Каллок, В. Питтс и Ф. Розенблатт.

Исторически первой была модель искусственного интеллекта, основанного на явных знаниях, — технология экспертных систем, которая безраздельно господствовала с XIII до середины XX в. Возникновение в 1940-х гг. альтернативной стратегии и появление первого нейрокомпьютера, названного *перцептроном* (*перцептроном*), равно как и первые его успехи, были встречены с большими надеждами и энтузиазмом, который, впрочем, продержался недолго. К середине 1980-х гг. подавляющее большинство систем искусственного интеллекта представляли собой экспертные системы, ориентированные на конкретные предметные области. Однако победа высокоуровневой стратегии над низкоуровневой оказалась временной. Конец XX в. и начало XXI в. ознаменовались широкомасштабным наступлением нейросетевых и нейрокомпьютерных техноло-

гий, причем имело место как применение этих технологий в новых предметных областях, так и вытеснение экспертных систем из тех областей, где они традиционно применялись ранее.

Уже в течение полувека между высоко- и низкоуровневой стратегиями искусственного интеллекта идет жесточайшая конкуренция. Вместе с тем многие специалисты отмечают наметившийся в последнее время процесс интеграции этих конкурирующих стратегий. Известны, например, успешные попытки создания и применения *гибридных интеллектуальных систем*, комбинирующих идеи обеих этих альтернативных стратегий.

Третья стратегия, появившаяся на свет в середине 1970-х гг. благодаря работам профессора Мичиганского университета Дж. Холланда, навеяна Дарвиновской теорией возникновения и эволюции жизни на Земле. *Эволюционные модели* начинают свою работу с создания целой популяции особей — кандидатов на решение проблемы. Отдельные особи этой популяции оцениваются по определенному критерию, позволяющему отобрать из них лучшие, которые затем скрещиваются, наследуют положительные свойства родителей и, мутируя, образуют новые поколения. Этот подход предполагает поиск источника интеллекта в самом процессе эволюции и взаимодействия особей. Новые знания же в эволюционных моделях проявляются в ходе конкурентной борьбы между особями.

Подводя итог краткого обзора существующих стратегий искусственного интеллекта, отметим, что сегодня безусловным лидером как в области теоретических исследований, так и в части практических приложений являются нейросетевые технологии. Моделирование мозга на самом нижнем уровне абстрагирования — на уровне нейронной структуры приводит к наиболее плодотворным технологиям создания интеллектуальных информационных систем.

1.3. Искусственный интеллект как лидирующая научная отрасль

В разные периоды существования человеческой цивилизации лидирующими направлениями научных исследований были философия, геометрия, астрология, алхимия, химия, такие разделы физики как механика, электричество и магнетизм, физика элементарных частиц и квантовая механика, ядерная физика. Однако начиная со второй половины XX в. и по настоящее время лидирующие позиции по темпам развития прочно занимают компьютерные науки. Схема, отражающая современное состояние распределения научных приоритетов, приведена на рис. 1.5. Из нее нетрудно видеть, что:

- из всего множества научных отраслей наиболее быстро развивающейся является *информатика*;
- из всех разделов информатики наиболее быстро развивается (в области как теоретических разработок, так и практических приложений) *искусственный интеллект*;
- из всех разделов искусственного интеллекта до середины 1980-х гг. лидирующие позиции в области практических приложений удерживали *экспертные системы*, однако в течение двух последних десятилетий они постепенно уступают лидерство другому разделу искусственного интеллекта — *нейронным сетям*.

Искусственный интеллект давно уже стал «горячей точкой» научных исследований. Здесь, как в фокусе, сконцентрированы усилия математиков, физиков, нейробиологов, психологов, программистов, философов и инженеров. Здесь решаются глобальные вопросы, связанные с дальнейшими путями развития научной мысли, с воздействием компьютерной индустрии на жизнь будущих поколений. Здесь рождаются и получают право на жизнь «пограничные» идеи как результат объединения различных дисциплин, научных направлений и областей. Здесь возникает

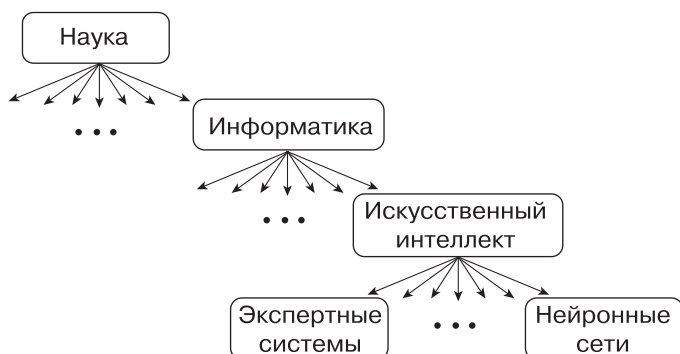


Рис. 1.5. Современное распределение приоритетных направлений развития научных отраслей

и формируется то, что называется философским осмыслением научных результатов.

Быстрому развитию искусственного интеллекта как научной отрасли в значительной мере способствует и доступность компьютерной техники. Благодаря ей у современных исследователей существует возможность быстрого компьютерного воплощения новых идей — их немедленной реализации и всесторонней практической проверки. Такая возможность является сильнейшим стимулом, способствующим лидерству искусственного интеллекта в темпах его развития.

1.4. Основные сферы применения искусственного интеллекта

Сегодня искусственный интеллект — это обширная область исследований и разработки интеллектуальных систем, предназначенных для самых разнообразных сфер человеческой деятельности, ранее недоступных для применения компьютеров из-за трудностей с их формализацией. В частности, для задач, решаемых методами искусственного интеллекта, характерно наличие большого числа степеней свободы с приближающимся к бесконечности количеством вариантов поиска решения. В отличие от жестко детерминированных компьютерных программ, системы

искусственного интеллекта сами ищут пути решения поставленной задачи. При этом они могут менять свои параметры и структуру, реагировать на изменения внешней среды, самообучаться, самосовершенствоваться и развиваться, — практически жить самостоятельной, не зависящей от разработчика жизнью.

Распознавание образов. К распознаванию образов в искусственном интеллекте относят достаточно широкий круг задач по распознаванию изображений, символов, текстов, запахов, звуков, шумов. Среди современных программных средств уже имеются системы оптического распознавания символов (такие как FineReader или CuneiForm), основанные на распознавании букв по некоторым их признакам, оснащенные базами данных и знаний, имеющие возможность адаптации и обучения. Однако в последнее время становятся популярными гибридные системы, в которых наряду с технологией экспертных систем используются и нейросетевые технологии.

Игры и творчество. Традиционно сфера применения технологий искусственного интеллекта включает в себя интеллектуальные задачи, решаемые при игре в шахматы, шашки, го, калах. Здесь очень широко применяется такой метод, как лабиринтная модель, плюс эвристики. Кроме того, в современных программах-игроках наиболее полно удалось реализовать основополагающие идеи искусственного интеллекта — обучение, самообучение и самоорганизацию.

В широком смысле слова под «игрой» в особом разделе математики, называемом *теорией игр*, понимается некая конфликтная ситуация, участники которой своими действиями не только достигают своих личных целей, но и могут влиять на достижимость целей другими участниками игры. Ясно, что под такое толкование термина «игра» попадают также и многие экономические, политические и военные конфликты.

Компьютерное творчество представляет пока чисто теоретический интерес. Наибольший прогресс в этой области достигнут в сочинении компьютерной музыки. Разработа-

ны также различные модели художественного и поэтического творчества, пока имеющие больше познавательный, чем практический интерес.

Компьютерная лингвистика. Начиная с 1950-х гг. и по настоящее время одной из популярных тем исследований в сфере искусственного интеллекта является машинный перевод с одного языка на другой. В 1954 г. в США была создана компьютерная программа, которая успешно перевела с одного языка на другой 60 фраз. В СССР работы в области машинного перевода начались в 1955 г. с создания программы-переводчика с английского языка на русский.

В первых программах-переводчиках применялась идея пословного перевода, что, естественно, не позволяло получать высокое качество результатов из-за большого количества синонимов и многозначных слов в естественных языках. Однако постепенно структуры естественно-языковых интерфейсов усложнялись, и в настоящее время в программах-переводчиках в обязательном порядке применяются:

- *морфологический анализ* — анализ слов в тексте;
- *синтаксический анализ* — анализ предложений, грамматики и связей между словами;
- *семантический анализ* — анализ смысла каждого предложения на основе базы знаний, на которую ориентирована конкретная программа-переводчик;
- *прагматический анализ* — анализ смысла предложений в окружающем контексте с помощью базы знаний.

Добавление баз знаний и использующих их анализаторов в программах-переводчиках позволило существенно улучшить качество перевода. Особенно ощутимых результатов удалось добиться при переводе узкоспециализированных текстов, например таких, как прогноз погоды. Тем не менее полноценный компьютерный перевод текста, как правило, пока что весьма далек от совершенства (особенно для художественных текстов).

Другой проблемой компьютерной лингвистики является разработка естественно-языкового интерфейса между человеком и машиной. Здесь важную роль могут сыграть

нейросетевые технологии, с помощью которых удастся научить компьютер правильному произношению слов и распознаванию устной речи. В проектах создания компьютеров V-го и VI-го поколений решению этой проблемы уделено первостепенное внимание.

Интеллектуальные роботы. Роботы — это технические устройства, предназначенные для автоматизации человеческого труда. Само слово «робот» появилось в 1920-х гг.; его автор — чешский писатель Карел Чапек, а само это слово происходит от чешского «робота» — тяжелый труд.

В настоящее время в промышленности применяется огромное количество роботов-манипуляторов, работающих по жесткой схеме управления. В отличие от них интеллектуальные роботы обладают способностью самообучаться и самоорганизовываться, адаптироваться к изменяющейся окружающей обстановке.

Если при создании первых интеллектуальных роботов использовалась технология экспертных систем, согласно которой в систему управления роботами закладывались базы правил, предписывающих их поведение, то позже здесь нашли широкое применение нейрокомпьютерные технологии. Благодаря этому роботы приобрели способность адаптироваться к окружающей среде, гибко реагировать на ее изменения, самонастраиваться и самосовершенствоваться.

Интересные эксперименты проводятся и по созданию эволюционных моделей управления роботами. Снабжение их природными механизмами самосовершенствования живых организмов, таких как селекция, скрещивание, мутация, естественный отбор, представляется весьма перспективным.

Искусственная жизнь, генетические алгоритмы и мультиагентные системы. Эти три направления берут свое начало с опытов профессора Мичиганского университета Дж. Холланда, который, пытаясь подтвердить Дарвиновскую теорию эволюции жизни на Земле, населил компьютер виртуальными особями — специально написанными программами, заставив их, подобно реальным земным су-

ществам, бороться за выживание, добывать себе пищу, размножаться и умирать. Научными работами такого рода под общим названием «Искусственная жизнь» и сейчас занимаются многие биологи, экологи и другие ученые, работающие в области наук о Земле и человеке. Однако широкое практическое приложение идей Дж. Холланда началось, когда было обнаружено, что дарвиновский алгоритм самосовершенствования всего живого может с успехом применяться для решения оптимизационных задач. Выяснилось, что природные оптимизационные алгоритмы, названные *генетическими*, легко справляются с поисками глобальных экстремумов многоэкстремальных функций, с которыми традиционные градиентные методы, как правило, справиться не могут. В настоящее время генетические алгоритмы активно применяются при решении многих оптимизационных задач, встречающихся преимущественно в экономико-математическом моделировании, а также при обучении нейронных сетей.

Другое практическое применение алгоритмов искусственной жизни связано с использованием *агентных (мультиагентных) систем*. Программами-особями, называемыми *агентами*, населяется сеть Интернет. Действуя самостоятельно, такие агенты выполняют полезную работу в интересах своего резидента-пользователя. Они следят за событиями в мире, добывая полезную информацию, играют на бирже ценных бумаг, добываясь максимальных дивидендов для хозяина, сообщают ему об угрозах и предотвращают их, иногда вступая в борьбу с чужими агентами.

Компьютерные вирусы. Сегодня трудно назвать компьютерного пользователя, избежавшего знакомства с этим видом «программной продукции». Следует признать, что последние поколения вирусов обладают уже всеми свойствами систем искусственного интеллекта. Они свободно перемещаются по компьютерам, мутируют и размножаются, самообучаются, самостоятельно изменяют свои параметры и структуру. Вирусы живут самостоятельной, независимой от их создателей жизнью.

Вредоносное воздействие компьютерных вирусов значительно возросло с появлением сети Интернет. Однако, по прогнозам специалистов, неприятности, с которыми мы встречаемся сегодня, представляются ничтожными по сравнению с теми, которые ожидают нас с проникновением компьютерных вирусов в сферу деятельности интеллектуальных роботов.

ДатаМайнинг (от англ. *Data mining*) — **извлечение знаний из данных**. Эта задача стала особенно актуальной в связи с обилием статистической информации, которой буквально «забиты» сегодня сеть Интернет и множество компьютеров, обслуживающих работу различных фирм и организаций. Однако за множеством этих безликих цифр скрываются важные закономерности, которые представляют собой те самые знания, которые необходимы менеджерам для выработки правильных решений. Увидеть эти закономерности человек не в состоянии. Незаменимым инструментом для создания систем извлечения знаний из данных и поддержки принятия решений на их основе являются методы искусственного интеллекта, в частности, нейросетевые технологии.



Коротко о главном

Основными современными стратегиями создания систем искусственного интеллекта являются технологии экспертных систем, нейросетевые технологии и технологии эволюционного моделирования. Согласно первой стратегии, знания о предметной области формализуются и в явном виде помещаются в экспертную систему, которая на основе этих знаний делает логически обоснованные выводы. Согласно второй стратегии, знания (так же как и в мозге) хранятся в неявной форме — кодируются в виде множества сил синаптических связей. Эти знания приобретаются в ходе обучения на примерах, характеризующих моделируемую предметную область. Наконец, в эволюционных моделях знания приобретаются в ходе конкурентной борьбы между особями.



Вопросы и задания

1. Перечислите основные стратегии создания систем искусственного интеллекта. Назовите примерные даты появления и основателей каждого из этих стратегических направлений.
2. Как приобретаются и в каком виде хранятся знания в системах искусственного интеллекта, созданных согласно трем основным стратегиям?
3. Какая научная область, по вашему мнению, является лидирующей в настоящее время?
4. Проведите в сети Интернет поиск по ключевым словам «искусственный интеллект», «экспертные системы», «нейронные сети», «нейроинформатика», «нейрокомпьютер», «агентные системы», «интеллектуальные роботы» и другим терминам искусственного интеллекта. Выберите заинтересовавшую вас информацию и напишите на ее основе реферат или подготовьте доклад на одну из предложенных учителем тем, например:
 - успехи искусственного интеллекта;
 - история искусственного интеллекта;
 - современные проблемы искусственного интеллекта;
 - современные технологии искусственного интеллекта;
 - будущее искусственного интеллекта и т. д.

ГЛАВА 2

ЭКСПЕРТНЫЕ СИСТЕМЫ

2.1. Данные и знания

Как правило, любая, даже самая простая, компьютерная программа оперирует не только с *данными*, но и со *знаниями*. Например, фрагмент программы, предназначенной для вычисления площади круга радиусом 20 см, на языке Pascal выглядит следующим образом:

```
R := 20;  
Pi := 3.14;  
S := Pi * R * R;  
writeln ('Площадь круга S=', S, ' кв.см');
```

Первые два оператора представляют собой данные, а вот третий оператор — это уже *знание*. Это — всем известная формула для вычисления площади круга через его радиус и число π , которая является результатом интеллектуальной деятельности древних геометров.

Прежде чем сформулировать определение знаний, вспомним, что собой представляют данные.

Данные — это отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства.

При обработке данные последовательно трансформируются, их вид может меняться:

- данные, существующие как результат измерений и наблюдений;
- данные на материальных носителях информации (в таблицах, протоколах, справочниках);

- данные, представленные в виде диаграмм, графиков или функций;
- данные в компьютере на языке описания данных;
- базы данных.

Знания основываются на данных, но представляют собой результат мыслительной деятельности человека, обобщающей его опыт, полученный в ходе практической деятельности или научных исследований. Знания могут выражать законы природы и общества, закономерности конкретных предметных областей.

При обработке на ЭВМ знания трансформируются аналогично данным:

- знания, существующие в памяти человека как результат обучения, воспитания, мышления;
- знания, помещенные на материальных носителях (в учебниках, инструкциях, методических пособиях, книгах);
- знания, описанные на языках представления знаний и помещенные в компьютер;
- базы знаний.

Итак, для хранения данных в компьютере используют базы данных, а для хранения знаний используются базы знаний. Для баз данных характерны большой объем и относительно небольшая стоимость информации. Базы знаний же, наоборот, отличаются сравнительно небольшими объемами, но значительно более высокой ценой.

В приведенном ранее фрагменте программы вычисления площади круга знания «растворены» в самом тексте программы. Такой вид представления знаний называют *процедурным*. Корректировка таких знаний требует изменения самого текста программы, поэтому с развитием искусственного интеллекта все большую часть знаний стали сосредоточивать в отдельных структурах. Такие знания называются *декларативными*.

Существуют десятки способов представления декларативных знаний. Мы рассмотрим только три наиболее употребительных способа:

- продукционные правила;
- фреймы;
- семантические сети.

2.2. Способы представления знаний

2.2.1. Продукционные правила

Продукционная система состоит из трех основных компонентов, схематично изображенных на рис. 2.1. Первый из них — это *база знаний*, состоящих из правил типа:

ЕСЛИ «условие», ТО «действие»

Примеры:

ЕСЛИ «холодно», ТО «надеть шубу»

ЕСЛИ «идет дождь», ТО «взять зонтик»

Вторым компонентом является *рабочая память*, в которой хранятся исходные данные к задаче и выводы, полученные в ходе работы системы.

Третий компонент — *механизм логического вывода*, использующий правила в соответствии с содержимым рабочей памяти.

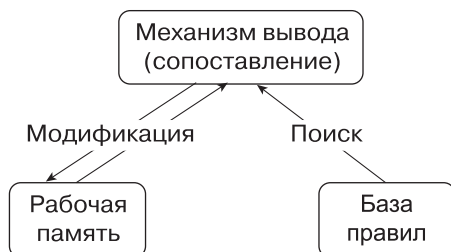


Рис. 2.1. Блок-схема продукционной системы

Рассмотрим конкретный пример. Пусть в базе правил экспертной системы имеются два правила.

Правило 1: ЕСЛИ «намерение — отдых» и «дорога ухабистая», ТО «использовать джип».

Правило 2: ЕСЛИ «место отдыха — горы», ТО «дорога ухабистая».

Предположим, что в рабочую память поступили следующие исходные данные:

«намерения — отдых»; «место отдыха — горы»

Механизм вывода начинает сопоставлять образцы из условных частей правил с образцами, хранимыми в рабочей памяти. Если образцы из условной части имеются в рабочей памяти, то условная часть считается истинной; в противном случае — ложной.

В данном примере при рассмотрении правила 1 оказывается, что образец «намерение — отдых» имеется в рабочей памяти, а образец «дорога ухабистая» отсутствует, поэтому условная часть правила 1 считается ложной. При рассмотрении же правила 2 выясняется, что его условная часть истинна. Тогда механизм вывода выполняет заключительную часть этого правила, и образец «дорога ухабистая» заносится в рабочую память, а само правило 2 при этом выбывает из числа кандидатов на рассмотрение.

Затем снова рассматривается правило 1, условная часть которого теперь становится истинной, и содержимое рабочей памяти пополняется образцом «использовать джип». В итоге получается, что правил, которые можно было бы еще применять, больше не остается, и система останавливает свою работу.

В рассмотренном примере показан *прямой вывод* — от данных к поиску цели. Однако в других задачах может применяться и *обратный вывод* — от цели (для ее подтверждения) к данным. Продемонстрируем этот способ на нашем примере. Пусть наряду с исходными данными «намерения — отдых»; «место отдыха — горы» у нас также имеется цель — «использовать джип».

Согласно правилу 1, для достижения этой цели требуется выполнение условия «дорога ухабистая», поэтому такое условие становится новой целью. При рассмотрении же правила 2 оказывается, что условная часть этого правила в данный момент истинна, поэтому рабочая память пополняется образцом «дорога ухабистая». А при повторном рассмотрении правила 1 подтверждается цель: «использовать джип».

При обратном выводе система останавливается в двух случаях: когда достигается первоначальная цель либо когда заканчиваются правила. При прямом же выводе система останавливается только тогда, когда заканчиваются правила, либо при появлении в рабочей памяти специально предусмотренного образца (например, «использовать джип»).

Для обратного вывода характерна тенденция исключения из рассмотрения правил, не имеющих прямого отношения к заданной цели, что позволяет повысить эффективность вывода. Обратный вывод применяют в тех случаях, когда цели известны и их сравнительно немного.

Продукционные правила — это наиболее часто используемый способ представления знаний в современных экспертных системах. Основными преимуществами таких экспертных систем являются высокая модульность, легкость внесения изменений и дополнений, простота механизма логического вывода.



Коротко о главном

Любая компьютерная программа, как правило, содержит не только данные, но и знания. Знания можно хранить в виде продукционных правил.



Вопросы и задания

1. Сформулируйте определения данных и знаний.
2. Как данные и знания трансформируются в процессе их обработки?
3. Какие знания называются процедурными, а какие — декларативными?
4. В чем заключается способ представления знаний с помощью продукционных правил?

2.2.2. Фреймы

В психологии и философии существует понятие абстрактного образа. Например, слово «автомобиль» вызывает у слушателей образ устройства, способного перемещаться, имеющего четыре колеса, салон для шофера и пассажиров, двигатель и руль. Считается, что современный человек широко использует абстрактные образы для хранения в своей памяти информации об окружающем мире.

Фрейм — это модель абстрактного образа, которую программисты используют для хранения знаний о рассматриваемой предметной области. Фрейм состоит из имени и отдельных единиц, называемых *слотами*, и обычно имеет следующую структуру:

Имя фрейма

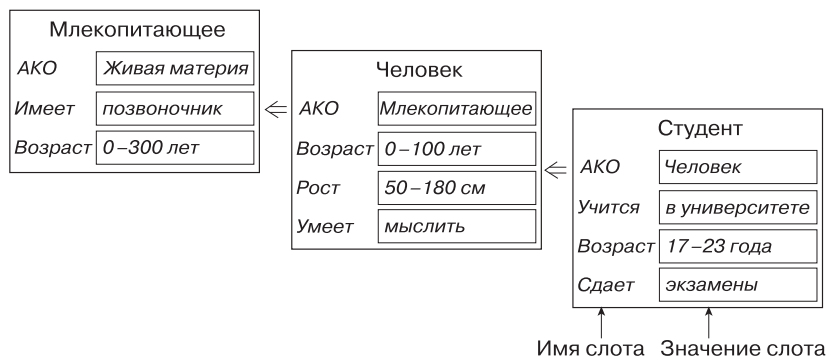
Имя 1-го слота: значение 1-го слота

Имя 2-го слота: значение 2-го слота

• • • • •

Имя N-го слота: значение N-го слота

В качестве значения слота может выступать в том числе имя другого фрейма, тем самым отдельные фреймы объединяются в сеть. Свойства фреймов наследуются сверху вниз (от вышестоящих к нижестоящим) через *АКО-связи* (от начальных букв английских слов *a kind of*, что можно перевести как «это»). Слот с именем АКО при этом указыва-

**Рис. 2.2.** Пример сети фреймов

ет на имя фрейма более высокого уровня иерархии. Например, на рис. 2.2 фрейм «Студент» имеет ссылки на вышестоящие фреймы «Человек» и «Млекопитающее», поэтому на вопрос: «Может ли студент мыслить?» ответ будет положительным, так как этим свойством обладает вышестоящий фрейм «Человек». Но если одно и то же свойство указывается в нескольких связанных между собой фреймах, то приоритет отдается нижестоящему фрейму. Так, значение «возраст» для фрейма «Студент» не наследуется из вышестоящих фреймов, а извлекается непосредственно из данного фрейма.

Основными преимуществами фреймов как способа представления знаний являются их наглядность и гибкость в употреблении. Кроме того, по мнению многих психологов, фреймовая структура согласуется с современными представлениями о хранении информации в памяти человека.

2.2.3. Семантические сети

В основе этого способа представления знаний лежит идея о том, что любые знания можно представить в виде совокупности *понятий* (объектов) и *отношений* (связей). На рис. 2.3 приведен пример графического изображения сети, вершины которой представляют собой понятия предметной об-

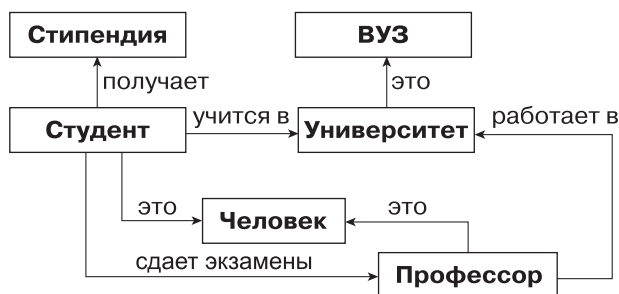


Рис. 2.3. Пример семантической сети

ласти, а связывающие их линии — отношения между этими понятиями. (Сам термин «семантическая» означает «смысловая».)

Основным преимуществом этой модели является ее наглядность, а ее главные недостатки — сложность поиска вывода и сложность корректировки, т. е. удаления и дополнения такой сети новыми знаниями.



Коротко о главном

Существуют три возможных способа представления знаний: продукционные правила, фреймы и семантические сети. В современных экспертных системах наибольшее распространение нашли продукционные правила.



Вопросы и задания

1. Перечислите основные преимущества и недостатки известных вам способов представления знаний.
2. Какой способ представления знаний наиболее распространен в современных экспертных системах?
3. Какой из рассмотренных способов представления знаний наиболее близок к способу, который использует мозг человека?

2.3. Технология экспертных систем

2.3.1. Назначение

Экспертная система — это программа, предназначенная для моделирования деятельности эксперта (специалиста) в какой-либо предметной области.

Знания, которыми обладают эксперты, можно разделить на формализуемые, плохо формализуемые и неформализуемые.

Формализуемые знания излагаются в книгах и руководствах в виде законов, формул, моделей, алгоритмов и характерны для точных наук, таких как математика, физика, химия, астрономия.

Науки, которые принято называть описательными, обычно оперируют с *плохо формализуемыми знаниями*. К таким наукам можно отнести, например, зоологию, ботанику, экологию, социологию, педагогику, медицину и др.

Существуют также знания, которые не попадают в книги из-за их неконкретности, субъективности, приблизительности. Знания такого рода являются результатом многолетних наблюдений, опыта работы, интуиции, они обычно представляют собой некие эмпирические и эвристические приемы и правила. Обычно такие знания передаются из поколения в поколение в виде определенных навыков, ноу-хау или секретов ремесла.

Возможны также знания, которые вообще не поддаются формализации и не могут быть выражены ни в математическом виде, ни в терминах обычного человеческого языка. Такими знаниями обладают, например, религиозные деятели, экстрасенсы, контактёры, шаманы.

Класс задач, относящихся к плохо формализуемым знаниям, значительно шире класса задач, для которых знания легко поддаются формализации. Экспертные системы предназначены для работы именно с плохо формализуемыми знаниями. Этим объясняется особая популярность и быстрое распространение экспертных систем в 1970-х гг., когда благодаря им появилась возможность проникнове-

ния компьютерных технологий в предметные области, характеризующихся плохо формализуемыми знаниями.

2.3.2. Блок-схема экспертной системы

Типичная блок-схема экспертной системы представлена на рис. 2.4. Обычно в ее состав входят следующие взаимосвязанные функциональные блоки:

- *база знаний* — ядро экспертной системы, которое представляет собой совокупность знаний из некоторой предметной области, записанных с помощью какого-либо способа представления знаний (например, с помощью продукционных правил, фреймов или семантических сетей — см. раздел 2.2);
- *интерфейс разработчика* — программа, предоставляющая инженеру-когнитологу и программисту возможность создавать базу знаний в диалоговом режиме, а также включает в себя системы вложенных меню, шаблонов языка представления знаний, подсказок («help-режим») и других сервисных средств, облегчающих работу с базой знаний;
- *интерфейс пользователя* — комплекс программ, реализующих диалог пользователя с экспертной системой на стадиях как ввода информации, так и получения результатов;

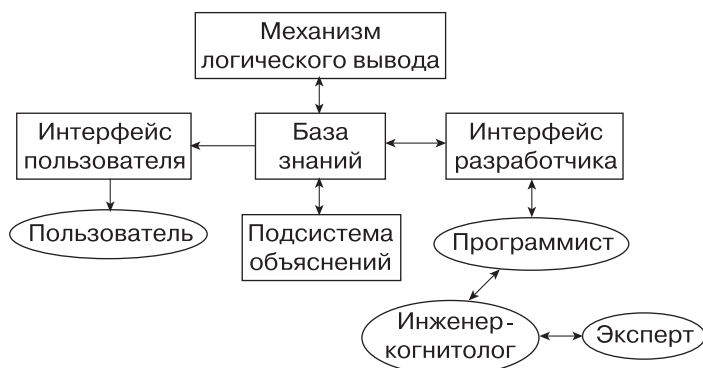


Рис. 2.4. Типичная блок-схема экспертной системы

- *механизм логического вывода (дедуктивная машина, решатель, блок логического вывода)* — программа, осуществляющая логический вывод путем моделирования хода рассуждений эксперта на основании знаний, имеющихся в базе знаний;
- *подсистема объяснений* — программа, позволяющая пользователю получить ответ на вопрос: «На основании чего сделано то или иное заключение?» Принцип работы этой программы заключается в том, что она последовательно выводит на экран компьютера все правила, которые были задействованы при получении заключения, чтобы при прочтении этих правил пользователю стала понятной логика проделанного экспертной системой вывода.

В коллектив разработчиков экспертной системы входят, как минимум, четыре специалиста (или четыре группы специалистов): *эксперт, инженер-когнитолог, программист и пользователь*. Возглавляет коллектив инженер-когнитолог — ключевая фигура при разработке экспертных систем. Обычно это руководитель проекта, в задачу которого входит организация всего процесса создания экспертной системы. С одной стороны, он должен быть специалистом в области искусственного интеллекта, а с другой — должен разбираться в данной предметной области, общаться с экспертом, извлекая и формализуя его знания, и передавать эти знания программисту, кодирующему и помещающему их в базу знаний экспертной системы.

Экспертная система может работать в двух режимах — приобретения знаний и решения задач (проведения консультаций). В режиме приобретения знаний производится формирование базы знаний. В режиме же решения задач общение с экспертной системой осуществляет уже конечный пользователь.

Обычно знания, которыми располагает эксперт, различаются степенью достоверности, надежности, важности и четкости. В этом случае они снабжаются некими весовыми

коэффициентами, которые называют *коэффициентами доверия (уверенности)*. Такие коэффициенты доверия могут указываться в баллах или в процентах. Например, если у больного высокая температура и он кашляет, то с уверенностью 90% можно утверждать, что он простужен, и с уверенностью 30% подозревать, что у него воспаление легких. Коэффициенты доверия вместе с правилами обрабатываются решателем экспертной системы; в результате вывод, который делает экспертная система, тоже снабжается некоторым коэффициентом доверия, который показывает, насколько такому выводу можно доверять. (Процесс обработки коэффициентов доверия решателем осуществляется с помощью *алгоритмов нечеткой логики*, которых мы в данном учебном курсе касаться не будем.)

В процессе опытной эксплуатации коэффициенты доверия могут подвергаться корректировке. В этом случае говорят, что происходит *обучение* экспертной системы. Процесс обучения экспертной системы может производиться автоматически с помощью обучающего алгоритма либо путем вмешательства инженера-когнитолога, выполняющего роль «учителя».

2.3.3. Этапы разработки экспертной системы

Технология разработки экспертных систем несколько отличается от технологии разработки обычных компьютерных программ. Это отличие обусловлено различиями в их характеристиках (табл. 2.1).

При разработке экспертных систем широко используются *прототипы* — компьютерные модели будущего программного продукта. Идея «прототипизации» состоит в том, что на ранней стадии работы над проектом создается упрощенная версия конечной программы, которая является доказательством продуктивности основных идей, заложенных в проекте. Такой прототип должен быть способен решать какую-либо из нетривиальных задач, характерных для заданной области применения. На основе анализа опыта работы с прототипом разработчики могут уточнить требования к системе в целом и к ее основным функциональ-

Таблица 2.1

**Отличия экспертных систем
от обычных компьютерных программ**

Характеристика	Экспертные системы	Традиционные компьютерные программы
Тип обработки	Символьная	Числовая
Метод	Эвристический поиск	Алгоритм
Задание шагов решения	Неявное	Точное
Искомое решение	Удовлетворительное	Оптимальное
Управление и данные	Перемешаны	Разделены
Знания	Неточные	Точные
Модификации	Частые	Редкие

ным характеристикам, а заказчики экспертной системы получают возможность оценить целесообразность выполнения проекта еще до того, как на него будут потрачены значительные средства.

Принято различать следующие прототипы экспертных систем:

- *демонстрационный прототип* — экспертная система, которая решает часть требуемых задач, демонстрируя жизнеспособность используемого метода инженерии знаний. База знаний при этом состоит всего из 50–100 правил. Время разработки такой экспертной системы — от 6 до 12 месяцев;
- *исследовательский прототип* — экспертная система, которая решает все требуемые задачи, но может быть неустойчива в работе и не полностью проверена. База знаний содержит 200–500 правил. Разработка занимает от 3 до 6 месяцев;

- *действующий прототип* — надежно решает все задачи, но для решения сложных задач может потребовать много времени и памяти. База знаний содержит 500–1000 правил. Время разработки — от 6 до 12 месяцев;
- *промышленная экспертная система* — обеспечивает высокое качество решения всех задач при минимуме времени и памяти (что достигается переписыванием программ с использованием более совершенных инструментальных средств и языков низкого уровня). База знаний содержит 1000–1500 правил. Время разработки — от 1 до 1,5 лет;
- *коммерческая экспертная система* — отличается от промышленной тем, что кроме «внутреннего» использования она может продаваться различным потребителям. База знаний содержит 1500–3000 правил. Время разработки — от 1,5 до 3 лет. Стоимость — от 0,3 до 5 млн долларов США.



Коротко о главном

Экспертная система — это сложный программный комплекс, предназначенный для моделирования деятельности специалиста (эксперта) в какой-либо предметной области. Ее главное отличие от обычной программы состоит в том, что в нее заложены знания эксперта, с помощью которых она может делать логически обоснованные выводы.



Вопросы и задания

1. Приведите примеры научных областей, в которых знания хорошо формализованы и в которых они плохо поддаются формализации.
2. Сформулируйте определение экспертной системы.
3. Перечислите функциональные блоки, из которых состоит типичная экспертная система, и укажите их назначение.

4. Что такое коэффициенты доверия, для чего они вводятся?
5. Что понимается под обучением экспертной системы?
6. Укажите несколько различий между экспертной системой и традиционной компьютерной программой.
7. Назовите примерную стоимость современной коммерческой экспертной системы.
8. Из каких специалистов, по вашему мнению, должен состоять коллектив разработчиков экспертной системы?
9. Кто был создателем первой экспертной системы? Для чего она была предназначена?

2.3.4. Программный инструментарий при разработке экспертных систем

Инструментальные программы, применяемые при разработке экспертных систем, можно разделить на четыре достаточно больших категории.

Оболочки экспертных систем. Системы этого типа создаются, как правило, на основе какой-нибудь экспертной системы, достаточно хорошо зарекомендовавшей себя на практике. При создании такой оболочки из экспертной системы-прототипа удаляются компоненты, слишком специфичные для области ее непосредственного применения, и оставляются те из них, которые не имеют узкой специализации. Примером может служить система EMYCIN, созданная на основе широко известной экспертной системы медицинской диагностики MYCIN.

Достоинством программ-оболочек является то, что они позволяют специалистам, не знакомым с программированием, воспользоваться результатами работы программистов, уже решавших аналогичные проблемы. Недостатком же таких оболочек является их ориентированность на достаточно узкий класс задач, хотя он может быть и более

широким, чем исходная экспертная система, на основе которой была создана та или иная оболочка.

Традиционные языки программирования. Экспертные системы могут создаваться с помощью традиционных языков программирования, таких как C, C++, Java, Basic, Fortran, Pascal и др. Эти языки дают проектировщику экспертных систем значительно бóльшую свободу действий, чем оболочки. Гибкость, предоставляемая традиционными языками программирования, особенно важна, когда невозможно заранее определить оптимальные способы представления знаний и управления логическим выводом. Преимуществом использования традиционных языков программирования является возможность достижения высокой эффективности программного продукта, связанная с их близостью к машинной архитектуре. Еще одно их достоинство — возможность включения разрабатываемых интеллектуальных подсистем в крупные программные комплексы общего назначения. Недостатком же использования традиционных языков программирования являются значительные трудозатраты программистов, обусловленные слабой приспособленностью этих языков для работы с символьными и логическими данными.

Языки искусственного интеллекта. К языкам искусственного интеллекта в первую очередь принято относить LISP и PROLOG, которые отличаются богатыми возможностями обработки символьных и логических данных. При этом язык LISP скрыл от программистов устройство памяти ЭВМ, а PROLOG позволил им не заботиться о потоке управления в программе. Однако такое повышение языкового уровня привело к снижению гибкости и универсальности по сравнению с традиционными языками программирования.

Язык LISP был разработан в Стэнфордском университете в начале 1960-х гг., а PROLOG — в Марсельском университете в 1971 г. Наибольшую популярность язык PROLOG приобрел в 1980-х гг., когда благодаря усилиям математиков был обоснован логический базис этого языка, а также когда в японском проекте вычислительных систем

V-го поколения язык PROLOG был принят в качестве базового для машины вывода.

Следует, однако, отметить, что в последнее время сфера применения языков LISP и PROLOG заметно сократилась. Это объясняется тем, что системы искусственного интеллекта часто служат модулями других больших программных приложений, когда стандарты разработки приводят к необходимости использования единого языка программирования для всего такого приложения.

Интегрированные среды поддержки разработки интеллектуальных систем. Это своего рода надстройки над языками искусственного интеллекта, такие как ART, KEE, G2. Фактически они являются большим набором хорошо интегрированных парадигм искусственного интеллекта, что позволяет пользователю комбинировать в процессе разработки экспертной системы разные стили программирования.

2.3.5. Инженерия знаний

При разработке экспертных систем одним из важнейших этапов является проектирование баз знаний, т. е. получение знаний специалистов предметных областей. Этот раздел искусственного интеллекта получил название *инженерии знаний*.

В настоящее время сложилось несколько стратегий получения знаний.

Наиболее ранним по времени появления является способ получения знаний в результате живого контакта инженера по знаниям с экспертом или с другим источником знаний (например, со специальной литературой). Многие авторы называют эту стратегию *извлечением знаний*.

Желание автоматизировать трудоемкий процесс получения знаний привело к появлению двух других стратегий, основанных на применении ЭВМ: приобретения знаний и формирования знаний.

Приобретение знаний — это способ построения базы знаний посредством диалога эксперта с компьютерной программой. Обычно такие программы ориентированы на кон-

кретные предметные области, разработаны с учетом их структуры и конкретных особенностей, т. е. не являются универсальными. Например, одна из первых систем подобного рода — TEIRESISA — была предназначена для пополнения базы знаний системы медицинской диагностики MYCIN.

Стратегия, называемая *формированием знаний*, включает в себя применение алгоритмов автоматического порождения гипотез методом обучения на примерах принятия решений в предметной области.

Обе эти стратегии автоматизированного получения знаний (приобретение и формирование знаний) являются весьма перспективными и быстро развивающимися разделами инженерии знаний, но до сих пор еще не достигли промышленного уровня и представляют больше научный, чем практический интерес. Наибольшее же распространение в настоящее время по-прежнему имеет стратегия «ручного» проектирования баз знаний, т. е. извлечение знаний путем живого общения инженера по знаниям с экспертом или другим источником знаний. Это весьма трудоемкая процедура, и, как отмечают многие специалисты, она является самым узким местом в общем процессе создания экспертной системы.

Существует множество рекомендаций по организации работы с экспертом и по выбору его поведения. Знакомство с предметной областью рекомендуется начинать с изучения соответствующей литературы, учебников, монографий, статей, отчетов, сведений из сети Интернет. Затем целесообразно применение *коммуникативных методов*, подразумевающих вступление в контакт с экспертами. Коммуникативные методы при этом разделяют на пассивные и активные. *Пассивные методы* подразумевают наблюдения за объектом (технологическим процессом и др.), составляющим предметную область, за работой эксперта и процессом принятия решений. К пассивным методам также относятся прослушивание лекций, читаемых экспертом, изучение его инструкций и протоколов «мыслей вслух» — попыток

объяснить принимаемые им решения. *Активные же методы* отличаются более широким разнообразием. Это анкетирование, интервью, свободный диалог, экспертные игры, дискуссии за круглым столом с участием нескольких экспертов, «мозговой штурм».

В результате инженер по знаниям формирует *концептуальную структуру* предметной области — модель предметной области, включающую описание ее объектов и связей между ними. Такую концептуальную структуру изображают в виде графов, фреймов или описывают словесно. Затем формируется *функциональная структура* предметной области — модель рассуждений эксперта и процесса принятия решений, которая представляется в виде таблицы, графа или предложений на естественном языке. В ней также могут присутствовать математические формулы, отражающие внутренние закономерности предметной области, и продукционные правила, имеющие эвристическую природу, а потому снабженные коэффициентами уверенности.

Процесс проектирования базы знаний заканчивается ее формализацией и программной реализацией.



Вопросы и задания

1. Приведите примеры известных вам экспертных систем.
2. Что такое оболочка экспертной системы?
3. Назовите и охарактеризуйте основные стратегии получения знаний.
4. Почему задачу извлечения знаний называют узким местом в проектировании экспертных систем? Какие идеи вы могли бы предложить для устранения этой ситуации?

ГЛАВА 3

НЕЙРОННЫЕ СЕТИ

3.1. Персептрон и его развитие

3.1.1. Мозг и компьютер

В самом начале компьютерной эры, в середине XX в., были предложены различные варианты принципов действия и архитектурного исполнения электронно-вычислительных машин. Многие из этих вариантов впоследствии были отвергнуты и забыты. Наиболее удачной оказалась архитектура машины фон Неймана, которую и имеет большинство современных компьютеров. Однако наряду с машиной фон Неймана до наших дней дошла еще одна схема, которая в последние годы стремительно развивается и находит все более широкое применение. Это нейрокомпьютеры и нейронные сети.

Нейронные сети и нейрокомпьютеры — одно из направлений компьютерной индустрии, в основе которого лежит идея создания искусственных интеллектуальных устройств по образу и подобию человеческого мозга. Дело в том, что традиционные компьютеры, выполненные по схеме фон Неймана, по своей структуре и свойствам весьма далеки от нашего естественного компьютера — человеческого мозга. В подтверждение этому в табл. 3.1 приведены основные признаки, отличающие человеческий мозг от фон-неймановского компьютера.

Основатели же *нейрокибернетики* задались целью создания электронных устройств, адекватных мозгу не только на функциональном, но и на структурном уровне. Для этого им пришлось обратиться за сведениями к биологам.

Таблица 3.1

Сопоставление принципов построения и свойств современного компьютера (машины фон Неймана) и человеческого мозга

Признаки	Компьютер	Человеческий мозг
Процессор	Сложный	Простой
	Высокоскоростной	Низкоскоростной
	Один или несколько	Большое количество
Память	Отделена от процессора	Интегрирована в процессор
	Локализованная	Распределенная
Вычисления	Централизованные	Распределенные
	Последовательные	Параллельные
	Жесткие алгоритмы	Самообучение
Надежность	Высокая уязвимость	Живучесть

Как же устроен человеческий мозг? Известно, что мозг человека состоит из белого и серого вещества: белое вещество — это тела нервных клеток, называемых *нейронами*, а серое вещество — это соединяющие их нервные волокна. Каждый нейрон состоит из трех частей: *тела клетки*, *дендритов* и *аксона* (рис. 3.1). Дендриты и аксон — это нервные отростки, через которые нейрон обменивается электрическими сигналами с другими нейронами. Каждый нейрон может иметь до 10 000 дендритов и всего лишь один аксон. Через дендриты нейрон принимает электрические сигналы, поступающие от других нейронов по *нервным волокнам*, как по проводам. Если таких сигналов много и они достаточно интенсивны, то нейрон переходит в возбужденное состояние и сам вырабатывает электрический сигнал, который передает в аксон. Аксон же на своем другом конце разветвляется на тысячи нервных волокон, которые соединяются с дендритами других нейронов. Места соединения нервных волокон с дендритами называются *синапсами*.

Как же человеческий мозг запоминает информацию и ее обрабатывает? Увы, найти исчерпывающий ответ на этот вопрос биологи не смогли до сих пор. Но зато они знали, что общее число нейронов в течение жизни человека практически не изменяется. Это значит, что мозг ребенка и мозг взрослого человека содержат приблизительно одинаковое количество нейронов. Различия же между ними состоят в величинах электропроводностей синапсов.

Как известно из электротехники, электропроводность проводника ρ — это величина, обратная его электросопротивлению R и имеющая размерность $1/\Omega\text{м}$. Биологи называли электропроводность синапсов *силами межнейронных синаптических связей*; по мнению ученых, мозг одного человека отличается от мозга другого человека прежде всего именно силами межнейронных синаптических связей. На этом основании была высказана гипотеза, что все наши мысли, эмоции, знания, вся информация, хранящаяся в человеческом мозге, закодирована в виде огромного количества цифр, характеризующих силы межнейронных синаптических связей.

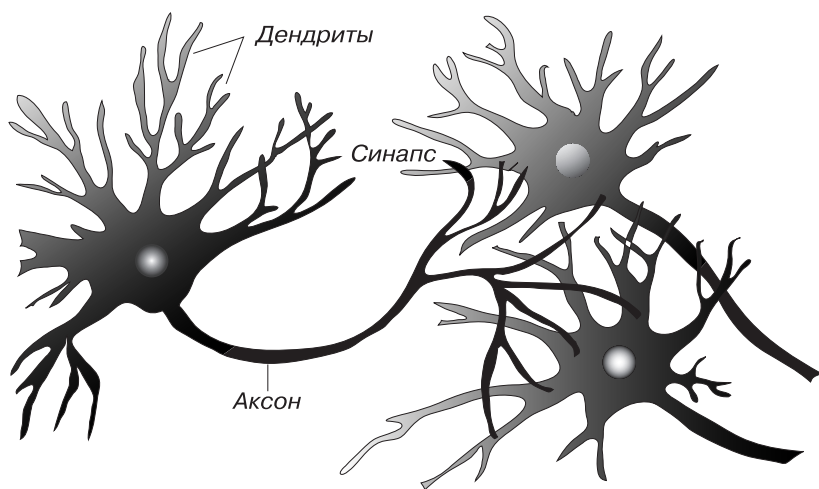


Рис. 3.1. Нейроны человеческого мозга

А теперь давайте попробуем оценить, сколько чисел способен запомнить такой гипотетический мозг, если принять, что с помощью одной синаптической связи можно закодировать одно число.

В человеческом мозге содержится приблизительно 10^{11} нейронов, и каждый нейрон связан с 10^3 – 10^4 другими нейронами. Тогда биологическая нейронная сеть, составляющая мозг человека, содержит 10^{14} – 10^{15} синапсов. Получается, что именно такое количество цифр способен хранить человеческий мозг и что именно таким количеством цифр закодированы в нем все знания, весь жизненный опыт, все мысли и эмоции, вся информация, которую человек получает на протяжении своей жизни.

Приведенные здесь представления о строении и функционировании мозга в настоящее время считаются научно обоснованным фактом. Ни у кого из ученых не вызывает сомнений, что разум человека создается огромным количеством мельчайших нервных клеток — нейронами, непрерывно исполняющими свой «информационный танец». А вот что это за «танец», мы рассмотрим в следующих разделах.

3.1.2. Биологический и математический нейрон

Первой работой, заложившей теоретический фундамент для создания интеллектуальных устройств, моделирующих человеческий мозг на самом низшем, структурном уровне, принято считать опубликованную в 1943 г. статью Уоррена Мак-Каллока и Вальтера Питтса «Идеи логических вычислений в нервной деятельности». Ее авторы, американские математики-нейрофизиологи, по праву считающиеся основателями нейроинформатики, предложили математическую модель нейрона мозга человека, назвав ее *математическим*, или *модельным нейроном*.

Математический нейрон Мак-Каллока — Питтса изображен на рис. 3.2 в виде кружочка. Он имеет несколько входов и один выход, показанные на рисунке стрелками. Через входы, количество которых мы обозначим как J , математический нейрон принимает *входные сигналы* x_j , ко-

которые он суммирует, умножая каждый входной сигнал на некоторый *весовой коэффициент* w_j :

$$S = w_1x_1 + w_2x_2 + \dots + w_Jx_J = \sum_{j=1}^J w_jx_j. \quad (3.1)$$

После выполнения такой операции суммирования математический нейрон формирует выходной сигнал y согласно следующему правилу:

$$y = \begin{cases} 1, & \text{если } S \geq \theta, \\ 0, & \text{если } S < \theta, \end{cases} \quad (3.2)$$

где θ — *порог чувствительности нейрона*.

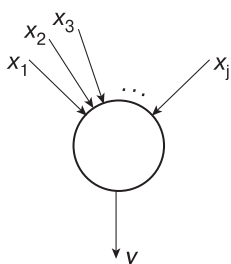


Рис. 3.2.
Математический
нейрон
Мак-Каллока —
Питтса

Таким образом, математический нейрон, как и его биологический прототип, может существовать в двух состояниях — возбужденном и невозбужденном. Как видно из формулы (3.2), если взвешенная сумма входных сигналов S меньше некоторой пороговой величины θ , то математический нейрон не возбужден, а его выходной сигнал равен нулю. Если же входные сигналы достаточно интенсивны и их сумма достигает порога чувствительности, то нейрон переходит в возбужденное состояние, а на его выходе формируется сигнал $y = 1$.

Весовые коэффициенты w_j имеют вполне определенный физический смысл. Они имитируют электропроводности нервных волокон — те самые, которые биологи называют силами межнейронных синаптических связей, или *синаптическими весами*. Чем эти силы выше, тем больше вероятность перехода нейрона в возбужденное состояние.

Ранее мы уже отмечали, что биологи придают силам синаптических связей очень важное значение. Считается, что именно с их помощью человеческий мозг кодирует всю име-

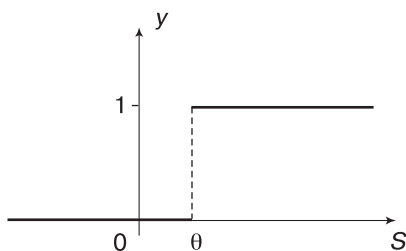


Рис. 3.3. Пороговая активационная функция нейрона, заданная формулой (3.2)

ющуюся у него информацию. Как мы увидим далее, в искусственном мозге — в нейронных сетях и в нейрокомпьютерах — происходит то же самое. Вся информация в них тоже кодируется в виде множества цифр, характеризующих силы межнейронных синаптических связей w_j .

Логическая функция (3.2) называется *активационной функцией нейрона*. Ее графическое изображение, представленное на рис. 3.3, по форме напоминает ступеньку, поэтому ее часто называют *функцией-ступенькой*.

Таким образом, математический нейрон представляет собой пороговый элемент с несколькими входами и одним выходом, причем каждый математический нейрон имеет свое определенное значение порога θ .

С помощью математического нейрона можно моделировать различные логические функции, например, функцию логического умножения «И» (ее также обозначают как «AND»), функцию логического сложения «ИЛИ» («OR») и функцию логического отрицания «НЕ» («NOT»). Таблицы истинности этих логических функций приведены в табл. 3.2, в которых истинное значение логических функций закодировано единицей, а ложное значение — нулем.

С помощью этих таблиц и формул (3.1), (3.2) нетрудно убедиться (рис. 3.4), что математический нейрон, имеющий два входа с единичными силами синаптических связей $w_1 = w_2 = 1$, моделирует функцию логического умножения «И» при $\theta = 2$. Такой же нейрон моделирует функцию

Таблица 3.2

Таблицы истинности логических функций

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

«И»

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

«ИЛИ»

x	y
0	1
1	0

«НЕ»

логического сложения «ИЛИ» при задании $\theta = 1$. Математический же нейрон с одним входом моделирует функцию «НЕ» при задании $w = -1$ и $\theta = 0$.

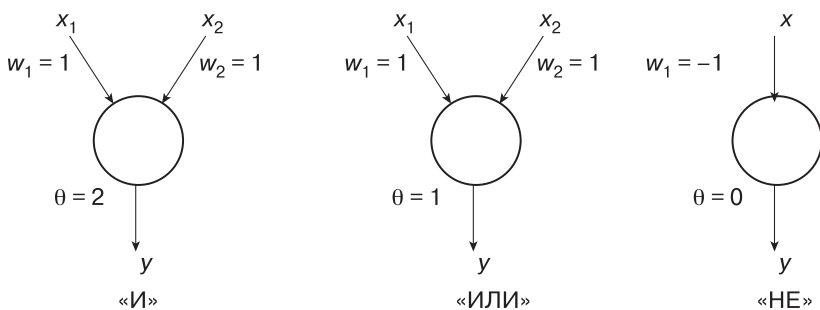


Рис. 3.4. Математические нейроны, моделирующие логические функции



Моделированию простейших логических функций с помощью математического нейрона посвящена **лабораторная работа № 1** из состава лабораторного практикума, размещенного на сайте <http://www.LbAI.ru>.



Коротко о главном

Согласно наиболее распространенным современным научным представлениям, вся информация и все знания в человеческом мозге кодируются и хранятся в виде множества цифр, характеризующих силы межнейронных синаптических связей. Математический нейрон Мак-Каллока — Питтса — это математическая модель биологического нейрона мозга, учитывающая его структуру и функциональные свойства.



Вопросы и задания

1. Назовите несколько различий по принципам действия между современным компьютером, выполненным по схеме фон Неймана, и человеческим мозгом.
2. Сколько нейронов содержит человеческий мозг?
3. Сколько дендритов и сколько аксонов может иметь один нейрон? Каково их назначение?
4. Сколько нервных волокон, соединяющих нейроны между собой, имеется в человеческом мозге?
5. В каком виде информация хранится в человеческом мозге?
6. Объясните на языке электротехники значение термина «сила синаптической связи». В каких единицах она измеряется?
7. Какой объем памяти имеет человеческий мозг? Сколько чисел он может запомнить?
8. Сколько входов и выходов имеет биологический нейрон?
9. Сколько входов и выходов может иметь математический нейрон Мак-Каллока — Питтса?
10. Напишите формулы, с помощью которых происходит преобразование сигналов в математическом нейроне Мак-Каллока — Питтса.

11. Нарисуйте графическое изображение активационной функции математического нейрона Мак-Каллока — Питтса.
12. Нарисуйте математические нейроны, реализующие логические функции «И», «ИЛИ», «НЕ», и приведите соответствующие им значения сил синаптических связей и порогов.
13. Чем весовые коэффициенты w_j отличаются от синаптических весов и от сил синаптических связей?

3.1.3. Персептрон и его обучение

Итак, как мы узнали из предыдущего параграфа, американские ученые У. Мак-Каллок и В. Питтс предложили математическую модель нейрона мозга человека, назвав ее *математическим нейроном*. Так же, как и биологический нейрон, математический нейрон имеет несколько входов и один выход и может существовать в возбужденном и невозбужденном состояниях, причем переход в возбужденное состояние зависит от величины поступающих к нему сигналов и сил синаптических связей. Таким образом, математический нейрон весьма правдоподобно имитирует структуру и свойства своего прототипа — биологического нейрона мозга. На этом основании У. Мак-Каллок и В. Питтс высказали весьма смелое и даже несколько фантастичное предположение, которое впоследствии легло в основу современной нейроинформатики. Они предположили, что если математические нейроны связать между собой проводниками, имитирующими нервные волокна (рис. 3.5), то такой искусственный мозг будет способен решать интеллектуальные задачи — подобно тому, как это делает естественный человеческий мозг.

Идея Мак-Каллока — Питтса была воплощена в жизнь в 1958 г. американским ученым Фрэнком Розенблаттом, которого также считают основателем нейроинформатики. Сначала он создал компьютерную программу для электронно-вычислительной машины IBM-794, имитирующую (*эмули-*

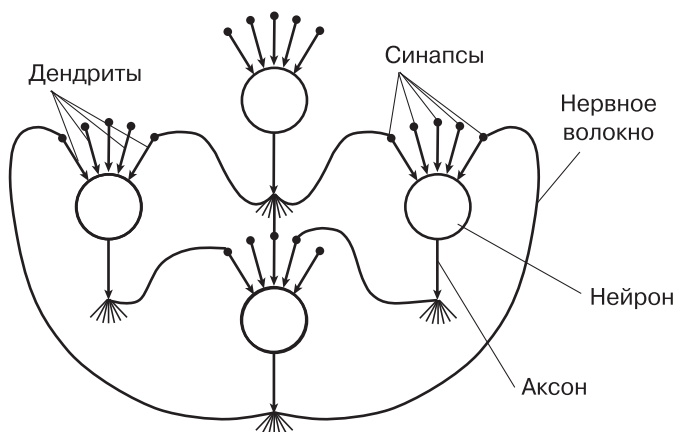


Рис. 3.5. Математические нейроны, связанные между собой в нейронную сеть

рующую) деятельность математических нейронов. Это и была первая *нейронная сеть* или, сокращенно, *нейросеть*. Она была названа *персептроном* от английского слова *perception* — «осознание».

Затем, спустя два года, Ф. Розенблатт смонтировал электронное устройство, в котором функции математических нейронов выполняли отдельные электросхемы, работающие на электронных лампах. Это был первый *нейрокомпьютер*, который успешно решал сложнейшую интеллектуальную задачу: распознавал буквы латинского алфавита, изображенные на карточках, подносимых к его считывающему устройству — «электронному глазу».

Таким образом, смелая гипотеза Мак-Каллока — Питтса была блестяще подтверждена. Но раз эксперимент удался, значит оказались правильными наши представления о биологической структуре и строении мозга, о его внутренних электрофизиологических процессах, о способе запоминания и хранения информации! Была подтверждена и адекватность математического нейрона как модели биологического нейрона, а также адекватность нейросети и нейрокомпьютера как моделей мозга. «Нельзя сказать, что мы точно воспроизводим работу человеческого мозга, —

писал Ф. Розенблатт, — но пока персептрон ближе всего к истине».

Рассмотрим принцип действия персептрона на примере решения конкретных задач. На рис. 3.6 приведен один из простейших вариантов персептрона, предназначенного для классификации чисел на четные и нечетные. Представим себе матрицу из 12 фотоэлементов, расположенных в виде четырех горизонтальных рядов по три фотоэлемента в каждом ряду. На матрицу фотоэлементов накладывается карточка с изображением цифры (например, «4», как изображено на рис. 3.6). Если на какой-либо фотоэлемент попадает фрагмент цифры, то этот фотоэлемент вырабатывает сигнал в виде единицы, в противном случае — сигнал «ноль». На рис. 3.6 на первый фотоэлемент не попал фрагмент цифры, поэтому его сигнал $x_1 = 0$; на второй фотоэлемент попал фрагмент цифры, поэтому он вырабатывает сигнал $x_2 = 1$, и т. д.

Согласно формулам (3.1), (3.2), математический нейрон выполняет суммирование входных сигналов x_j , помноженных на синаптические веса w_j , после чего результат суммирования S сравнивается с порогом чувствительности θ и вырабатывается выходной сигнал y .

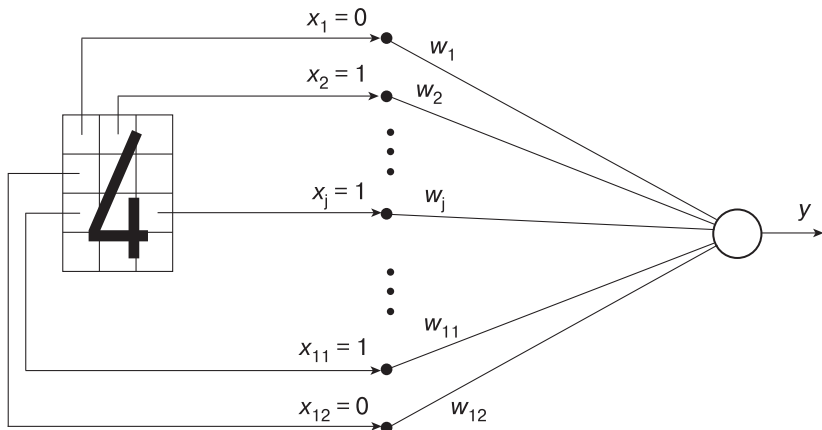


Рис. 3.6. Персептрон, классифицирующий числа на четные и нечетные

Первоначальные значения синаптических весов w_j и порога чувствительности θ Ф. Розенблатт задавал при помощи датчика случайных чисел, поэтому на выходе персептрона случайным образом вырабатывался сигнал: либо 0, либо 1.

Далее задача состояла в следующем. Требовалось подобрать значения синаптических весов w_j такими, чтобы выходной сигнал y принимал значение «единица», если на карточке было изображено четное число, либо «нуль», если число было нечетным.

Эту задачу Ф. Розенблатт решил путем поочередного накладывания на фотоэлементы различных карточек и *обучения* персептрона, заключающегося в корректировке синаптических весов w_j . Если, например, на вход персептрона предъявлялась карточка с цифрой «4» и выходной сигнал y случайно оказывался равным единице (что означает четность), то корректировать синаптические веса не требовалось, так как реакция персептрона правильна. А вот если выходной сигнал в этом случае оказывался равным нулю (что неправильно), то следовало увеличить («поощрить») веса тех активных входов, которые должны были способствовать возбуждению нейрона; в данном случае увеличению подлежали значения w_2, w_{11} и др.

Следуя этой идее, можно сформулировать *итерационный алгоритм* корректировки синаптических весов, обеспечивающий обучение персептрона в нужном направлении.

Шаг 1. С помощью датчика случайных чисел всем синаптическим весам w_j ($j = 1, \dots, 12$) и порогу чувствительности нейрона θ присвоить некоторые малые случайные значения.

Шаг 2. Предъявить персептрону какую-либо цифру — системой фотоэлементов вырабатываются входные сигналы x_j ($j = 1, \dots, 12$).

Шаг 3. Нейрон выполняет взвешенное суммирование входных сигналов:

$$S = \sum_{j=1}^{12} w_j x_j$$

и вырабатывает выходной сигнал $y = 1$, если $S \geq \theta$, или $y = 0$, если $S < \theta$.

Шаг 4а. Если выходной сигнал правильный, то перейти на шаг 2.

Шаг 4б. Если выходной сигнал неправильный и равен нулю, то увеличить веса активных входов: добавить к каждому j -му синаптическому весу величину j -го входного сигнала:

$$w_j(t+1) = w_j(t) + x_j.$$

Тогда, если вход был неактивен, т. е. $x_j = 0$, то j -й синаптический вес не изменится. Если же вход был активен, т. е. $x_j = 1$, то j -й синаптический вес увеличивается на 1.

(Здесь и далее t обозначает номер итерации, которые в искусственном интеллекте называют *эпохами*; $w_j(t+1)$ — новое значение на новой эпохе j -го синаптического веса; $w_j(t)$ — его старое значение на предыдущей эпохе.)

Шаг 4в. Если выходной сигнал неправильный и равен единице, то надо уменьшить веса активных входов, например с помощью аналогичной формулы:

$$w_j(t+1) = w_j(t) - x_j.$$

Шаг 5. Перейти на шаг 2 или завершить процесс обучения.

В приведенном здесь алгоритме шаг 4б называют *первым правилом Хебба*, а шаг 4в — *вторым правилом Хебба* в честь канадского ученого физиолога Д. О. Хебба, предложившего этот алгоритм в 1949 г.

Отметим, что правила Хебба удивительным образом напоминают процесс обучения ребенка или школьника методом «поощрения — наказания» (или дрессировки животного методом «кнута и пряника»). Обратим внимание также и на то, что первоначальные значения синаптических весов w_j задаются датчиком случайных чисел. Это соответствует тому, что при рождении человека или животного его мозг еще не накопил знаний, поэтому силы синаптических связей w_j имеют какие-то случайные значения. Как и в слу-

чаях с ребенком, школьником или животным, обучаемым методом «поощрения-наказания», алгоритм обучения персептрона за конечное число попыток (которые мы называли *эпохами*) может привести к цели: персептрон в конце концов усвоит необходимые знания, закодирует их в виде конкретных значений матрицы сил синаптических связей w_j и таким образом научится различать четные и нечетные числа.



Убедиться в способности персептрона распознавать четные и нечетные числа можно при выполнении **лабораторной работы № 2** из состава лабораторного практикума, размещенного на сайте <http://www.LbAI.ru>.

Естественно возникает вопрос, всегда ли алгоритм обучения персептрона приводит к желаемому результату? Ответ на этот вопрос дает *теорема сходимости персептрона*.

Если существует множество значений весов, которые обеспечивают требуемое распознавание образов, то в итоге алгоритм обучения персептрона приводит либо к этому множеству, либо к другому множеству, такому, что требуемое распознавание образов будет достигнуто.

В настоящее время считается, что по количеству предложенных доказательств теорема сходимости персептрона занимает первое место в мире (ранее самой «доказываемой» в мире теоремой считалась теорема Пифагора).



Коротко о главном

Нейрокомпьютер, построенный по образу и подобию человеческого мозга и обученный по алгоритму, похожему на алгоритм обучения человека, оказался способным решать сложнейшую интеллектуальную задачу — классифицировать (распознавать) числа на четные и нечетные.



Вопросы и задания

1. Чем нейронная сеть отличается от нейрокомпьютера?
2. Как вырабатываются входные сигналы $x_1, x_2, x_3, \dots$ персептрона, классифицирующего числа на четные и нечетные?
3. Как задаются первоначальные значения синаптических весов $w_1, w_2, w_3, \dots$ и как они затем корректируются?
4. Как персептрон хранит знания, необходимые для распознавания цифр?
5. Какая теорема считается самой доказываемой в мире?
6. Сформулируйте теорему сходимости персептрона.

3.1.4. Дельта-правило и распознавание букв

Рассмотренный ранее алгоритм обучения персептрона можно представить в более общей форме. Если через d обозначить требуемый выходной сигнал (от «*desire response*», что в переводе с английского означает «желаемый отклик»), то на каждой эпохе обучения можно определять разницу между желаемым ответом персептрона d и действительным значением y , вычисляемым на его выходе:

$$\varepsilon = d - y.$$

Тогда:

- случай $\varepsilon = 0$ соответствует шагу 4а;
- случай $\varepsilon > 0$ соответствует шагу 4б;
- случай $\varepsilon < 0$ соответствует шагу 4в.

Идея алгоритма обучения персептрона с помощью правил Хебба сохранится, если итерационный процесс корректировки весов вести по формулам:

$$w_j(t+1) = w_j(t) + \Delta w_j; \quad (3.3)$$

$$\Delta w_j = \varepsilon x_j, \quad (3.4)$$

где $w_j(t)$ и $w_j(t+1)$ — это, соответственно, старое и новое значения весовых коэффициентов персептрона, а j — номер входного сигнала.

В итерационную формулу полезно также ввести *коэффициент скорости обучения* η , с помощью которого можно управлять величиной коррекции синаптических весов:

$$\Delta w_j = \eta \epsilon x_j. \quad (3.5)$$

При слишком больших значениях коэффициента η обычно теряется устойчивость процесса обучения, а при слишком малых его значениях процесс обучения может затянуться на неоправданно долгое время. На практике коэффициент скорости обучения η обычно задают в пределах от 0,05 до 1.

Алгоритм обучения персептрона с использованием этих формул известен под названием *дельта-правила*.

Дальнейшее развитие идеи персептрона и алгоритмов обучения связано с усложнением его структуры и функциональных свойств. На рис. 3.7 представлена схема персептрона, предназначенного для распознавания букв русского алфавита. В отличие от предыдущей схемы, такой персептрон имеет 33 выходных нейрона (каждой букве алфавита соответствует свой выходной нейрон). Предполагается, что сигнал первого выходного нейрона y_1 должен быть равен единице, если персептрону предъявлена буква «А», и равен нулю, если предъявлена любая другая буква. Аналогично выходной сигнал второго нейрона y_2 должен быть равен единице, если персептрону предъявлена буква «Б», и равен нулю во всех остальных случаях. И так до буквы «Я» включительно.

Обратим внимание, что весовые коэффициенты w_{ij} теперь имеют двойную индексацию, означающую, что сигнал через данную синаптическую связь идет от j -го входа сети к i -му выходному нейрону.

Алгоритм обучения такого персептрона выглядит следующим образом.

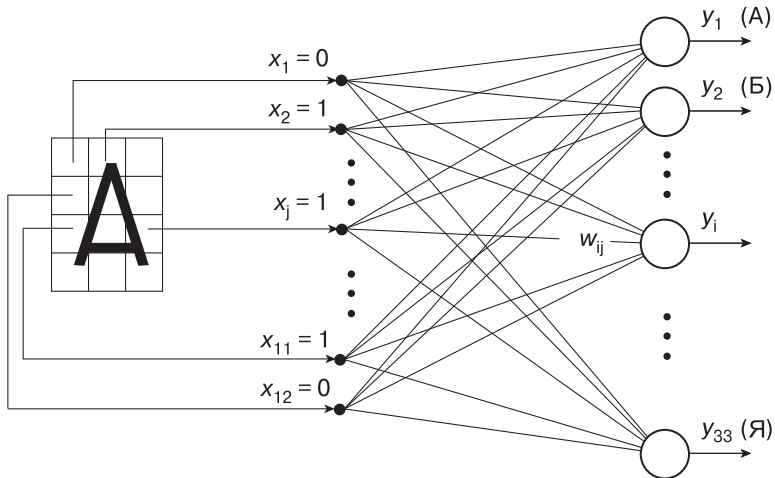


Рис. 3.7. Персептрон, предназначенный для распознавания букв русского алфавита

Шаг 1. При помощи датчика случайных чисел всем весовым коэффициентам w_{ij} ($i = 1, \dots, 33; j = 1, \dots, 12$) присваиваются некоторые малые случайные значения.

Шаг 2. Персептрону предъявляется какая-либо буква алфавита — системой фотоэлементов вырабатываются входные сигналы x_j ($j = 1, \dots, 12$).

Шаг 3. Каждый нейрон выполняет взвешенное суммирование входных сигналов:

$$S_i = \sum_{j=1}^{12} w_{ij} x_j$$

и вырабатывает выходной сигнал $y_j = 1$, если $S_i \geq 0$, или $y_i = 0$, если $S_i < 0$.

Шаг 4. Для каждого нейрона вычисляется его *ошибка*:

$$\varepsilon_i = d_i - y_i,$$

где d_i — правильные (желаемые) ответы персептрона; например, для буквы «А» $d_1 = 1, d_2 = 0, \dots, d_{33} = 0$.

Шаг 5. Производится корректировка весовых коэффициентов с помощью формул дельта-правила:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}, \quad (3.6)$$

$$\Delta w_{ij} = \eta \varepsilon_i x_j, \quad (3.7)$$

где t — номер эпохи.

Шаг 6. Повторение шагов 2–5 необходимое количество раз.

Как уже отмечалось ранее, первый действующий персептрон был создан в 1958–1961 гг. Он был предназначен для распознавания букв латинского алфавита. Буквы, отпечатанные на карточках, поочередно накладывали на табло из фотоэлементов и осуществляли процесс обучения персептрона согласно приведенному здесь алгоритму. После выполнения достаточно большого количества эпох персептрон научился безошибочно распознавать все буквы, участвовавшие в обучении. Таким образом, была подтверждена гипотеза о том, что компьютер, построенный по образцу и подобию человеческого мозга, может решать интеллектуальные задачи и, в частности, решать задачу распознавания образов.

Но это было еще не все. Кроме того, что персептрон научился распознавать знакомые образы (те, которые демонстрировались ему в процессе обучения), он успешно справлялся и с распознаванием образов, которые «видел» впервые! Так, выяснилось, что персептрон оказался способен распознавать буквы, отпечатанные с небольшими искажениями и даже другим шрифтом, если только шрифт не слишком сильно отличался от используемого при обучении персептрона.

Свойство мозга узнавать образы, которые ему встретились впервые, называется *свойством обобщения*. Нетрудно догадаться, что это свойство было «унаследовано» персептроном от его прототипа — живого мозга — благодаря тому, что персептрон является адекватной моделью мозга, удачно отражающей его как структурные, так и функциональные качества. Именно свойство обобщения впослед-

ствии дало возможность применять нейронные сети для решения широчайшего круга практических задач, не доступных для традиционных методов информатики. Именно благодаря этому свойству нейронные сети стали эффективным инструментом научных исследований и практических приложений, а нейросетевые и нейрокомпьютерные технологии заняли то лидирующее положение, которое они занимают в настоящее время (см. рис. 1.5).



В способности персептрона распознавать буквы алфавита можно убедиться при выполнении **лабораторных работ № 3 и 4** из состава лабораторного практикума, размещенного на сайте <http://www.LbAI.ru>.



Коротко о главном

Дельта-правило — это алгоритм обучения персептрона, являющийся обобщением правил Хебба.

Персептрон можно обучить распознавать не только цифры, но и буквы. Такой персептрон должен иметь столько выходных нейронов, сколько букв требуется распознавать.

Персептрон обнаружил новое качество, унаследованное им от своего прототипа — мозга. Он научился распознавать не только те образы (в данном случае буквы), на которых его обучали, но и образы, которые он «видел» впервые. Это свойство человеческого мозга называется свойством обобщения.



Вопросы и задания

1. Что подразумевается под введенными нами обозначениями d_i и y_i ? Чем они различаются?
2. Запишите формулу, с помощью которой вычисляется нейронная ошибка ε_i .
3. Запишите формулы, по которым, согласно алгоритму дельта-правила, корректируются синаптические веса и нейронные смещения.

4. Что такое коэффициент скорости обучения? Для чего он нужен и в каких пределах его обычно задают? К чему приводят слишком большие и слишком маленькие значения скорости обучения?
5. Чем отличается схема персептрона, предназначенного для классификации чисел на четные и нечетные, от схемы персептрона, распознающего буквы русского алфавита?
6. Какое количество выходных нейронов должен иметь персептрон, предназначенный для распознавания букв английского алфавита?
7. Какое количество выходных нейронов должен иметь персептрон, предназначенный для распознавания как русских букв, так и цифр от 0 до 9?
8. Что понимается под свойством обобщения, которым обладает мозг человека и его модель — персептрон?
9. Как научить персептрон распознавать не только печатные, но и рукописные буквы?

3.1.5. Дальнейшее развитие идеи персептрона

Персептрон, схема которого была приведена на рис. 3.7, предназначен для распознавания букв русского алфавита. Можно попытаться использовать его и для решения других практических задач, например, для постановки диагнозов болезней. Все зависит от того, какой смысл придавать входным сигналам x_j и выходным сигналам y_j . Однако для других задач, например, для прогноза погоды, температуры воздуха, котировок акций и курсов валют, такой персептрон не годится, так как он может выдавать только *бинарные* выходные сигналы «нуль» и «единица».

Круг решаемых задач значительно расширится, если научить персептрон выдавать не только бинарные, но и аналоговые выходные сигналы, имеющие непрерывные значения. Такое развитие персептрона было сделано американскими учеными Б. Уидроу и М. Е. Хоффом, которые

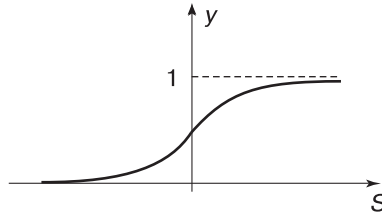


Рис. 3.8. Сигмоидная активационная функция $y = f_{\sigma}(S)$

вместо ступенчатой (см. рис. 3.3) ввели непрерывную функцию активации:

$$y = \frac{1}{1 + e^{-S}}, \quad (3.6)$$

график которой изображен на рис. 3.8.

Эту функцию называли *сигмоидой*, поскольку ее графическое изображение напоминает латинскую букву «S», и приняли для нее обозначение $y = f_{\sigma}(S)$. Другое ее название — *логистическая функция*.

Подобно обычной пороговой функции активации, сигмоида выполняет преобразование точек области определения $(-\infty, +\infty)$ в значения из интервала $(0, +1)$.

Появление персептронов с непрерывными активационными функциями обусловило появление новых подходов к их обучению. Авторы сигмоидной активационной функции Б. Уидроу и М. Е. Хофф вместо функции-ошибки предложили минимизировать *квадратичную ошибку*, определяемую формулой:

$$\varepsilon = \sum_{i=1}^I (d_i - y_i)^2, \quad (3.7)$$

в которой, как и раньше, d_i — это требуемый (желаемый) выход i -го нейрона, а y_i — выход, который получается в действительности в результате работы персептрона.

Рассмотрим алгоритм коррекции весовых коэффициентов персептрона, имеющего J входов и I выходов (рис. 3.9).

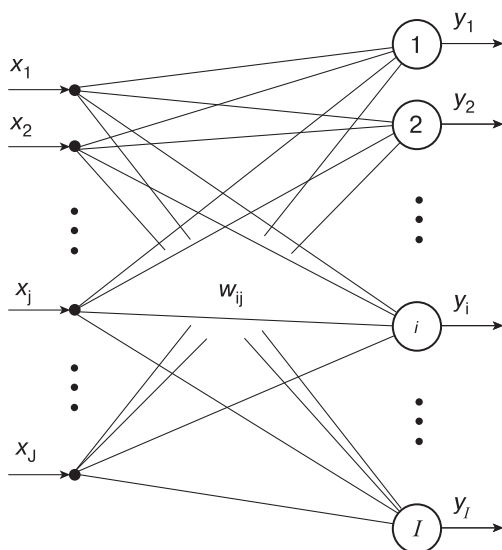


Рис. 3.9. Персептрон с J входами и I выходами

Квадратичная ошибка обучения персептрона ε зависит от того, какими являются весовые коэффициенты w_{ij} . Другими словами, ε является функцией от весовых коэффициентов: $\varepsilon = \varepsilon(w_{ij})$.

В школьном курсе математики и на других предметах обычно изучаются функции только от одного аргумента: $y = y(x)$, которые на координатной плоскости (x, y) изображаются в виде кривых линий. Если же функция z зависит от двух аргументов: $z = z(x, y)$, то она изображается в трехмерной системе координат (x, y, z) в виде *поверхности*. Функция-ошибка персептрона $\varepsilon = \varepsilon(w_{ij})$ и вовсе зависит от большого количества аргументов w_{ij} , поэтому для ее графического представления требуется *многомерная система координат*, которую мы в нашем трехмерном мире представить себе не можем. В этой многомерной системе координат функция $\varepsilon = \varepsilon(w_{ij})$ изображается в виде *многомерной поверхности*, которую называют *гиперповерхностью*.

Чтобы хоть как-то представить себе гиперповерхность, предположим, что все аргументы w_{ij} «заморожены»

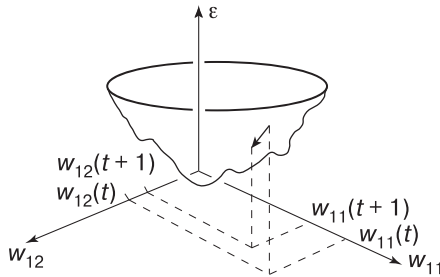


Рис. 3.10. Графическое изображение функции-ошибки персептрона $\varepsilon = \varepsilon(w_{ij})$ в трехмерной системе координат $(w_{11}, w_{12}, \varepsilon)$

(не меняются), за исключением двух из них, — например w_{11} и w_{12} , которые являются переменными. Тогда в трехмерной системе координат $(w_{11}, w_{12}, \varepsilon)$ гиперповерхность будет иметь вид фигуры, напоминающей параболоид, которую мы назовем *псевдопараболоидом* (рис. 3.10).

Процесс обучения персептрона теперь можно представить как отыскание такого сочетания весовых коэффициентов w_{ij} , которому соответствует самая нижняя точка *гиперпсевдопараболоида*. Такие задачи называются *оптимизационными*. При этом говорят, что оптимизационная задача состоит в *минимизации целевой функции* $\varepsilon = \varepsilon(w_{ij})$ в многомерном пространстве ее аргументов — весовых коэффициентов w_{ij} .

Таким образом, если раньше мы говорили, что персептрон обучают методом «поощрения — наказания», то теперь мы рассматриваем задачу обучения персептрона как задачу оптимизации (точнее, минимизации) персептронной ошибки (погрешности).

Существует множество методов решения оптимизационных задач. Наиболее простым методом является простой перебор весовых коэффициентов w_{ij} с последующим вычислением и сравнением между собой соответствующих этим коэффициентам значений функции ε . Однако «слепой» перебор координат требует очень много времени и поэтому практически не применяется. Более эффективны методы,

осуществляющие перебор координат в определенном направлении, например в сторону наибольшего убывания функции. Для выбора такого направления обычно вычисляют *градиент функции* ε .

Градиент функции является очень важным математическим понятием, которое проходят на первых курсах вузов. Здесь мы не будем заострять на нем внимание, а только укажем, что градиент функции представляет собой вектор, проекциями которого на оси координат являются производные от этой функции по ее координатам, и что градиент функции всегда направлен в сторону ее наибольшего возрастания. Но поскольку наша задача состоит в отыскании *минимума* функции $\varepsilon = \varepsilon(w_{ij})$, нам надо *опускаться* по этой поверхности (см. рис. 3.10), что обеспечивается движением в сторону, противоположную градиенту, т. е. в сторону *антиградиента*, показанного на рис. 3.10 в виде стрелки.

Методы оптимизации, использующие понятие градиента функции, называют *градиентными*. С одним из частных случаев применения градиентного оптимизационного метода мы уже встречались на уроках алгебры, когда искали экстремум параболы, приравнивая к нулю ее производную, которая, как следует из только что данного определения понятия градиента, является его частным случаем. В курсе алгебры мы познакомились с формулой для вычисления координаты экстремума параболы:

$$x_0 = -\frac{b}{2a},$$

где a и b — коэффициенты параболы, заданной уравнением $y = ax^2 + bx + c$.

В случае многомерной поверхности, которой изображается функция ошибок персептрона $\varepsilon = \varepsilon(w_{ij})$, мы не можем повторить тот же прием из-за сложности вида этой функции. Поэтому поиск ее экстремума осуществляется *итерационным путем*, т. е. в ходе последовательных приближений к точке экстремума. Сначала при помощи датчика случайных чисел задается начальная точка с координатами $w_{ij}(t)$. Затем в этой точке вычисляются производные по

координатам от функции $\varepsilon = \varepsilon(w_{ij})$ и определяется ее градиент, после чего делается небольшой шаг в сторону, противоположную градиенту. Процесс такого *градиентного спуска* повторяется заданное количество раз (эпох) или же до тех пор, пока значение ошибки обучения ε не достигнет заданного минимального значения. Величина шага в сторону антиградиента регулируется при этом все тем же коэффициентом скорости обучения η .

Опуская промежуточные выкладки, включающие взятие необходимых для определения градиента производных, приведем окончательные итерационные формулы:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}, \quad (3.8)$$

$$\Delta w_{ij} = \eta \delta_i x_j, \quad (3.9)$$

$$\delta_i = y_i(1 - y_i)(d_i - y_i). \quad (3.10)$$

Введенную здесь с помощью формулы (3.10) величину δ_i называют *нейронной ошибкой*.

Формулы (3.8)–(3.10) называют *обобщенным дельта-правилом*. Его преимущество по сравнению с обычным дельта-правилом (см. формулы (3.6), (3.7)) состоит в более быстрой сходимости и в возможности более точной обработки входных и выходных непрерывных сигналов, т. е. в расширении круга решаемых персептронами задач.



Коротко о главном

Введение сигмоидной функции активации вместо функции-ступеньки и появление нового алгоритма обучения — обобщенного дельта-правила — расширило область применения персептрона. Теперь он может оперировать не только бинарными («ноль» и «единица»), но и непрерывными (аналоговыми) выходными сигналами.



Вопросы и задания

1. Нарисуйте графическое изображение сигмоидной активационной функции и запишите ее математическую формулу.
2. Чем сигмоидная функция активации лучше (или хуже) функции-ступеньки?
3. Чем сигмоидная активационная функция отличается от логистической?
4. Запишите формулу для вычисления квадратичной ошибки персептрона. От каких величин она зависит?
5. В виде какой геометрической фигуры изображается квадратичная ошибка обучения персептрона?
6. Чем гиперпсевдопараболоид отличается от псевдопараболоида?
7. Что собой представляет градиент функции? В какую сторону он направлен?
8. В чем заключается идея метода градиентного спуска?
9. Можно ли применять алгоритм дельта-правила для обучения персептрона с сигмоидными активационными функциями?
10. Какие преимущества и недостатки имеет обобщенное дельта-правило перед обычным дельта-правилом?

3.1.6. Ограниченность однослойного персептрона

Как уже говорилось ранее, ученым удалось обучить персептрон распознавать буквы алфавита. Это был колоссальный успех: электронное устройство, созданное по образу и подобию человеческого мозга и обученное подобно человеку, успешно моделировало интеллектуальные функции человека.

Это был успех в познании самой природы человеческого мышления. Мозг начал раскрывать свои тайны. Появи-

лась возможность исследовать мозг методами моделирования, не прибегая к сложнейшим, антигуманным и мало что дающим натурным экспериментам. Это была сенсация, приковавшая к себе внимание мыслящих людей всего мира. Казалось, что ключ к интеллекту был найден, и полное воспроизведение человеческого мозга и всех его функций — это всего лишь вопрос времени. Писателям-фантастам, ученым, инженерам, бизнесменам, политикам виделись самые радужные перспективы практического применения идей искусственного интеллекта. Правительство Соединенных Штатов Америки выделило крупные субсидии на развитие нового перспективного научного направления. Благодаря изобретению сигмоидных активационных функций и алгоритма градиентного спуска класс решаемых задач расширился. Делались попытки применения персептронов для решения задач прогнозирования, таких как предсказание погоды, курсов валют и акций. Персептроны пытались применять для анализа электрокардиограмм, решения задач медицинской диагностики и т. д.

Однако по мере расширения фронта научных исследований появлялись все новые трудности. Неожиданно оказалось, что многие новые задачи персептрон решить не мог, хотя эти задачи внешне практически ничем не отличались от тех, с которыми персептрон успешно справлялся ранее. Возникла необходимость объяснения подобных парадоксов, необходимость глубокого анализа и создания теоретической базы нейроинформатики.

Следующий период истории искусственного интеллекта начался с появления в 1969 г. книги двух известных американских математиков М. Минского и С. Пайперта «Персептроны». Авторы этой книги математически строго доказали, что использовавшиеся в то время *однослойные персептроны* в принципе не способны решать многие простые задачи. Одну из таких задач, вошедшую в историю под названием «проблемы Исключающего ИЛИ», мы рассмотрим более подробно.

«Исключающее ИЛИ» — это логическая функция двух аргументов, каждый из которых может иметь значение «истинно» либо «ложно». Сама она принимает значение «истинно», когда только один из аргументов имеет значение «истинно», а во всех остальных случаях эта функция принимает значение «ложно». Если закодировать значение «истинно» единицей, а значение «ложно» — нулем, то требуемое соответствие между аргументами x_1 , x_2 и самой функцией y можно представить в виде табл. 3.3, называемой *таблицей истинности логической функции*.

Задача состоит в том, чтобы научиться моделировать функцию «Исключающее ИЛИ» с помощью однонейронного персептрона с двумя входами x_1 и x_2 и одним выходом y (рис. 3.11).

М. Минский и С. Пайперт в своей книге предложили геометрическую интерпретацию к «проблеме Исключающего ИЛИ», состоящую в следующем. Они предложили изобразить на координатной плоскости (x_1, x_2) все возможные комбинации входных сигналов в виде четырех точек: A, B, C, D , как показано на рис. 3.12. Точка A при этом имеет координаты $x_1 = 0, x_2 = 0$; точка B имеет координаты $x_1 = 0, x_2 = 1$ и т. д., согласно табл. 3.4.

Таблица 3.3

Таблица истинности логической функции
«Исключающее ИЛИ»

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

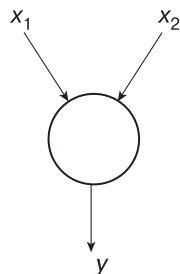


Рис. 3.11. Одно-нейронный персептрон с двумя входами и одним выходом

Таблица 3.4

Таблица истинности логической функции
«Исключающее ИЛИ», дополненная точками A, B, C, D

Точки	x_1	x_2	y
A	0	0	0
B	0	1	1
C	1	0	1
D	1	1	0

Тогда в точке A выход персептрона y должен быть равен нулю, в точке B — единице, в точке C — единице, а в точке D — снова нулю.

В соответствии с формулами (3.1), (3.2) однопейронный персептрон (см. рис. 3.11) осуществляет преобразование:

$$S = w_1 x_1 + w_2 x_2; \quad (3.11)$$

$$y = \begin{cases} 1, & \text{если } S \geq \theta, \\ 0, & \text{если } S < \theta. \end{cases} \quad (3.12)$$

Заменяем в уравнении (3.8) S на θ :

$$w_1 x_1 + w_2 x_2 = \theta. \quad (3.13)$$

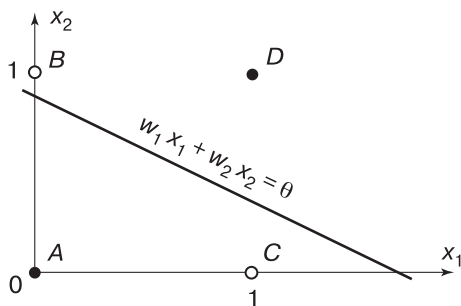


Рис. 3.12. Геометрическая интерпретация к объяснению «проблемы Исключающего ИЛИ»

Если в этом уравнении величины x_1 и x_2 считать переменными, а θ , w_1 и w_2 — константами, то на координатной плоскости (x_1, x_2) рассматриваемое уравнение можно изобразить в виде прямой, положение и наклон которой определяются значениями коэффициентов w_1 , w_2 и порога θ . Для всех точек плоскости x_1, x_2 , лежащих на этой линии, выполняется равенство $S = \theta$, поэтому, согласно формуле (3.12), выход персептрона равен единице. Для точек, лежащих выше указанной линии, сумма $w_1x_1 + w_2x_2$ больше θ , поэтому, согласно этой же формуле, выход персептрона также равен единице. А для точек, лежащих ниже этой линии, сумма $w_1x_1 + w_2x_2$ меньше θ , и выход персептрона равен нулю. Поэтому линию, изображающую уравнение (3.13), называют *пороговой прямой*.

А теперь посмотрим на таблицу истинности функции «Исключающее ИЛИ» (см. табл. 3.4). Согласно этой таблице, в точках A и D выход персептрона должен быть нулевым, а в точках B и C — единичным. Для этого надо расположить пороговую прямую так, чтобы точки A и D лежали ниже этой линии, а точки B и C — выше, что невозможно! Это значит, что, сколько бы такой персептрон мы ни обучали, какие бы значения ни придавали его синаптическим весам и порогу, этот персептрон в принципе не способен воспроизвести соотношение между входами и выходом, требуемое таблицей истинности функции «Исключающее ИЛИ».

Кроме «проблемы Исключающего ИЛИ», в упомянутой выше книге «Персептроны» М. Минский и С. Пайперт привели ряд других задач, в которых точки, изображающие входные сигналы, не могут быть разделены пороговой прямой (в многомерных случаях — плоскостью или гиперплоскостью). Такие задачи получили название *линейно неразделимых*.

После выхода в свет книги М. Минского и С. Пайперта всем стало ясно, что активно предпринимавшиеся в то время попытки обучать персептроны решению многих задач, — которые, как оказалось, относятся к классу линейно нераз-

делимых, — с самого начала были обречены на провал, и что решение нужно искать в изменении самой конструкции персептрона.



Коротко о главном

Однонейронный персептрон не позволяет моделировать логическую функцию «Исключающее ИЛИ» и решать другие линейно неразделимые задачи.



Вопросы и задания

1. Запишите таблицы истинности логических функций «И», «ИЛИ» и «Исключающее ИЛИ».
2. Повторите рис. 3.12 и начертите на нем пороговую прямую так, чтобы однонейронный персептрон, параметры которого соответствуют нарисованной вами пороговой прямой, моделировал:
 - логическую функцию «И»;
 - логическую функцию «ИЛИ».
3. Пользуясь вашим рисунком, объясните, почему однонейронный персептрон не может моделировать функцию «Исключающее ИЛИ».
4. Сформулируйте определение линейно неразделимых задач.
5. Сформулируйте определение пороговой прямой.
6. Подумайте над тем, как заставить персептрон решать линейно неразделимые задачи.

3.1.7. Решение «проблемы Исключающего ИЛИ»

Появление книги М. Минского и С. Пайперта «Персептроны» вызвало настоящий шок в научном мире. Строгие математические доказательства М. Минского и С. Пайперта были неуязвимы. Всеобщий энтузиазм сменился не менее всеобщим пессимизмом. В газетах стали появляться критические статьи с сообщениями, что ученые в своих иссле-

дованиях зашли в тупик, впустую израсходовав деньги налогоплательщиков. Правительство США немедленно прекратило финансирование нейропроектов и приступило к поискам виновных в растрате государственных денег. Бизнесмены, потерявшие надежду вернуть вложенные капиталы, отвернулись от ученых, и нейроинформатика была предана забвению, длившемуся более 20 лет.

Тем не менее работы в области нейросетевых и нейрокомпьютерных технологий продолжались отдельными энтузиастами. Такие работы продолжались и в засекреченных научно-исследовательских институтах Советского Союза, отделенного в то время от Запада пресловутым «железным занавесом». Не имея информации о мнениях и настроениях зарубежных коллег, советские ученые спокойно продолжали заниматься захватившей их умы темой и к началу 1980-х гг. удивили мир появлением ракет и самолетов, управлявшихся компьютерами нового поколения — *нейрокомпьютерами*. Советские компьютеры, в отличие от американских, стойко переносили довольно серьезные повреждения, продолжая работать в сложных условиях, что было особенно важно для объектов военного назначения. Тем самым выявилось и еще одно свойство нейрокомпьютеров, «унаследованное» ими от живого мозга, — *свойство живучести*.

Как же удалось достигнуть таких успехов? Советским ученым С. О. Мкртчаном была издана книга «Нейроны и нейронные сети. Введение в теорию формальных нейронов», в которой он показал, что с помощью *многослойных персептронов* может быть смоделирована любая логическая функция, если только известна ее логическая формула. Более того, им был разработан специальный математический аппарат, позволяющий конструировать такие персептроны! Оказалось, что «проблема Исключающего ИЛИ», явившаяся «камнем преткновения» для одностеронного персептрона, может быть решена с помощью нейронной сети, состоящей из трех нейронов, — *трехнейронного персептрона*, изображенного на рис. 3.13.

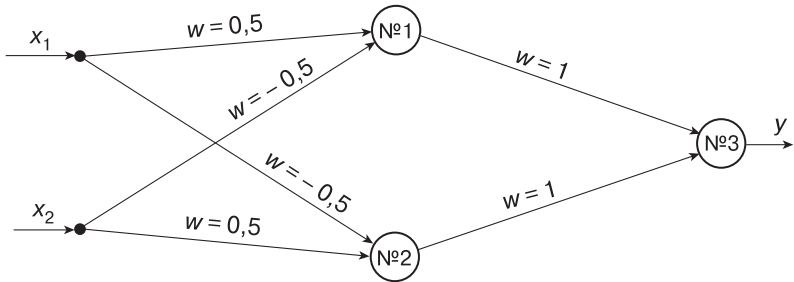


Рис. 3.13. Нейронная сеть, моделирующая функцию «Исключающее ИЛИ»

Работа такого персептрона происходит по следующему алгоритму.

Нейрон № 1:

$$S_1 = 0,5 \cdot x_1 + (-0,5) \cdot x_2;$$

$$y_1 = \begin{cases} 1, & \text{если } S_1 \geq \theta, \\ 0, & \text{если } S_1 < \theta. \end{cases}$$

Нейрон № 2:

$$S_2 = (-0,5) \cdot x_1 + 0,5 \cdot x_2;$$

$$y_2 = \begin{cases} 1, & \text{если } S_2 \geq \theta, \\ 0, & \text{если } S_2 < \theta. \end{cases}$$

Нейрон № 3:

$$S_3 = 1 \cdot y_1 + 1 \cdot y_2;$$

$$y_3 = \begin{cases} 1, & \text{если } S_3 \geq \theta, \\ 0, & \text{если } S_3 < \theta. \end{cases}$$

Задав значение порога $\theta = 0,5$ и заполнив с помощью этих формул табл. 3.5, легко убедиться, что трехнейронный персептрон успешно моделирует функцию «Исключающее ИЛИ».

Позже было показано, что и другие линейно неразделимые задачи, приведенные в книге М. Минского и С. Пайперта, могут быть решены с помощью нейросетей, содержа-

Таблица 3.5

Процесс формирования сигналов в трехнейронном персептроне

x_1	x_2	S_1	S_2	y_1	y_2	S_3	y_3	y
0	0	0	0	0	0	0	0	0
0	1	-0,5	0,5	0	1	1	1	1
1	0	0,5	-0,5	1	0	1	1	1
1	1	0	0	0	0	0	0	0

щих один или несколько *скрытых нейронных слоев* — дополнительных слоев нейронов, расположенных между входным и выходным слоями.

Многие исследователи понимали, что нужно создавать нейросети все более сложной архитектуры, содержащие скрытые слои нейронов, но не представляли, как обучать такие сети. Ведь правила Хебба, дельта-правило и алгоритм градиентного спуска годились только для корректировки синаптических весов нейронов выходного слоя, тогда как вопрос о настройке параметров скрытых нейронных слоев пока еще оставался открытым.



Коротко о главном

Логическую функцию «Исключающее ИЛИ» может моделировать нейронная сеть, состоящая из трех нейронов, которая изображена на рис. 3.13.



Вопросы и задания

1. Нарисуйте персептрон, моделирующий функцию «Исключающее ИЛИ».
2. С помощью формул, описывающих работу математического нейрона, убедитесь, что нарисованный вами персептрон действительно моделирует функцию «Исключающее ИЛИ».

3. Попробуйте изобразить другой персептрон (другой структуры), тоже способный моделировать логическую функцию «Исключающее ИЛИ».
4. Почему не удастся применять уже известные вам алгоритмы обучения (правила Хебба, дельта-правило, обобщенное дельта-правило) для обучения персептронов, моделирующих функцию «Исключающее ИЛИ»?
5. Попытайтесь придумать алгоритм обучения персептрона, содержащего один скрытый слой нейронов.

3.1.8. Алгоритм обратного распространения ошибки

Эффективный алгоритм обучения многослойных персептронов, открывший путь их широкого практического применения, стал известен только в 1986 г. благодаря публикациям Д. Румельхарта, Г. Хилтона и Р. Вильямса. Идея этого алгоритма заключается в том, что ошибки нейронов выходного слоя $\varepsilon_i = d_i - y_i$ используются для вычисления ошибок нейронов, расположенных в скрытых слоях. Значения ошибок при этом как бы распространяются от выходного слоя нейронов внутрь сети, от последующих нейронных слоев к предыдущим. Отсюда и название этого метода: *алгоритм обратного распространения ошибки (back propagation)*.

Интересно отметить, что алгоритм обратного распространения ошибки был предложен на один год раньше в работах еще двух ученых — А. Паркера и А. Ле-Кана, изданных независимо друг от друга. Более того, еще в 1974 г. этот простой и изящный алгоритм был защищен П. Вербосом в его докторской диссертации. Однако тогда он остался незамеченным и только спустя более десяти лет был «переоткрыт» заново, получил всеобщее признание и применение. Остались незамеченными и работы советских ученых, которые еще раньше разрабатывали подобные алгоритмы в своих засекреченных институтах и успешно применяли их при построении систем управления объектами военного назначения.

Рассмотрим идею алгоритма обратного распространения ошибки, попытавшись распространить обобщенное дельта-правило на случай обучения *двухслойного персептрона*, имеющего N входов, I выходов и скрытый слой из J нейронов (рис. 3.14). Нужно отметить, впрочем, что этот персептрон на самом деле имеет *три* слоя, однако в литературе его называют двухслойным, поскольку нейроны входного слоя имеют всего один вход, не имеют синаптических весов и не выполняют суммирования входных сигналов, а лишь передают один-единственный входной сигнал нейронам следующего слоя.

Алгоритм корректировки синаптических весов нейронов выходного слоя мы оставим таким же, как и для однослойного персептрона (см. обобщенное дельта-правило — формулы (3.8)–(3.10)), заменив в них x_j на y_j ¹:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}, \quad (3.14)$$

$$\Delta w_{ij} = \eta \delta_i y_j, \quad (3.15)$$

$$\delta_i = y_i(1 - y_i)(d_i - y_i). \quad (3.16)$$

Синаптические веса нейронов скрытого слоя мы попытаемся корректировать с помощью все тех же формул (3.8)–(3.10), в которых индекс i заменим на j , а индекс j — на индекс n :

$$w_{jn}(t+1) = w_{jn}(t) + \Delta w_{jn}; \quad (3.17)$$

$$\Delta w_{jn} = \eta \delta_j x_n; \quad (3.18)$$

$$\delta_j = y_j(1 - y_j)(d_j - y_j). \quad (3.19)$$

При использовании этих формул возникает вопрос о вычислении нейронной ошибки $(d_j - y_j)$, которая для скрытого слоя неизвестна. Идея авторов алгоритма обрат-

¹ Причины этой замены становятся понятными, если проследить пути прохождения сигналов, сравнивая рис. 3.9 с рис. 3.14: если в однослойном персептроне (рис. 3.9) на вход нейронов выходного слоя подавались входные сигналы x_j , то в двухслойном персептроне (рис. 3.14) на вход нейронов выходного слоя подаются выходные сигналы нейронов с предыдущего (скрытого) слоя y_j .

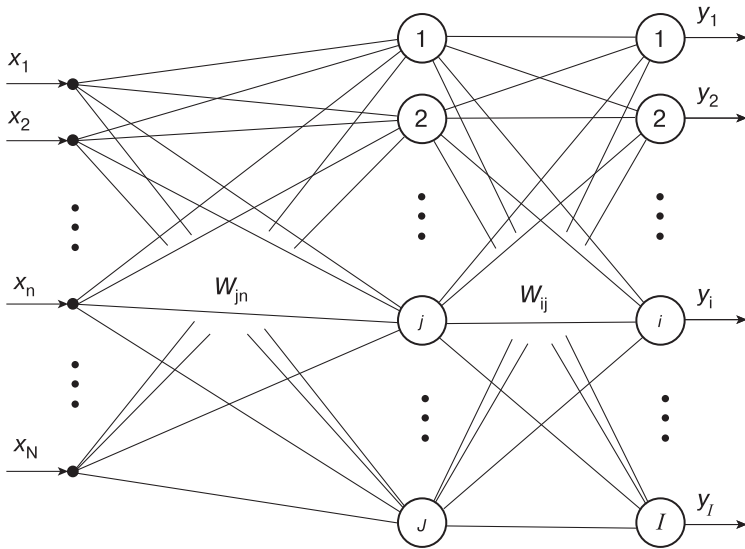


Рис. 3.14. Двухслойный персептрон с N входами, I выходами и скрытым слоем из J нейронов

ного распространения ошибки состояла в том, чтобы в качестве этой ошибки использовать суммарные нейронные ошибки с выходного слоя, умноженные на силы соответствующих синаптических связей, т. е.:

$$(d_j - y_j) = \sum_{i=1}^I \delta_i w_{ij}. \quad (3.20)$$

Тогда для скрытого слоя мы окончательно получаем:

$$\Delta w_{jn} = \eta \delta_j x_n; \quad (3.21)$$

$$\delta_j = y_j(1 - y_j) \sum_{i=1}^I \delta_i w_{ij}. \quad (3.22)$$

Используя эту идею, можно расписать и алгоритм обратного распространения ошибки для обучения персептрона, имеющего произвольное количество скрытых слоев. Однако здесь мы этого делать не будем, а только укажем, что в полном объеме этот алгоритм можно найти в вузов-

ских учебниках, например [23, 24]. Кроме того, отметим, что алгоритм обратного распространения ошибки запрограммирован практически во всех *нейропакетах* — компьютерных программах, предназначенных для работы с нейронными сетями.



Данный алгоритм также используется и для обучения многослойных персептронов при выполнении **лабораторных работ № 5–8** из состава лабораторного практикума, размещенного на сайте **<http://www.LbAI.ru>**.

В связи с этим, естественно, возникает вопрос: зачем нам изучать этот алгоритм, если он уже запрограммирован, и им можно воспользоваться путем нажатия кнопок нейропакета? Действительно, как вы убедитесь при изучении нашего предмета, очень многие задачи можно решать, освоив входящий в наш учебно-методический комплекс лабораторный практикум (на сайте **<http://www.LbAI.ru>**). Однако если у вас появится желание сделать действительно профессиональный, высококачественный программный продукт, предназначенный для коммерческого использования, то вам, скорее всего, придется самим программировать его на языках низкого уровня, в том числе выполнять программную реализацию алгоритмов обучения персептрона.



Коротко о главном

Подведем итоги изучения алгоритмов обучения нейронных сетей.

Первым таким алгоритмом были правила Хебба, предназначенные для обучения однослойного персептрона с нейронами, имеющими ступенчатые активационные функции. Позже было введено понятие нейронной ошибки как разницы между желаемым выходом нейрона d_i и его действительным значением y_i ; в результате алгоритм обучения персептрона с помощью правил Хебба был обобщен в виде алгоритма дельта-правила. В итерационных формулах этого алгоритма также появился коэффи-

циент скорости обучения η , позволяющий влиять на величину итерационного шага.

Далее была предложена сигмоидная активационная функция и введено понятие квадратичной ошибки обучения персептрона. В результате появился алгоритм обобщенного дельта-правила, реализующий метод градиентного спуска и позволяющий работать не только с бинарными, но и с непрерывными сигналами. Алгоритм же обратного распространения ошибки является развитием обобщенного дельта-правила и позволяет обучать не только однослойные, но и многослойные персептроны.



Вопросы и задания

1. В чем состоит идея алгоритма обратного распространения ошибки? Отражает ли название этого алгоритма его основную идею?
2. Какую роль в методе обратного распространения ошибки выполняет коэффициент скорости обучения η ?
3. Сколько алгоритмов обучения нейронных сетей вы теперь знаете? Назовите их и охарактеризуйте их возможности.
4. Годится ли алгоритм обратного распространения ошибки для обучения персептрона со ступенчатыми активационными функциями?
5. Годятся ли правила Хебба для обучения персептрона с нейронами, имеющими сигмоидные функции активации?
6. Годится ли дельта-правило для обучения персептрона с нейронами, имеющими сигмоидные функции активации?
7. Назовите преимущества и недостатки алгоритма обратного распространения ошибки по сравнению со всеми изученными ранее методами обучения нейронных сетей.

3.1.9. Виды активационных функций

Как мы уже поняли из предыдущих разделов, активационные функции осуществляют преобразование взвешенной суммы входных сигналов нейрона:

$$\Sigma = \sum_{j=1}^J w_j x_j$$

в его выходной сигнал y . В современных реализациях средств нейросетевых технологий используют нейроны с различными типами активационных функций. Перечислим некоторые из них.

Пороговые активационные функции (функции-ступеньки). Пороговую активационную функцию имеет, в частности, математический нейрон Мак-Каллока — Питтса, с которого мы начали изучение нейроинформатики. Типичный вид пороговой активационной функции был приведен на рис. 3.3, а ее математическое выражение — представлено формулой (3.2).

Пороговые активационные функции (их также называют ступенчатыми) обычно применяют при решении задач распознавания образов, классификации объектов, диагностики и в других задачах, требующих бинарных ответов: «да» или «нет», «единица» или «нуль».

Персептроны со ступенчатыми активационными функциями обучают с помощью правил Хебба и дельта-правила. Обобщенное дельта-правило и алгоритм обратного распространения ошибки для обучения таких персептронов непригодны, так как они являются градиентными, а значит, предполагают нахождение производных от активационных функций, что возможно только для непрерывных функций.

Линейные активационные функции. На рис. 3.15а представлен график линейной активационной функции:

$$y = S,$$

область изменения которой неограничена. Такие активационные функции обычно применяются в нейронах входного слоя. Нейронные сети с линейными активационными

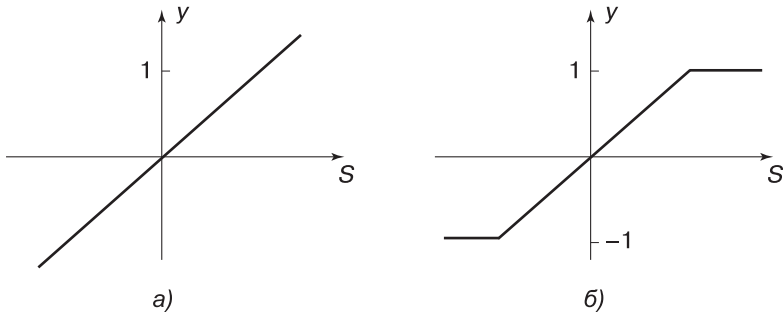


Рис. 3.15. Линейные активационные функции с неограниченной (а) и ограниченной (б) областями изменения

функциями также неплохо работают при решении простых линейно разделимых задач, причем для обучения таких персептронов пригодны алгоритмы дельта-правила, обобщенного дельта-правила и обратного распространения ошибки.

Иногда также применяют линейные активационные функции с ограниченной областью изменения (рис. 3.15б):

$$y = \begin{cases} 1, & \text{если } S < -1, \\ S, & \text{если } -1 \leq S \leq 1, \\ -1, & \text{если } S > 1. \end{cases}$$

Сигмоидные активационные функции. На рис. 3.16а изображен график сигмоидной функции, заданной уравнением:

$$y = \frac{1}{1 + e^{-\alpha S}},$$

а на рис. 3.16б — график функции, заданной уравнением:

$$y = \frac{1 - e^{-\alpha S}}{1 + e^{-\alpha S}}.$$

В обоих этих уравнениях коэффициент α влияет на угол наклона сигмoиды к оси S .

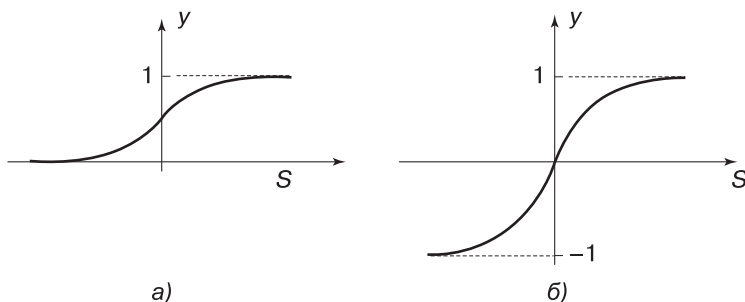


Рис. 3.16. Сигмоидные активационные функции с несимметричной (а) и симметричной (б) областями изменения

Аналогичный представленному на последнем графике вид имеет функция арктангенса $y = \frac{2}{\pi} \arctg \alpha S$, а также функция $y = \frac{\alpha S}{1 + |\alpha S|}$, которые тоже называют сигмоидными.

Область изменения сигмоиды, изображенной на рис. 3.16а, ограничена интервалом $(0, +1)$, а на рис. 3.16б — интервалом $(-1, +1)$.

Персептроны с сигмоидными активационными функциями обычно обучают с помощью алгоритма обратного распространения ошибки. Если же персептрон однослойный, то для его обучения также может использоваться дельта-правило и обобщенное дельта-правило.

Область применения нейронных сетей с сигмоидными активационными функциями чрезвычайно широка. Кроме задач распознавания, классификации и диагностики, они успешно применяются в задачах прогнозирования и управления, примеры решения которых мы рассмотрим позже.

Логарифмические активационные функции. На рис. 3.17 представлен график активационной функции, заданной уравнением:

$$y = \ln \left(S + \sqrt{S^2 + 1} \right).$$

В отличие от сигмоиды, эта функция имеет неограниченную область изменения. В некоторых случаях, рассмат-

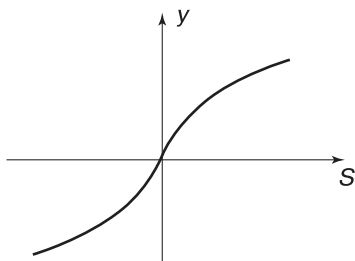


Рис. 3.17. Логарифмическая активационная функция

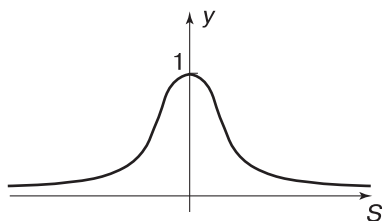


Рис. 3.18. Радиально-базисная активационная функция

риваемых далее, использование логарифмических активационных функций вместо сигмоды облегчает процесс обучения персептронов.

Радиально-базисные активационные функции. В последнее время получают распространение нейросети, нейроны которых имеют активационные функции в виде кривой, изображенной на рис. 3.18. Такие активационные функции называют *радиально-базисными (RBF)*, а соответствующие нейронные сети — *RBF-сетями*.

RBF-сети являются сравнительно новым разделом нейроинформатики, которому не исполнилось еще и 20 лет. Есть мнение, что RBF-сети в некоторых случаях имеют преимущества перед нейросетями персептронного типа, поэтому они включены во многие современные нейропакеты. Однако в нашем элективном курсе RBF-сетями мы заниматься не будем.



При выполнении лабораторных работ из состава лабораторного практикума, размещенного на сайте <http://www.LbAI.ru>, у вас будет возможность конструировать и использовать нейронные сети с нейронами, имеющими ступенчатые, сигмодные и логарифмические активационные функции.



Коротко о главном

Активационные функции осуществляют преобразование взвешенной суммы входных сигналов нейрона в его выходной сигнал. От вида активационных функций зависят функциональные возможности нейронных сетей, а также выбор способов их обучения.



Вопросы и задания

1. Приведите примеры активационных функций, используемых в современных нейропакетах. Постройте их графики. Для каждой функции укажите область ее определения и область изменения.
2. Какие алгоритмы обучения персептронов пригодны для работы со ступенчатыми активационными функциями?
3. Какие алгоритмы обучения персептронов пригодны для работы с сигмоидными активационными функциями?
4. Какие алгоритмы обучения персептронов пригодны для работы с логарифмическими активационными функциями?
5. Какое преимущество дает использование сигмоидной активационной функции вместо функции-ступеньки?
6. Каковы преимущества и недостатки использования логарифмической активационной функции вместо сигмоидной?
7. Какие преимущества и недостатки может дать использование радиально-базисных активационных функций?

3.2. Возможности и сферы применения персептронов

3.2.1. Новый способ построения математических моделей

С появлением алгоритма обратного распространения ошибки начался период широкого практического применения нейросетевых технологий для решения самых разных задач. С помощью многослойного персептрона стало возможным построение математических моделей, выполняющих сложные многомерные преобразования входного комплекса параметров $x_1, x_2, \dots, x_N$ в выходной комплекс параметров $y_1, y_2, \dots, y_I$.

В нейроинформатике входной комплекс параметров принято обозначать буквой X , называя его *входным вектором*, компонентами которого являются входные параметры $x_1, x_2, \dots, x_N$. Аналогично, выходной комплекс параметров обозначают буквой Y , называя его *выходным вектором*, компонентами которого являются выходные параметры $y_1, y_2, \dots, y_I$.

Задачи построения математических моделей, выполняющих преобразование входного вектора X на выходной вектор Y , часто встречаются в самых разнообразных и, казалось бы, не имеющих ничего общего областях, таких как промышленность, экономика, бизнес, финансы, политология, социология, криминалистика, экология, медицина и т. д. Практически в каждой проблеме, решаемой прикладными науками, требуется построить модель явления, процесса или объекта, т. е. выявить и математически описать зависимость одного комплекса параметров от другого. Обычно при этом требуется построить математические функции, которые можно использовать для более глубокого анализа моделируемого объекта, например, найти оптимальное сочетание управляющих параметров, обеспечивающих максимум некоторой целевой функции — рентабельности, прибыльности, прочности, температуры, скорости, высоты и т. д. Либо требуется выполнить

прогнозирование, т. е. предсказать, как будут развиваться моделируемые события в зависимости от того или иного воздействия на моделируемый объект, и определить, как можно повлиять на эти события путем выбора нужного воздействия.

При обучении в школе, да и просто в повседневной жизни вы, наверное, не раз сталкивались с *методом математического моделирования* и оценили его эффективность. Например, на уроках физики вы решали задачу о движении тела, брошенного под углом к горизонту. Используя физические законы, вы построили математическую модель этого явления, связав уравнениями входные и выходные параметры: угол бросания, начальную скорость, высоту подъема и дальность полета тела. Исследуя эту математическую модель, решая уравнения и просчитывая разные варианты выбора входных параметров, вы легко определяли оптимальный угол бросания, обеспечивающий максимальную дальность полета тела, и максимальную высоту его подъема. Для этого вам не надо было ставить никаких натуральных экспериментов, не требовалось выходить во двор и бросать камни.

Говоря о важности метода математического моделирования в нашей жизни, отметим, что в настоящее время он является одним из самых эффективных методов получения результатов как в науке, так в практической деятельности. Методом математического моделирования рассчитываются, проектируются, оптимизируются новые инженерные и строительные конструкции, делается прогноз погоды, предсказываются стихийные бедствия, выполняются экономические прогнозы, на основе которых строится финансовая политика как отдельных фирм, так и целых государств.

Новые научные и практические знания, полученные методом математического моделирования, не раз оказывали решающее влияние на формирование нашей цивилизации, на геополитическое положение государств на нашей планете. Например, определение формы крыла самолета, обеспечивающей максимальную подъемную силу, выпол-

ненное методом математического моделирования, дало сильнейший толчок развитию авиации — как гражданской, так и военной. На основе математического моделирования процессов горения пороха была построена теория кумулятивного взрыва. В результате были созданы противотанковые заряды, применение которых решающим образом повлияло на исход Второй мировой войны. Создание космических ракет, да и сами космические полеты были бы невозможными без компьютерных экспериментов, выполняемых методами математического моделирования. И наконец, благодаря методу математического моделирования было открыто явление «ядерной зимы» — глобального понижения температуры поверхности планеты, вызванного массовыми ядерными взрывами. Никто это явление в действительности еще, слава богу, не наблюдал — оно было открыто только на экране компьютера в результате вычислительных экспериментов над математическими компьютерными моделями. И это открытие радикальным образом повлияло на государственную политику великих держав: стало ясно, что победителей в ядерной войне не будет! Бессмысленная гонка вооружений была прекращена, а глобальный мир на Земле был сохранен.

До появления нейронных сетей и нейрокомпьютеров математические модели традиционно строились на основе фундаментальных законов природы (таких как законы сохранения массы, энергии, количества движения и др.), записанных в виде алгебраических либо дифференциальных и интегральных уравнений, к которым добавлялись уравнения, отражающие закономерности конкретных предметных областей. Для получения результата приходилось разрабатывать и применять алгоритмы совместного решения всех этих уравнений, составляющих математическую модель исследуемой предметной области.

Нейроинформационные технологии открыли иной подход к самой методике построения компьютерных математических моделей. Появилась возможность переложить часть высокоинтеллектуальной работы по созданию математических моделей на сам компьютер. Появилась и воз-

возможность, не задумываясь над законами физики, химии, биологии, медицины, общественного развития и т. д., а исходя из одного только эмпирического опыта в виде обучающих примеров, строить математические модели, которые *сами* извлекают эти законы и позволяют эффективно использовать их для решения широкого круга практических задач. Появился новый инструмент *извлечения знаний из данных*, позволяющий заново открывать фундаментальные законы природы, выявлять ранее неизвестные и никогда не исследованные зависимости и закономерности и использовать их для решения конкретных практических задач. Этот новый инструмент оказался особенно эффективен при построении математических моделей в плохо формализуемых предметных областях, в которых многие законы, закономерности и внутренние взаимосвязи неизвестны человеку. К одной из таких трудных для моделирования предметных областей относится медицина.

3.2.2. Диагностика в медицине

Проведем наблюдение за тем, как врач ставит диагноз болезни пациента. Прежде всего он выясняет и записывает имя, возраст, пол, место работы пациента; затем, как правило, врач измеряет давление, проводит внешний осмотр, выслушивает жалобы больного, знакомится с историей его болезни и результатами анализов, изучает электрокардиограмму. В результате у врача накапливается от 20 до 100 и более параметров, характеризующих пациента и его состояние здоровья. Это и есть исходные параметры, составляющие входной вектор X . Обработав эти параметры с помощью своих медицинских знаний и опыта, врач делает заключение о заболевании пациента — ставит ему диагноз.

Задавшись целью построить нейросетевую математическую модель врача-диагноста, мы прежде всего должны определить входной вектор X и выходной вектор Y (их содержимое).

В качестве компонентов вектора X логично предусмотреть параметры, которые врач выясняет у больного. Например, в качестве значения компоненты x_1 можно задать дату

рождения, в качестве x_2 закодировать пол (например, нулем или единицей), в качестве x_3 — вес больного, x_4 — артериальное давление, x_5 — температуру тела и т. д. Нелишне учесть также цвет глаз, цвет волос, знак зодиака и другие данные, определяющие особенности организма и, следовательно, влияющие на вероятность возникновения тех или иных заболеваний (или, по крайней мере, параметры, которые можно заподозрить в таком слиянии).

В выходном векторе $\mathbf{Y}$ следует закодировать все возможные диагнозы болезней, которые способен обнаружить врач. Естественно, что количество компонентов вектора $\mathbf{Y}$ можно существенно снизить, если моделировать врача, специализирующегося в узкой области медицины. Так, если мы выберем модель врача-кардиолога, то с помощью компонентов вектора $\mathbf{Y}$ потребуются кодировать только кардиологические заболевания. Например, можно принять $y_1 = 1$, если больному ставится диагноз «инфаркт», и $y_1 = 0$, если инфаркта нет. Аналогично, с помощью y_2 можно закодировать наличие или отсутствие порока сердца, y_3 — ишемической болезни сердца и т. д. Таким образом, выходной вектор персептрона $\mathbf{Y}$ будет состоять из множества нулей и одной или нескольких единиц (если болезней несколько). Однако диагнозы болезней лучше кодировать по пяти-, десяти- или стобалльной шкале. Тогда на этапе подготовки обучающего множества примеров с помощью баллов можно учитывать степень уверенности врача в правильности его диагноза или степень развитости заболевания, а на этапе эксплуатации — вероятность правильного ответа персептрона. Например, если врач подозревает, что у больного инфаркт миокарда, а также знает, что у больного нет порока сердца, и полностью уверен, что больной страдает ишемической болезнью сердца, то он может указать: $y_1 = 20\%$, $y_2 = 0\%$, $y_3 = 100\%$.

Далее следует подготовить множество обучающих примеров, состоящее из множества пар входных и выходных векторов. Однако на этом этапе выходной вектор персептрона и его компоненты мы будем обозначать буквами $\mathbf{D}$ и $(d_1, d_2, \dots)$, подразумевая под ними жела-

емые (как вы помните, образованные от слова *desire* — «желать») выходные значения персептрона, оставив за $\mathbf{Y}$ и $(y_1, y_2, \dots)$ действительные выходные значения (те, которые персептрон будет вычислять после его обучения). Напомним, что ошибка обучения персептрона вычисляется как разница между его желаемыми $(d_1, d_2, \dots)$ и действительными $(y_1, y_2, \dots)$ выходными значениями. В дальнейшем мы также должны не забывать, что желаемые выходные значения персептрона образуют вектор $\mathbf{D}$, а действительные выходные значения — вектор $\mathbf{Y}$, и не путать их между собой.

Воздержимся от советов по организации совместного труда эксперта-врача и программиста, в результате которого будет создано необходимое количество обучающих примеров. Отметим только, что качество нейросетевой диагностической системы напрямую зависит от квалификации практикующего врача, на примерах работы которого она обучается. Дело в том, что нейронная сеть «наследует» от врача не только его знания, но и все пробелы в его медицинском образовании! Понятно, что она будет допускать те же самые врачебные ошибки, которые допускает врач, поэтому для обеспечения высокого качества диагностики нейронную сеть следует обучать на примерах работы высококвалифицированного врача или даже на результатах работы целого врачебного консилиума. А если к работе по обучению нейронной сети привлечь еще и патологоанатома, исключающего ошибки врачебной диагностики, то появятся все основания надеяться, что обученная таким способом нейросеть по качеству выставляемых диагнозов превзойдет большинство обычных врачей. Более того, — подобная нейронная сеть может обнаружить и заложить в модель такие закономерности человеческого организма, которые современной медицине вообще неизвестны!

Итак, в результате совместной работы коллектива специалистов-медиков и программистов будет накоплено множество обучающих примеров, состоящее из множества пар векторов $\mathbf{X}_q$ и $\mathbf{D}_q$, где $q = 1, 2, \dots, Q$, q — номер обучающего примера (номер пациента), а Q — их общее количество. Теперь задача состоит в том, чтобы спроектировать персеп-

трон и путем обучения передать ему знания и опыт, содержащиеся в этом множестве обучающих примеров. Вопросы проектирования персептронов, т. е. подбора количества скрытых слоев, содержащихся в них нейронов и типов активационных функций рассматриваются далее в разделе 3.3, поэтому сейчас мы этим заниматься не будем. В качестве же метода обучения такого персептрона можно использовать рассмотренный нами ранее алгоритм обратного распространения ошибки.

В результате наш персептрон должен научиться отображать любой входной вектор обучающего множества $\mathbf{X}_q$ на выходной вектор $\mathbf{Y}_q$, совпадающий (либо почти совпадающий) с выходным вектором желаемых ответов $\mathbf{D}_q$. Кроме того, при появлении нового пациента, характеризующегося новым входным вектором $\mathbf{X}$, персептрон должен вычислять для него новый вектор $\mathbf{Y}$, содержащий правильный диагноз, поставленный персептроном уже без помощи врача-эксперта. Другими словами, персептрон должен уметь *обобщать* переданный ему опыт на новые, не встречавшиеся ранее примеры предметной области, — ставить диагнозы болезней новым, не встречавшимся ранее пациентам.

В заключение еще раз укажем причины, на основании которых можно ожидать, что такой «искусственный» врач может превзойти натурального. Во-первых, качество работы искусственного врача всегда стабильно и не зависит от его настроения и состояния здоровья. Во-вторых же (и это главное) — нейронная сеть способна извлекать и применять знания, которые современной медицине еще неизвестны, поэтому есть все основания ожидать, что в будущем благодаря применению методов искусственного интеллекта несовершенство современной медицины будет в значительной степени ликвидировано.



При выполнении **лабораторных работ № 6 и 7** вам будет предоставлена возможность, пользуясь своими собственными познаниями в медицине, придумать небольшое множество обучающих примеров и научить нейронные сети ставить диагнозы таких заболеваний, как грипп, пневмония и ОРЗ.



Коротко о главном

Персептрон можно обучить ставить диагнозы заболеваний, т. е. моделировать деятельность врача-диагноста. При этом такая нейросистема может по качеству своей работы превзойти большинство обычных врачей.



Вопросы и задания

1. Когда появился метод математического моделирования?
2. Приведите примеры использования математического моделирования в различных областях науки и техники.
3. Какие параметры следует принять в качестве входных, а какие — в качестве выходных при построении модели движения тела, брошенного под углом к горизонту?
4. Чем методика построения математических моделей на основе нейротехнологий отличается от традиционной?
5. Опишите, как вы стали бы формировать примеры для обучения персептрона постановке медицинских диагнозов?
6. Сколько ваш персептрон должен иметь входов и выходов?
7. Как быть, если врач-эксперт не совсем уверен в правильности выставляемого им диагноза?
8. Почему искусственный нейросетевой врач может превзойти обычного врача по качеству постановки диагнозов заболеваний?
9. Откуда нейросетевой врач получает медицинские знания и в каком виде он хранит их в своей памяти?

3.2.3. Диагностика неисправностей сложных технических устройств

Диагностика неисправностей технических устройств считается менее сложной проблемой, чем диагностика заболеваний человека. Это утверждение относится к методу математического моделирования, основанному на законах природы — физики, химии, биологии и др. Дело в том, что для технического устройства, каким бы сложным оно ни было, всегда известно функциональное назначение любого сколь угодно мелкого узла, чего нельзя сказать о человеке. Конечно, это обстоятельство является серьезным препятствием, осложняющим построение адекватных математических моделей человека традиционными способами. Нейронная же сеть сама извлекает необходимые знания из обучающих примеров, сама устанавливает еще неизвестные законы и раскрывает функциональные связи между элементами исследуемого объекта. Поэтому в случае нейросетевого моделирования нет принципиальной разницы между диагностикой заболеваний человека и диагностикой неисправностей технического устройства.

Нейросетевой подход к решению задач медицинской диагностики, рассмотренный ранее, можно почти без изменений применить к решению проблемы диагностики неисправностей сложных технических устройств. Продемонстрируем это на примере диагностики неисправностей авиационных двигателей.

Специалисты, занимающиеся этой проблемой, устанавливают датчики, измеряющие параметры работы авиадвигателей во время полетов. Файл данных, полученных в ходе такого полетного мониторинга, обычно содержит следующие параметры: номер и дату полета, общую наработку двигателя, температуру и давление воздуха на входе в двигатель, температуру и давление газа за турбиной, температуру лопаток, уровень и температуру масла в маслоблоке и т. д. Количество таких полетных параметров может достигать сотни и более (что, кстати, соизмеримо с числом параметров больного при постановке диагноза его болезни).



Рис. 3.19. После выполнения определенного количества полетов двигатель снимается с самолета и подвергается стендовой разборке



После выполнения 200–300 полетов двигатель снимают с самолета и подвергают стендовой разборке (рис. 3.19), во время которой выявляют и устраняют его дефекты. Характерными дефектами авиадвигателей при этом являются трещины на сопловом агрегате, забоины, разрушение дефлектора, наличие стружки в масле и т. д. (всего около 30 видов дефектов).

Задача инженера-диагноста состоит в том, чтобы, используя данные мониторинга, выявить дефекты двигателя *до* его профилактической разборки. Традиционно эта задача решается с помощью методик, основанных на физических закономерностях: каждый дефект вызывает определенные отклонения тех или иных полетных параметров работы двигателя, поэтому, анализируя характер их изменения, можно сделать предположения о наличии дефектов, вызывающих эти изменения. Понятно, что ввиду значительных объемов информации и сложности существующих взаимосвязей между дефектами и измеренными параметрами задача анализа данных полетного мониторинга и выявления дефектов авиадвигателей является далеко не тривиальной и во многих случаях решается ненадежно и некачественно.

Рассмотрим теперь, как эту задачу можно решать с помощью нейросетевого моделирования. Прежде всего отметим, что во входном векторе персептрона X следует предусмотреть места для всех параметров полетного мониторинга, на значения которых оказывают влияние выявляемые дефекты. Возможные дефекты авиадвигателя можно закодировать в выходном векторе D с помощью все тех же нулей и единиц. Векторы желаемых выходов D_q составляются по результатам стендовых разборок двигателей. В отличие от медицинской диагностики здесь уже нет необходимости вводить многобалльную шкалу диагнозов, поскольку все диагнозы в обучающем множестве примеров имеют стопроцентную достоверность (т. е. ситуация аналогична случаю, когда диагноз болезни ставит патологоанатом при вскрытии пациента в морге).

Кроме того, разумно предположить, что при первых вылетах нового или только что отремонтированного самолета его двигатель полностью исправен и дефектов в нем нет, тогда как перед самым завершением полетного мониторинга двигатель уже имел те самые дефекты, которые и выявились при его разборке. Поэтому из всего множества параметров полетного мониторинга ценность для обучения нейросети имеют параметры именно первого и последнего полетов самолета. Тогда для каждого двигателя, побывавшего в разборке, нужно сформировать пару обучающих векторов X_q и D_q .

Если количество обучающих примеров будет достаточным, то правильно спроектированный многослойный персептрон обучится надежно ставить диагноз неисправности авиационных двигателей, в том числе и тех, которые в обучающем множестве не присутствовали. А если таким персептроном оборудовать бортовой компьютер самолета, то он будет сообщать о появлении дефекта двигателя в режиме реального времени — как только сложится соответствующая конфигурация вектора входных параметров — результатов измерений, снимаемых во время полета самолета. (Конечно, прежде чем попасть в кабину пилотов, сигналы персептрона должны обрабатываться компьютером, выра-

батывающим инструкции об адекватных действиях экипажа при создавшихся условиях.)

Укажем и еще одно преимущество нейросетевого метода диагностики перед традиционными диагностическими методиками, основанными на явных знаниях. Специалисты, занимающиеся традиционными методами диагностики, пытаются учесть как можно больше взаимосвязей между значениями измеряемых полетных параметров и появлением тех или иных дефектов двигателя. В сложных технических устройствах количество таких взаимосвязей настолько велико, а характер самих взаимодействий настолько сложен, что построить полную математическую модель, в должной мере адекватную моделируемому устройству, на современном этапе развития точных наук практически невозможно. Кроме того, в сложных технических устройствах существуют и такие взаимосвязи, о которых специалисты просто пока еще не знают. Встречаются также взаимосвязи, о которых специалисты догадываются, но объяснить их физическую природу пока не могут, а потому в расчет их не принимают. Например в двигателестроении известен дефект под названием «стружка в масле»: при работе трущихся частей иногда образуется мелкая стружка, которая падает в поддон с маслом. Насколько известно специалистам, появление этого дефекта не влияет ни на один из измеряемых датчиками параметров работы двигателя, поэтому не существует и каких-либо традиционных методов, способных выявлять этот дефект.

А вот в практике нейросетевого моделирования однажды встретился случай, когда нейронная сеть появление стружки в масле все-таки обнаруживала! Этот случай, конечно, вызвал удивление и непонимание у специалистов, но после длительных обсуждений и дискуссий парадокс объяснили тем, что, по-видимому, не оказывая заметного влияния на каждый отдельно взятый параметр двигателя, появление стружки все-таки влияет на общую конфигурацию всего вектора параметров, что и вызывает соответствующую реакцию нейронной сети.



Коротко о главном

Персептрон можно обучить обнаружению неисправностей сложных технических устройств. Он способен выявлять такие дефекты, которые обычными диагностическими системами не обнаруживаются.



Вопросы и задания

1. Опишите, как вы формировали бы примеры для обучения персептрона постановке диагнозов неисправностей автомобильного двигателя.
2. Сколько такой персептрон, диагностирующий неисправности автомобильных двигателей, должен, по вашему мнению, иметь входов и выходов?
3. Может ли нейросетевая диагностическая система превзойти по качеству постановки диагнозов традиционную диагностическую систему, основанную на явных знаниях (экспертную систему)? Объясните, почему.
4. Откуда нейросетевая диагностическая система получает технические знания и в каком виде она хранит их в своей памяти?

3.2.4. Нейросетевой «детектор лжи»

Правду ли говорит ребенок, обычно легко определить по выражению его лица, движению глаз, покраснению кожи. Ложь взрослого человека выявить значительно труднее. Если измерять давление крови, то можно выяснить, что у одних людей, говорящих неправду, оно повышается, а у других, наоборот, понижается. То же самое может происходить с пульсом, с дыханием, с движением глаз и т. д.

В следственной практике МВД России в настоящее время применяются «детекторы лжи» (*полиграфы*), система датчиков которых измеряет от 4 до 10 физиологических показателей (пульс, артериальное давление, частоту и глу-



Рис. 3.20. Экран полиграфа с графическим изображением физиологических параметров, отражающих состояние человека

бину дыхания, электросопротивление участков кожи и др.). Эти параметры в режиме реального времени отображаются на экране монитора (рис. 3.20) в виде пульсирующих кривых. Заключение о правдивости ответа подследственного дается компьютерной программой, которая анализирует получаемые кривые с помощью набора правил, обобщающих исследования психологов, физиологов и специалистов в области инструментальной детекции лжи.

Ненадежность заключения, вырабатываемого таким «детектором лжи», обусловлена тем, что к разным людям, по-разному реагирующим на стрессовые ситуации, применяется одна и та же система решающих правил. Так, широко распространенный в МВД России полиграфный аппарат системы «Эпос» дает правильные заключения только в 70% случаев. Поэтому, кроме компьютерной программы, специалисты-полиграфологи вынуждены применять це-

лую систему дополнительных, весьма трудоемких и кропотливых приемов. В результате степень объективности заключения об истинности ответов в значительной мере зависит от грамотности, компетентности и моральных качеств специалиста, проводящего допрос.

Нейросетевые технологии позволяют по-новому подойти к проблеме построения «детектора лжи». Они дают возможность создать компьютерную программу, которая может настраиваться на каждого конкретного человека и учитывать индивидуальные особенности его организма.

Принципиально задача выявления признаков лжи с помощью нейросетевых технологий ничем не отличается от рассмотренных ранее задач диагностики. В качестве входного вектора персептрона $\mathbf{X}$ можно использовать тот же самый набор параметров, что и в стандартном полиграфе. Выходной вектор $\mathbf{D}$ целесообразно принять состоящим всего из двух параметров (рис. 3.21): $d_1 = 1$, $d_2 = 0$, если опрашиваемый человек сказал правду, и $d_1 = 0$, $d_2 = 1$, если он лжет. Можно использовать и персептрон с одним-единственным выходом, значение которого $d = 1$, если человек говорит правду, и $d = 0$, если он лжет.

Множество обучающих примеров формируется в результате предварительных бесед оператора-полиграфолога

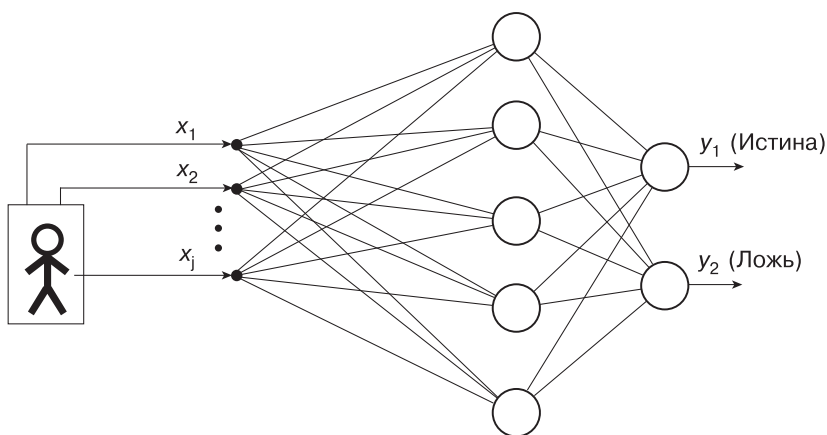


Рис. 3.21. Принципиальная схема нейросетевого «детектора лжи»

с обследуемым человеком. Оператор задает вопросы, ответы на которые ему известны. С опрашиваемого человека во время ответов снимаются физиологические параметры, формирующие входной вектор X_q . Выходной же вектор D_q формирует оператор, сопоставляя полученные ответы с тем, что ему заранее известно.

Накопив достаточное количество примеров и обучив на них персептрон, можно уже задать опрашиваемому человеку тот вопрос, ответ на который интересует следователя. Система датчиков сформирует и передаст персептрону вектор X , а персептрон выдаст в ответ вектор Y , позволяющий сделать заключение об истинности полученного ответа. Причем, поскольку персептрон обучался на физиологических реакциях, сформированных самим подследственным, то логично ожидать, что заключение такого персептрона будет объективно учитывать индивидуальные особенности организма допрашиваемого человека и его индивидуальные физиологические реакции на ложь и правду. И, как показал опыт, такой нейросетевой полиграф действительно обеспечивает правильные заключения более чем в 97% случаев.

Описанный способ создания настраиваемого «детектора лжи» обладает недостатком, заключающемся в повышенной трудоемкости: здесь для каждого обследуемого человека приходится заново формировать множество обучающих примеров (а это примерно 70–100 вопросов-ответов!) и обучать персептрон на каждом таком множестве. А после этого персептрон пригоден для тестирования только одного человека, на которого он был настроен. Таким образом, указанный способ, с одной стороны, позволяет проводить обследование с учетом индивидуальных особенностей организма опрашиваемого человека, что существенно повышает степень объективности и качества выяснения истины, а с другой — требует больших затрат труда и времени специалистов. Поэтому его можно рекомендовать лишь для особо важных и ответственных случаев, когда затраты на длительные предварительные беседы следователя с подследственным действительно оправданны.



Коротко о главном

Персептрон можно превратить в «детектор лжи» — научить его определять, правду ли говорит человек.



Вопросы и задания

1. Опишите принцип действия полиграфного аппарата, применяемого в настоящее время в органах МВД России. Чем можно объяснить его низкую надежность?
2. Сколько выходов должен иметь персептрон, предназначенный для распознавания признаков лжи в ответах человека?
3. Подумайте над тем, как обучить нейросетевой полиграфный аппарат обнаруживать случаи, когда его намеренно пытаются обмануть.
4. В каком виде хранятся знания в полиграфном аппарате, применяемом в органах МВД России, и в каком виде они хранятся в нейросетевом «детекторе лжи»?
5. Какой принцип действия «детектора лжи» вам представляется наиболее перспективным? Почему?

3.2.5. Управление кибернетическим объектом

Нейросеть, используемую для управления поведением робота, какого-либо кибернетического устройства либо персонажа компьютерной игры, называют *нейроконтроллером*. Такой управляемый объект находится в среде, которая все время меняется. Заранее предусмотреть при этом все возможные ситуации, разумеется, невозможно, поэтому идея применения нейросети состоит в том, чтобы обучить ее на конечном числе ситуаций, а затем предоставить ей возможность самостоятельно реагировать на ситуации, которых в обучающем множестве не было.

Предположим, что речь идет о компьютерной игре, в которой некий персонаж (например, вражеский солдат) в зависимости от ситуации может предпринять одно из следующих действий:

- 1) атаковать;
- 2) маневрировать;
- 3) спрятаться;
- 4) убежать.

Выбор конкретного действия при этом зависит от того, сколько врагов видит персонаж, насколько хорошо он вооружен и какое у него «здоровье» в данный момент.

В табл. 3.6 приведено множество обучающих примеров. В первом столбце (x_1) размещена информация о количестве врагов в поле зрения персонажа, во втором и третьем (x_2 и x_3) — информация о вооружении персонажа (например, $x_2 = 1$, если персонаж вооружен ножом, и $x_2 = 0$, если его нет; $x_3 = 1$, если у персонажа есть пистолет, и $x_3 = 0$, если его нет). Состояние «здоровья» персонажа закодировано в четвертом столбце: $x_4 = 0$ — если он тяжело ранен; $x_4 = 1$ — если он ранен легко; $x_4 = 2$ — если он полностью здоров.

Таблица 3.6

Множество обучающих примеров действий
персонажа компьютерной игры

Количество врагов (x_1)	Персонаж имеет нож (x_2)	Персонаж имеет пистолет (x_3)	Здоровье персонажа (x_4)	Действие персонажа (d)
0	0	1	1	2 (маневрировать)
1	0	1	1	1 (атаковать)
1	0	0	1	4 (убежать)
1	0	0	0	3 (спрятаться)

Окончание табл. 3.6

Количество врагов (x_1)	Персонаж имеет нож (x_2)	Персонаж имеет пистолет (x_3)	Здоровье персонажа (x_4)	Действие персонажа (d)
1	1	1	0	3 (прятаться)
1	1	0	1	1 (атаковать)
0	1	1	0	2 (маневрировать)
5	1	1	2	1 (атаковать)
50	1	1	2	4 (убежать)

Персептрон, снабженный четырьмя входными и одним выходным нейронами и обученный на примерах табл. 3.6, способен генерировать вполне адекватные рекомендации о действиях персонажа, причем он «не теряется» и в таких непредвиденных ситуациях, которых среди обучающих примеров не было.



Коротко о главном

Персептрон можно обучить управлять поведением робота, кибернетического устройства или персонажа компьютерной игры.



Вопросы и задания

1. Опишите принцип действия персептрона, управляющего персонажем компьютерной игры.
2. Благодаря какому свойству («унаследованному» от мозга) персептрон, управляющий компьютерным персонажем, адекватно реагирует даже на те ситуации, которые не встречались в примерах, на которых его обучали?

3.2.6. Нейросети в банковском деле

Банкротство фирм, кредитруемых банками, и невозможность возврата ими кредитных средств не раз являлись причиной кризисов и банкротств весьма солидных банков (вспомним хотя бы, с чего начинался в США недавний «всемирный кризис»). Поэтому вопрос о том, каким клиентам опасно выдавать кредиты, а каким нет и какова степень кредитного риска, для любого банка является одним из самых главных с точки зрения стабильности его существования.

Обычно клиенты банка — это частные лица и фирмы, занимающиеся различного рода бизнесом. Банки выдают кредиты под проценты, которые являются немаловажной статьей их доходов. Прежде чем принять решение о выдаче кредита, банкиры тщательно изучают бизнес-план кредитруемого проекта, где обычно указывается, как и на что будут расходоваться кредитные деньги. Это может быть приобретение сырья, оборудования или производственного помещения, аренда или покупка транспортных средств и др. Кроме того, в бизнес-плане рассчитываются и указываются сроки и объемы ожидаемой прибыли, сроки выплаты процентов и возврата всего кредита банку. Банкиры также запрашивают все возможные сведения о фирме-клиенте: дату создания и место регистрации фирмы, количество ее филиалов, численность персонала, средний возраст работников и уровень их образования, вид деятельности фирмы, ее оборот и его динамика, сведения об имуществе, недвижимости, транспортных средствах и пр. Далее эксперты кредитного отдела банка, изучив информацию обо всех фирмах, желающих получить кредит, дают свое заключение по каждому проекту. Окончательное же решение о возможности выдачи кредита обычно принимает руководитель банка, учитывая мнения экспертов и руководствуясь своим опытом и интуицией.

В последнее время руководители многих банков Великобритании стали спрашивать совета еще и у нейронной сети. Рассмотрим общие принципы создания и работы такой сети.

«Советчиком» английских банкиров является обычный многослойный персептрон. На его вход подается вектор X , в котором кодируются данные о фирме, подавшей заявку на получение кредита: время жизни фирмы, вид ее деятельности, средний возраст, пол и численность работников, уровень их профессионализма, количество филиалов фирмы и всевозможные экономические показатели за несколько лет. Такой персептрон имеет один выходной нейрон, который по многобалльной шкале вычисляет степень живучести, а следовательно, и платежеспособности фирмы.

Персептрон — советчик банкиров обучается на примерах, взятых из архивных материалов нескольких банков, в которых с английской щепетильностью собран многолетний опыт их кредитной деятельности: параметры фирм, некогда получавших кредиты, и результат их сотрудничества с банком — своевременность выплаты процентов и возврата кредита. По свидетельству банкиров, применявших подобную нейронную сеть, она помогла им выявить ряд потенциальных неплательщиков, вовремя скорректировать финансовую политику банков и таким образом избежать возможных неприятностей.



Коротко о главном

Персептрон можно обучить определению степени надежности клиентов банка и выявлению потенциальных неплательщиков, чтобы он мог давать разумные советы банкирам: каким клиентам можно выдавать кредиты, а каким — нет.



Вопросы и задания

1. Невозврат кредитов, выдаваемых банками фирмам и частным лицам, не раз являлся причиной банкротства банков. Но если это так опасно, то почему банки продолжают выдавать кредиты фирмам и частным лицам? Могли бы банки обойтись без этого вида деятельности?

2. Кто и на каком основании принимает решение о выдаче или отказе в выдаче кредита фирмам и частным лицам?
3. Объясните принцип действия персептрона, способного распознавать потенциально ненадежных клиентов банка.
4. Чем объяснить, что английские банкиры успешно применяют нейросети для выявления потенциально ненадежных клиентов, а в России это до сих пор не получается?

3.2.7. Прогнозирование курсов валют

Прогнозирование — это одна из самых востребованных задач, возникающих в самых разных областях человеческой деятельности. Задача прогнозирования в общем случае состоит в получении оценки будущих значений упорядоченных во времени данных на основе анализа уже имеющихся данных. Обычно речь идет о прогнозировании временного ряда, т. е. о получении совокупности значений прогнозируемого параметра на некотором интервале времени.

Существуют различные подходы к решению задач прогнозирования — от построения сложнейших нестационарных математических моделей, учитывающих физические, химические, биологические и другие законы природы, до применения статистических методик поиска зависимостей прогнозируемых параметров от времени. Однако в последнее время с традиционными подходами успешно конкурируют нейросетевые технологии, которые особенно хорошо зарекомендовали себя при прогнозировании финансовых показателей — котировок ценных бумаг, курсов валют, общих экономических индексов.

Рассмотрим пример применения нейросети для прогнозирования на три дня вперед курса американского доллара по отношению к российскому рублю.

На рис. 3.22 представлена номограмма, изображающая изменения курса доллара США с марта по май 2003 г.

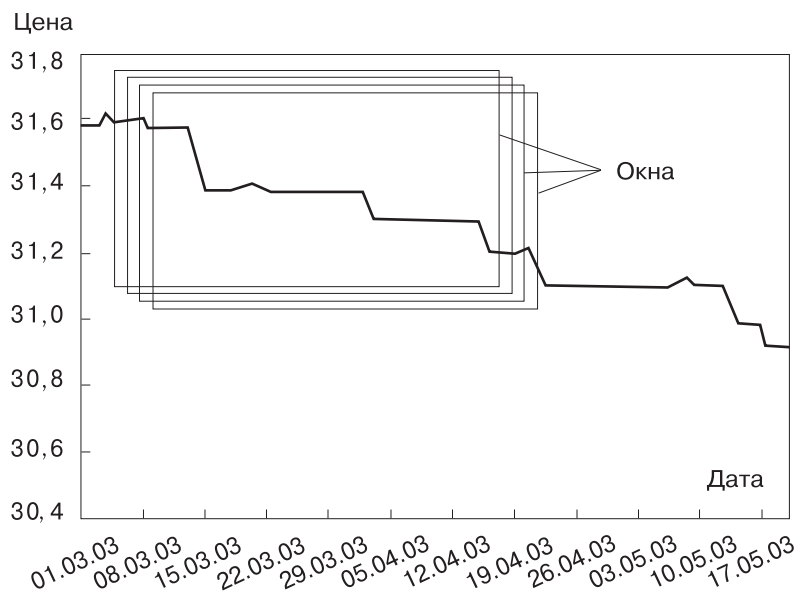


Рис. 3.22. Номограмма изменения курса американского доллара по отношению к российскому рублю за март-май 2003 г.

Множество обучающих примеров можно сформировать на ее основе с использованием *метода скользящих окон*. Согласно ему выбирается временной интервал (например, в 45 дней: от 4 марта до 17 апреля). В качестве x_1 задается курс доллара, который был зафиксирован 4 марта; в качестве x_2 — курс на 5 марта и т. д., а в качестве x_{45} — курс на 17 апреля. В качестве желаемого выхода сети d_1 тогда принимается курс доллара на 18 апреля, d_2 — курс на 19 апреля, а d_3 — курс на 20 апреля. Таким способом будет сформирован первый обучающий пример $X_1 — D_1$ для персептрона, имеющего 45 входов и 3 выхода.

Для формирования второго обучающего примера сдвинем окно на одну позицию (на один день) вправо и выполним аналогичные операции. Перемещая же это окно Q раз, мы получим множество из Q обучающих примеров.

Далее следует определиться с количеством внутренних слоев персептрона и количеством нейронов во внутренних

слоях, однако эти вопросы мы пока затрагивать не будем (они будут изучаться далее в разделе 3.3). Затем производится обучение персептрона, например, методом обратного распространения ошибки. Проверить качество получившейся прогностической программы, оценить точность прогноза можно на тестовых примерах, которые легко получить, располагая окна так, чтобы в них не попадали использованные при обучении даты. Персептрон будет выполнять рабочий прогноз на три дня вперед, если окно расположить в самом конце номограммы — так, чтобы в качестве x_{45} оказался курс доллара на сегодня.

Как известно, качество прогностических программ зависит от полноты учета факторов, влияющих на прогнозируемый параметр. На курс доллара США влияет политическая и экономическая обстановка в Америке и в России, в странах Европы, Азии и других континентов, наличие или отсутствие военных конфликтов на планете, результаты президентских выборов, факты или ожидания террористических актов либо стихийных бедствий и др. На курс доллара США также влияют погодно-климатические изменения, эффекты начала и конца рабочей недели, месяца или года, выходные и праздники. На курс доллара, если верить утверждениям финансистов, оказывают влияние даже явления космической природы: ведь всплески солнечной активности и магнитные бури вызывают изменение самочувствия, психологического состояния и жизненной активности многих людей, и это сказывается на экономическом состоянии отдельных фирм, регионов или даже стран, подвергшихся таким космическим воздействиям. «Барометром», фиксирующим такие изменения, как раз и служат курсы валют, котировки акций и общие экономические индексы. При этом все они являются функциями огромного количества аргументов, а их значения формируются в результате совместного действия множества разнообразных зависимостей и закономерностей, выявить и изучить которые традиционными способами не представляется возможным.

Рассмотрим, как эти вопросы можно решать с помощью нейросетевых технологий. Например, чтобы учесть эффект начала и конца рабочей недели, можно добавить в наш персептрон еще один входной нейрон, в качестве входного сигнала которого задавать единицу, если прогнозируемый день является понедельником, двойку, если он приходится на пятницу, или нуль для любого другого дня недели. Аналогично, посредством добавления новых входных нейронов можно учесть баллы солнечной активности, фазы Луны, эклиптические долготы планет, их положения в знаке зодиака, скорости и склонения и другие астрономические и астрологические события, приходящиеся на рассматриваемый период времени. Разумеется, многие из этих параметров не являются *значащими*, т. е. не влияют на интересующие нас прогнозируемые величины. Решить вопрос о значимости того или иного из этих факторов можно будет опять же с помощью нейросети, наблюдая на тестовых примерах, приводит ли добавление соответствующего входного нейрона к уменьшению ошибки прогноза.

Подводя итог, отметим, что в настоящее время брокеры, финансовые игроки, экономисты, бизнесмены разных стран активно используют нейросети, учитывающие множество факторов, в том числе астрологических. Нейросети стали незаменимым инструментом для выявления скрытых, ранее не известных и не исследованных закономерностей. В результате многие науки, считавшиеся ранее неофициальными (например финансовая астрология), получили мощный толчок для теоретических и прикладных исследований.



Коротко о главном

Персептрон можно научить выполнять прогнозы различных финансовых параметров — курсов валют, котировок акций и ценных бумаг, общих экономических индексов и пр.



Вопросы и задания

1. Объясните основную идею метода скользящих окон.
2. Какие факторы могут оказывать влияние на курсы валют? Как их можно учитывать при нейросетевом прогнозировании?
3. Как с помощью нейросети можно решить вопрос о значимости того или иного фактора?

3.2.8. Прогнозирование результатов выборов президента страны

Результаты выборов президента страны зависят от многих факторов. Это экономическая и политическая обстановка в мире, активность средств массовой информации, активность и степень политизированности самих избирателей, взаимоотношения и политические ходы партий, качество проведения предвыборной кампании и многое другое. Учесть и правильно оценить влияние всех этих факторов на исход выборов чрезвычайно сложно. Но есть и такие факторы, которые определяются только самой личностью претендента на президентское кресло: его возраст, пол, семейное положение, количество детей, род занятий, социальная принадлежность. Можно ли при построении прогностической программы ограничиться учетом только этих личностных характеристик претендента?

Одна из интереснейших попыток ответа на этот вопрос была предпринята студентами Пермского государственного педагогического университета и описана в [24, 25]. В экспериментах использовалась нейронная сеть с пятью входами: x_1 — возраст кандидата в президенты; x_2 — его происхождение (1 — из бедных, 2 — средний класс, 3 — интеллигенция, 4 — из богатых); x_3 — семейное положение (1 — женат, 0 — холост); x_4 — количество детей; x_5 — основная область деятельности (1 — военная, 2 — политическая, 3 — общественная, 4 — экономическая). Сеть имела один выходной нейрон, выходной сигнал которого d мог принимать значе-

ние 1, означающее победу кандидата на выборах, или 0 — поражение.

Множество примеров, взятых из истории президентских выборов США и Франции, было разбито на обучающее (табл. 3.7), использованное для обучения сети, и тестирующее (табл. 3.8), предназначенное для проверки ее прогностических свойств. (Разумеется, примеры тестирующего множества при обучении сети не использовались.)

Таблица 3.7

**Обучающее множество примеров президентских выборов
США и Франции**

Претенденты	Возраст	Происхождение	Семейное положение	Кол-во детей	Деятельность	Победа или поражение
	x_1	x_2	x_3	x_4	x_5	d
А. Гор	52	3	1	3	2	0
Б. Голдуотер	58	4	1	2	2	0
Б. Клинтон	46	2	1	2	2	1
В. Жискара	55	2	1	2	2	0
Д. Вашингтон	57	1	1	0	1	1
Д. Адамс	62	3	1	1	2	1
Д. Буш мл.	54	4	1	2	4	1
Д. Буш ст.	64	4	1	1	4	1
Д. Бьюкенен	66	4	0	0	2	1
Д. Доул	73	2	0	0	1	0
Д. К. Адамс	58	3	1	2	2	1
Д. Керри	61	4	1	2	2	0
Д. Медисон	58	4	1	0	2	1
Д. Монро	49	1	1	2	1	1
Д. Тейлер	51	3	1	15	2	1

Претенденты	Возраст	Происхождение	Семейное положение	Кол-во детей	Деятельность	Победа или поражение
	x_1	x_2	x_3	x_4	x_5	d
Д. Форд	61	2	1	2	1	1
Ж. М. Ле Пен	74	1	1	3	3	0
Ж. Ширак	63	2	1	2	2	1
Л. Жоспен	58	2	1	0	2	0
Р. Рейган	70	2	1	2	2	1
Т. Джеферсон	58	4	1	1	2	1
У. Мондейл	56	2	1	3	2	0
Ф. Миттеран	75	2	0	1	3	1
Ф. Пирс	49	4	1	0	2	1
Ш. де Голль	68	3	1	3	1	1

Таблица 3.8

Тестирующее множество примеров президентских выборов

Претенденты	Возраст	Происхождение	Семейное положение	Кол-во детей	Деятельность	Победа или поражение	Прогноз сети
	x_1	x_2	x_3	x_4	x_5	d	y
Д. Кеннеди	44	4	1	2	2	1	0,82
А. Линкольн	52	1	1	4	1	1	0,93
М. Дукакис	56	2	1	2	2	0	0,00
Д. Макговерн	59	2	1	0	2	0	0,11
Д. Эйзенхауэр	63	2	1	2	1	1	1,04

В табл. 3.8 в предпоследнем столбце *d* помещены желаемые выходы нейросети, т. е. факты, которые реально имели место в истории США и Франции, а в последнем столбце — прогноз сети *y*, обученной на множестве примеров предыдущей таблицы. Как видно из табл. 3.8, нейросеть хотя и с погрешностью, но совершенно правильно «предсказала» победу Д. Кеннеди, А. Линкольна и Д. Эйзенхауэра и поражение М. Дукакиса и Д. Макговерна. Это значит, что она смогла извлечь из весьма скромного обучающего множества некую скрытую закономерность и применить ее при тестовом прогнозировании.

Невероятно, но факт! Получается, что рейтинг претендентов в президенты стран практически не зависит от политической конъюнктуры, лозунгов, обещаний, политических платформ, что результат голосований в первую очередь определяется неким удачным сочетанием его субъективных личностных характеристик, которое выявила нейронная сеть!

Студентами Пермского госпедуниверситета, изучающими искусственный интеллект, была введена и еще одна малообоснованная гипотеза. Они приняли допущение, что менталитет россиян не отличается от менталитета французов и американцев, и что на выборах президента своей страны россияне будут отдавать голоса, пользуясь аналогичными критериями.

Собрав необходимые личностные данные о возможных российских претендентах (табл. 3.9) и применив к ним обученную на французском и американском опыте нейросеть, пермские студенты выполнили прогноз президентских выборов в России на 2008 год (см. последний столбец табл. 3.9).

На рис. 3.23 эти результаты представлены в виде гистограммы. Если полученные в результате прогноза цифры рассматривать как сравнительный рейтинг претендентов, то можно заключить, что наибольшие шансы стать президентом России нейронная сеть определила для Д. Медведева, а наименьшие — для Ю. Трутнева. Рейтинги же осталь-

Таблица 3.9

Личностные характеристики возможных претендентов на пост президента России и прогноз результатов выборов в 2008 г.

Претенденты	Возраст	Происхождение	Семейное положение	Кол-во детей	Деятельность	Прогноз сети
	x_1	x_2	x_3	x_4	x_5	y
С. Иванов	55	2	1	2	1	0,91
Д. Медведев	42	2	1	1	3	1,59
Б. Грызлов	58	2	1	2	4	1,01
Г. Зюганов	64	1	1	2	2	1,04
В. Жириновский	62	1	1	3	2	0,79
Д. Рогозин	45	3	1	1	2	1,14
Ю. Трутнев	52	2	1	2	2	0,13

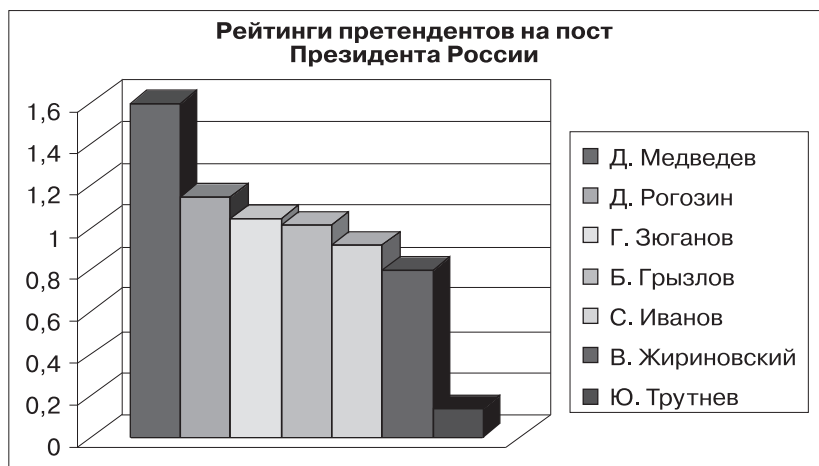


Рис. 3.23. Прогноз президентских выборов в России в 2008 г., выполненный студентами Пермского госпедуниверситета в начале 2007 г.

ных претендентов, занимая промежуточное положение, незначительно различаются между собой.

Еще раз обратим внимание, что данный прогноз был выполнен и опубликован в начале 2007г., т. е. более чем за год до выборов Президента России в 2008 г. Политическая обстановка в то время складывалась так, что фамилию кандидата в президенты России Дмитрия Медведева вообще мало кто слышал. Так что данные этого нейросетевого прогнозирования хотя и были опубликованы, но с комментариями, что «к этому прогнозу не стоит относиться серьезно», потому что «прогноз выполнен в предположении, что избирательские менталитеты французов, американцев и россиян совпадают, что, по-видимому, неверно. Такой рейтинг был бы у претендентов на пост президента не в России, а в Америке или во Франции». Теперь же, после того как Д. Медведев стал президентом России, мы видим, что авторы книг [24, 25] напрасно сомневались в результатах своего прогнозирования. Их нейронная сеть оказалась права! Прогноз результатов президентских выборов, выполненный студентами Пермского госпедуниверситета, полностью подтвердился, что можно считать серьезным успехом метода нейросетевого математического моделирования.

Отметим, что применение методов нейросетевого математического моделирования позволяет не только предсказывать будущие события, но и исследовать их, ставить над моделями виртуальные эксперименты и в итоге активно влиять на них, пытаясь изменять будущее в соответствии со своими желаниями и интересами. Варьируя входные параметры обученной нейронной сети и производя вычисления, можно получать ответы на многие вопросы, интересующие политтехнологов. Например, варьируя возраст политического деятеля и производя вычисления с помощью нейросети, можно проследить, как его рейтинг будет меняться со временем, обнаружить пики его популярности среди избирателей и выбрать момент, когда ему лучше всего уйти в отставку.

Так, продолжая свои научные исследования, пермские студенты расширили количество входных параметров и обу-

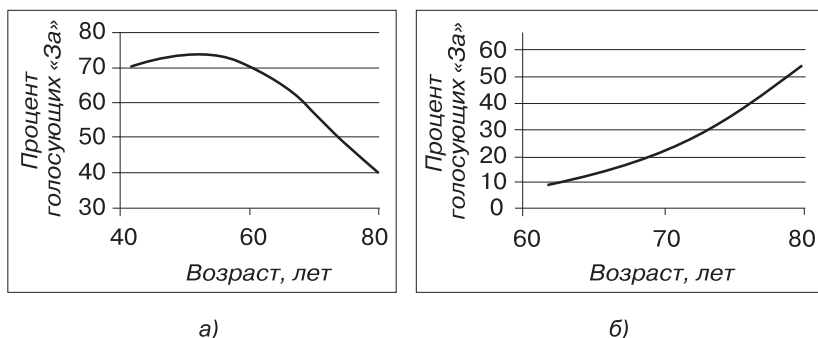


Рис. 3.24. Зависимость рейтинга Д. Медведева (а) и В. Жириновского (б) от их возраста

чили свою нейронную сеть как на зарубежном, так и на российском избирательном опыте. Анализируя полученные с помощью нейронной сети кривые, изображенные на рис. 3.24, они заключили, что пик популярности президента Д. Медведева придется на 50–55 лет, после чего он начнет резко падать. Рейтинг же В. Жириновского будет постоянно повышаться и к 80 годам может перевалить за 50%.

Проводя целенаправленные исследования нейросетевой модели, можно разработать рекомендации, как тому или иному политическому деятелю можно изменить свой рейтинг. На рис. 3.25 в виде гистограммы приведены результаты моделирования различных возможных изменений в личной и политической жизни известного политического деятеля, кандидата в президенты России на выборах 2004 г. Ирины Хакамады. Как видно на этой гистограмме, повторное участие в выборах увеличит количество голосующих за ее кандидатуру до 8,4% (против исходных 3,5%), а вот третья попытка стать президентом страны, наоборот, приведет к снижению количества голосов до 1,7%. Смена партии на «правящую» приведет к падению ее шансов до нуля. Если Ирина Хакамада сменит свою деятельность с экономической на спорт или культуру, то ее рейтинг увеличится до 8,5% голосов. Если она изберет военную карьеру, то это принесет ей 17,5% голосов. А если она станет преемником действующего президента страны, то за нее бу-

дут голосовать 67,4% избирателей. Если она разведется, то ее шансы победить на выборах упадут до нуля. Если она сменит пол, то ее рейтинг повысится до 5,5% голосов. Если И. Хакамада заведет еще одного ребенка, то ее рейтинг возрастет до 9,6% голосов; если она заведет двоих детей, то рейтинг повысится до 16,5%, а если троих — то до 24,1%. Если же И. Хакамаде удастся радикальным образом повысить свою популярность, то она имеет шансы получить 47,8% голосов российских избирателей.

Таким образом, применение нейросетевых технологий позволяет не только выявлять и анализировать закономерности такого сложного политического явления, как прези-

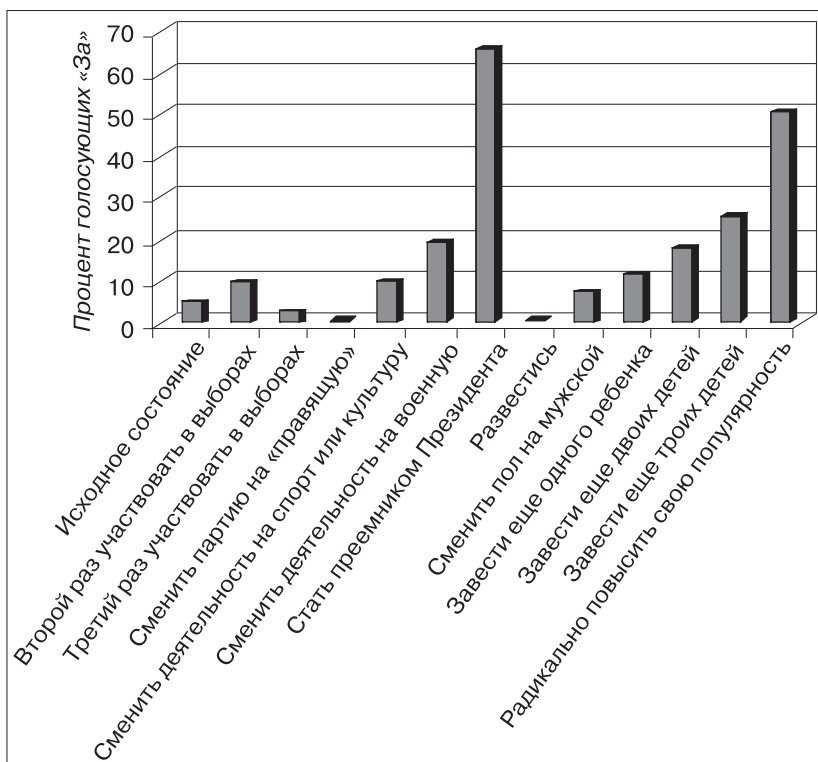


Рис. 3.25. Результаты моделирования влияния возможных изменений в личной и политической жизни И. Хакамады на ее рейтинг

дентские выборы, но и оказывать влияние на ход и результаты этого процесса.

В заключение отметим, что не доверять результатам прогнозирования, представленным на рис. 3.24 и 3.25, нет оснований, поскольку они получены путем проведения виртуальных экспериментов на математических моделях, адекватность которых установлена пермскими студентами путем проведения тестовых испытаний этих моделей. Отметим также, что кроме прогнозирования президентских выборов, нейросетевые математические модели могут быть использованы для прогнозирования исходов любых видов голосований, например выборов в законодательные собрания областей и краев, в академии РАН и др. Они могут быть использованы для выявления перспективных кандидатов, а также для разработки рекомендаций по влиянию на рейтинги участников выборов.



Коротко о главном

Применение методов искусственного интеллекта позволяет выявлять и анализировать закономерности такого сложного политического явления, как президентские выборы.

Нейросетевые математические модели предназначены не только для пассивного предсказания будущих событий — побед или поражений. Исследуя эти модели и выявляя нужные закономерности, можно понять, как надо оказывать активное влияние на эти будущие события, поворачивая их развитие в нужном направлении, создавая благоприятную ситуацию для достижения своих целей.



Вопросы и задания

1. Какая информация использовалась для создания обучающего множества примеров при построении программы, прогнозирующей результаты президентских выборов в России?

2. Для чего необходимо тестирующее множество примеров и чем оно отличается от обучающего множества?
3. Попробуйте создать нейропрограмму, прогнозирующую результаты президентских выборов в России, используя в качестве обучающих примеров отечественный опыт.
4. Как можно использовать созданную таким образом нейросеть для разработки рекомендаций политическим деятелям по повышению их рейтинга?

3.2.9. Невербальность, интуиция и живучесть нейрокомпьютеров

Нейронная сеть — это система, выполненная по образу и подобию человеческого мозга. Вполне естественно, что она «наследует» его свойства — как положительные, так и отрицательные.

Так же, как и человеческий мозг, нейронная сеть способна получать знания из данных, обнаруживать скрытые в них закономерности. Но, как и человек, нейронная сеть не способна объяснить, как именно она это делает.

Архимед открыл свой закон, лежа в ванне; Ньютон — наблюдая за падением яблока. Оба ученых гениальным образом *догадались*, нашли истину, не объяснив, однако, никому, как они это сделали.

Нейронная сеть, извлекая закономерности из данных, тоже способна получать знания, делать догадки, открывать законы природы. Но так же, как и человек, нейронная сеть не способна к четкой формулировке пунктов алгоритма, позволившего сделать то или иное умозаключение. Она не способна *вербализовать* (от французского *verbal* — «словесный») свои выводы, т. е. объяснить их словами. Знания, которыми она обладает, *невербальны*, т. е. необъяснимы словами.

Согласно некоторым современным эзотерическим теориям, гений в момент озарения «считывает информацию из параллельных миров», поэтому он не может выстроить

логическую цепочку, не может объяснить, как он догадался. Разумеется, нелепо было бы утверждать, что нейронная сеть в момент извлечения знаний продельывает аналогичную операцию, поэтому моделирование деятельности мозга дает нам основания возразить против эзотерической теории познания. По-видимому, многие из феноменов, которые демонстрируют экстрасенсы, контактеры или шаманы, можно объяснить свойством мозга извлекать информацию из неполных данных — свойством, которое удастся моделировать с помощью нейронных сетей.

В практике применения нейронных сетей известны случаи, когда нейронные сети демонстрируют феномен, называемый в жизни «шестым чувством». *«Шестое чувство»*, или *интуиция* — это способность человека делать правильные выводы и принимать правильные решения, руководствуясь информацией, которой для логического объяснения таких выводов и решений недостаточно. Обычно считается, что «шестым чувством» в большей степени обладает женская половина человечества: очень часто женщины совершают поступки и принимают решения, которые мужчины (с точки зрения их традиционной логики) объяснить не могут, — и нередко такие поступки и решения оказываются единственно правильными.

Нейронные сети с успехом извлекают знания при анализе информации, из которой, казалось бы, эти знания извлечь невозможно. Ранее в разделе 3.2.3 был приведен пример диагностики неисправностей авиационных двигателей по совокупности их полетных параметров. Феномен заключается в том, что кроме всех прочих неисправностей нейронная сеть диагностирует и такие, которые традиционными диагностическими методами, основанными на законах физики, выявить невозможно. Например, дефект «стружка в масле» считается побочным и обнаруживается только после вскрытия авиадвигателя на испытательном стенде. Этот дефект не выявляется традиционными диагностическими методами, поскольку его наличие, по мнению специалистов, никак не влияет ни на один из снимаемых с авиадвигателя полетных параметров. Тем не менее, несмотря

на отсутствие какой-либо логической связи между этим дефектом и параметрами работы авиадвигателя, нейронная сеть все-таки обнаруживает скрытую от обычных вербальных методов диагностики закономерность и ставит правильный диагноз относительно наличия или отсутствия стружки.

Как можно объяснить этот феномен?

Возможное объяснение состоит в следующем. Действительно, согласно общепринятому мнению специалистов, появление небольшого количества стружки в поддоне двигателя не сказывается на его температуре, вибрациях, шумах, перепадах давления и пр., а следовательно, не вызывает серьезных отклонений ни одного из значений параметров входного вектора $\mathbf{X}$: $x_1, x_2, \dots$. Но возможно, что небольшие, не заметные для специалистов изменения в параметрах работы двигателя все-таки есть. Причем это изменения не одного и не двух, а сразу многих компонентов вектора $\mathbf{X}$, и нейросеть реагирует на эти изменения. Ведь она «следит» не за одним и не за двумя, а сразу за всеми компонентами вектора $\mathbf{X}$!

Феномены подобного рода в практике применения нейросетевых технологий не редки. Особенно часто они наблюдаются при исследовании достаточно сложных объектов, когда нейронные сети выявляют связи и закономерности, о существовании которых специалисты, создавшие объект, не знают сами. Выявляются даже такие взаимосвязи, которые, при поверхностном рассмотрении, противоречат здравому смыслу специалистов и становятся понятными и объяснимыми только после более тщательного изучения объекта.

Таким образом, невербальность и интуиция нейронных сетей — это свойства, «унаследованные» ими от естественного прототипа — мозга. Причем если первое из этих свойств является вредным, то второе — полезным: если невербальность вызывает трудности в применении нейронных сетей и осложняет нахождение общего языка с потребителями программной продукции, которым часто требуется не только получить какой-либо вывод, но и объяснить,

на основании чего он сделан, то свойство интуиции, наоборот, делает нейросети одним из эффективнейших инструментов, полезных для научного и практического применения.

Интересно отметить, что кроме вышеназванных, обнаружилось и еще одно полезное свойство нейронных сетей, тоже «унаследованное» ими от мозга. Это — *свойство живучести*. Известно, что мозг очень живуч: были случаи, когда люди, потеряв в результате травм до 30% объема мозга, вели себя вполне адекватно и продолжали совершать обдуманные поступки. Как выяснилось, нейрокомпьютеры (в отличие от обычных фон-неймановских компьютеров) тоже продолжают функционировать при довольно значительных повреждениях! Они продолжают принимать правильные решения в случае нарушения одного или нескольких внутренних соединений — синаптических связей и продолжают решать поставленные перед ними задачи даже в случае удаления одного или нескольких нейронов.



Коротко о главном

Персептрон, являясь моделью человеческого мозга, «наследует» его свойства — не только положительные, но и отрицательные.

К положительным относятся свойства:

- обучения;
- обобщения;
- интуиции;
- живучести.

Отрицательным качеством нейросетей является их невербальность.



Вопросы и задания

1. Перечислите положительные и отрицательные свойства, которые персептрон «унаследовал» от своего прототипа — человеческого мозга.

2. Какие знания называют невербальными?
3. Вспомните, как вербализуются выводы, получаемые с помощью экспертных систем.
4. Подумайте над тем, как можно вербализовать нейросетевые знания.
5. Сформулируйте определение интуиции.
6. В каких случаях может оказаться полезным свойство живучести нейрокомпьютеров?

3.2.10. Круг задач, решаемых при помощи нейросетей

Список примеров практического применения нейросетевых технологий можно многократно увеличить. Однако и немногие приведенные выше примеры убедительно показывают, что нейросети — это универсальный и весьма эффективный инструмент для построения математических моделей самых разнообразных физических, технических, химических, экономических, социальных и других объектов, процессов и явлений. Исследуя такие модели, мы можем решать разнообразные практические задачи. Например, если нам удалось построить математическую модель какого-то сложного технологического процесса (скажем, выплавки стали в электродуговой печи, или крекинга нефти в химическом реакторе, или производства электроэнергии на электростанции), то исследуя эту компьютерную математическую модель, проводя над ней виртуальные эксперименты, изучая с их помощью влияние входных параметров на выходные, можно решать *задачу оптимизации* моделируемого технологического процесса. Это значит, что можно подобрать оптимальное сочетание входных параметров, обеспечивающих, например, максимально высокое качество выплавляемой стали; можно рассчитать наиболее благоприятный ход химической реакции крекинга нефти; можно выбрать наиболее эффективный режим работы электростанции и т. д.

Аналогично решаются и задачи оптимизации в сфере бизнеса и экономики. В этом случае выходом нейронной

сети может быть некая *целевая функция*, имеющая смысл экономической эффективности предприятия (валового продукта, прибыли или рентабельности фирмы).

Если математическая модель является *нестационарной*, т. е. составлена с учетом фактора времени, то ее можно использовать для решения *задач прогнозирования*. Это значит, что с помощью такой математической модели можно узнать, какими будут технологические, экономические, социальные, политические и другие показатели моделируемого объекта или процесса в будущем и как на них можно повлиять, принимая те или иные меры сегодня.

Если математическая модель работает в режиме реального времени, т. е. оперативно получает сведения о текущих изменениях параметров моделируемого объекта; если результаты математического моделирования могут быть оперативно переданы оператору, управляющему объектом, или непосредственно введены в приборы, дозирующие подачу руды, кислорода и других химических компонентов в электропечь либо в приборы, управляющие параметрами работы химического или ядерного реактора, то такая математическая модель будет решать *задачу управления* моделируемым объектом или процессом.

Наконец, если нейросеть каким-либо образом вербализовать, то ее можно использовать для *понимания* явлений и процессов, происходящих в изучаемой предметной области.

Кроме перечисленных задач — понимания, оптимизации, прогнозирования и управления, как мы уже видели, персептрон может также решать задачи *распознавания* и *классификации образов*, причем под образами могут пониматься зрительные изображения, символы, тексты, запахи, звуки, шумы, наборы параметров (векторы), характеризующие состояние здоровья пациента или состояние технического устройства, и пр.

Отметим, что во всех ранее рассмотренных примерах построения нейросетевых математических моделей не требовалось знание и использование законов природы, экономики, общества. Вместо этого нужно было лишь подготовить обучающее множество примеров, содержащее

статистические данные о предметной области. И если это множество оказывается достаточно *репрезентативным* (представительным), то нейросеть *сама* извлекает закономерности, необходимые для формирования математической модели, адекватной рассматриваемой предметной области.



Коротко о главном

С помощью нейросетевых технологий можно решать задачи оптимизации, прогнозирования, управления, понимания моделируемых объектов, процессов, явлений, а также задачи распознавания и классификации образов.



Вопросы и задания

1. В чем состоит задача оптимизации моделируемого объекта или процесса и как она решается с помощью нейросетевых технологий? Приведите примеры.
2. В чем состоит задача прогнозирования моделируемого объекта, явления или процесса и как она решается с помощью нейросетевых технологий? Приведите примеры.
3. В чем состоит задача управления моделируемым объектом или процессом и как она решается с помощью нейросетевых технологий? Приведите примеры.
4. В чем состоит задача распознавания (классификации) образов и как она решается с помощью нейросетевых технологий? Приведите примеры.
5. Чем отличается искусственная нейронная сеть от нейрокомпьютера?

3.3. Проблемы проектирования и применения персептронов

3.3.1. Теорема существования

Итак, мы уже убедились в том, что:

- 1) в жизни встречается множество практически важных задач, решить которые можно методом математического моделирования, т. е. путем построения некоторой сложной функции, осуществляющей преобразование вектора входных параметров X в вектор выходных параметров Y ;
- 2) универсальным инструментом построения такой функции являются нейросетевые технологии.

Однако при этом возникает вопрос: всегда ли можно построить нейронную сеть, выполняющую преобразование, заданное любым множеством обучающих примеров, и каким требованиям должна удовлетворять эта нейронная сеть?

Надо признать, что до сих пор материал, который мы изучали, носил позитивный, «рекламный» характер. Мы много говорили о преимуществах и широких возможностях метода нейросетевого моделирования, но почти ничего — о его недостатках. Мы не упоминали о трудностях, связанных с применением нового метода, но освоить любой метод можно, только познакомившись со всеми его проблемами, изучив способы их преодоления, изучив и поняв его теоретическую базу.

С одной из таких трудностей, называемой «Проблемой исключающего ИЛИ», мы уже сталкивались при выполнении **лабораторной работы № 1** и теперь знаем, что персептрон должен иметь скрытый слой нейронов, но остались другие вопросы:

- 1) всегда ли можно спроектировать и обучить многослойный персептрон, обеспечивающий решение любой задачи?

- 2) каким лучше задавать количество внутренних нейронных слоев и количество нейронов в них? Может быть, как в мозге, — 10^{11} нейронов? Может, чем их будет больше, тем лучше?

Ответы на эти вопросы мы выясним, познакомившись с *теоретической базой нейронных сетей*.

Важнейшее место в теории нейронных сетей занимает *теорема Арнольда — Колмогорова — Хехт-Нильсена*, доказательство которой достаточно сложно и поэтому в нашем курсе не рассматривается. Тем не менее сама история этой теоремы представляется весьма интересной и поучительной.

Как мы уже не раз отмечали, с физической точки зрения перцептрон — это устройство, моделирующее человеческий мозг на структурном уровне. Однако, анализируя формулы, по которым он преобразует сигналы, можно заключить, что с математической точки зрения перцептрон — это всего лишь *аппроксиматор*, заменяющий функцию многих аргументов суммой функций, каждая из которых зависит только от одного аргумента. Вопрос о том, всегда ли можно любую функцию многих аргументов представить в виде суммы функций меньшего количества аргументов, интересовал математиков на протяжении нескольких столетий.

Так, в 1900 г. на Всемирном математическом конгрессе в Париже знаменитый немецкий математик Давид Гильберт сформулировал 23 проблемы, которые он предложил решать математикам наступающего XX в. Одна из этих проблем (под номером 13) как раз и декларировала невозможность такого представления. Таким образом, приговор новой области искусственного интеллекта был вынесен еще за полвека до ее появления. Получалось, что перцептрон, сколько бы нейронов он ни имел, не всегда мог построить нужную функцию.

Многие проблемы Гильберта были успешно решены математиками XX в. Его утверждения-гипотезы подтверждались одна за другой, но вот с упомянутой тринадцатой возникли проблемы.

Сомнения относительно возможностей персептронов развеяли советские математики — академики А. Н. Колмогоров и В. И. Арнольд. Им удалось доказать, что любая непрерывная функция n аргументов $f(x_1, x_2, \dots, x_n)$ всегда может быть представлена в виде суммы непрерывных функций одного аргумента: $f_1(x_1) + f_2(x_2) + \dots + f_n(x_n)$. Тем самым гипотеза Гильберта была опровергнута, а нейроинформатике был открыт «зеленый свет».

В 1987–1991 гг. профессор Калифорнийского университета (США) Р. Хехт-Нильсен переработал теоремы Арнольда — Колмогорова применительно к нейронным сетям. Он доказал, что *для любого множества пар, различающихся между собой входными и выходными векторами произвольной размерности, существует двухслойный персептрон с сигмоидными активационными функциями и с конечным числом нейронов, который для каждого входного вектора $\mathbf{X}_q$ формирует соответствующий ему выходной вектор $\mathbf{D}_q$* . Тем самым была доказана принципиальная возможность построения нейронной сети, выполняющей преобразование, заданное любым множеством различающихся между собой обучающих примеров, и установлено, что такой универсальной нейронной сетью является двухслойный персептрон — персептрон с одним скрытым слоем, причем активационные функции его нейронов должны быть сигмоидными.

Теорема Арнольда — Колмогорова — Хехт-Нильсена имеет очень важное для практики следствие в виде формулы, с помощью которой можно определять необходимое количество синаптических весов нейронной сети:

$$\frac{N_y Q}{1 + \log_2(Q)} \leq N_w \leq N_y \left(\frac{Q}{N_x} + 1 \right) (N_x + N_y + 1) + N_y, \quad (3.23)$$

где N_y — количество нейронов выходного слоя, Q — количество элементов множества обучающих примеров, N_w — необходимое число синаптических связей, N_x — количество нейронов входного слоя.

Оценив с помощью этой формулы необходимое число синаптических связей N_w , можно рассчитать и необходимое количество нейронов в скрытых слоях. Например, количество нейронов скрытого слоя двухслойного персептрона будет равно:

$$N = \frac{N_w}{N_x + N_y}. \quad (3.24)$$

Последняя формула становится очевидной, если ее левую и правую части умножить на $(N_x + N_y)$ и нарисовать схему двухслойного персептрона (т. е. персептрона с одним скрытым слоем).

А теперь давайте зададимся вопросом: что было бы, если гипотеза 13-й проблемы Гильберта оказалась бы верной? Очевидно, тогда у нейроинформатики не было бы теоретического фундамента. Ее прикладные возможности были бы серьезно ограничены; более того — сама правомерность ее существования как области науки оказалась бы под вопросом! Кому нужен нейрокомпьютер, который в принципе не может решать ряд задач, для решения которых он предназначен? Не было бы у нейроинформатики и ее многочисленных практических приложений...

Но они есть! В настоящее время нейроинформатику можно заносить в Книгу рекордов Гиннеса: она побила все рекорды по количеству приложений в самых разнообразных областях! И сама жизнь блестяще подтвердила теоремы и доказательства, выполненные в 1956–1957 гг. нашими соотечественниками — Андреем Николаевичем Колмогоровым и Владимиром Игоревичем Арнольдом!

3.3.2. Проектирование персептронов

Как следует из теорем Арнольда — Колмогорова — Хейт-Нильсена, для построения нейросетевой модели любого сколь угодно сложного объекта достаточно использовать персептрон с одним скрытым слоем сигмоидных нейронов, количество которых определяется формулами (3.23), (3.24).

Однако в практических реализациях персептронов количество как слоев, так и нейронов в каждом из них нередко отличаются от теоретических. Дело в том, что иногда бывает целесообразно использовать персептроны с большим количеством скрытых слоев.

Строгой теории выбора оптимального количества скрытых слоев и нейронов в скрытых слоях пока не существует. На практике чаще всего используют персептроны, имеющие один или два скрытых слоя, причем количество нейронов в скрытых слоях обычно колеблется от $N_x/2$ до $3N_x$.

При проектировании персептрона необходимо понимать, что он должен не только правильно реагировать на примеры, на которых он обучен, но и уметь *обобщать* приобретенные знания, т. е. правильно реагировать на примеры, которых в обучающем множестве не было. Чтобы оценить способность сети к обобщению, кроме *обучающего множества* примеров L (рис. 3.26), в рассмотрение вводят некоторое количество тестовых примеров, образующих *тестирующее множество* T , где примеры тестирующего множества относятся к той же самой предметной области, но в процессе обучения не участвуют.

После обучения вычисляют среднеквадратичную (или максимальную) ошибку нейронной сети как разницу между прогнозным вектором Y и желаемым вектором D . При этом ошибку, вычисленную на примерах обучающего множества L , называют *ошибкой обучения* и обозначают как ε_L , а ошибку, вычисленную на примерах тестирующего множества T , называют *ошибкой обобщения*, или *ошибкой тестирования* и обозначают как ε_T .

Обратим внимание, что ошибка обучения ε_L вычисляется на тех примерах предметной области, на которых сеть обучалась. Ошибка же обобщения вычисляется на примерах той же предметной области, которые сеть никогда «не видела». Поэтому ошибки ε_L и ε_T имеют разную природу и, соответственно, разный характер поведения: при увеличении количества N нейронов скрытых слоев персептрона ошибка обучения ε_L обычно уменьшается, тогда как ошибка обобщения ε_T сначала уменьшается, а затем (начиная

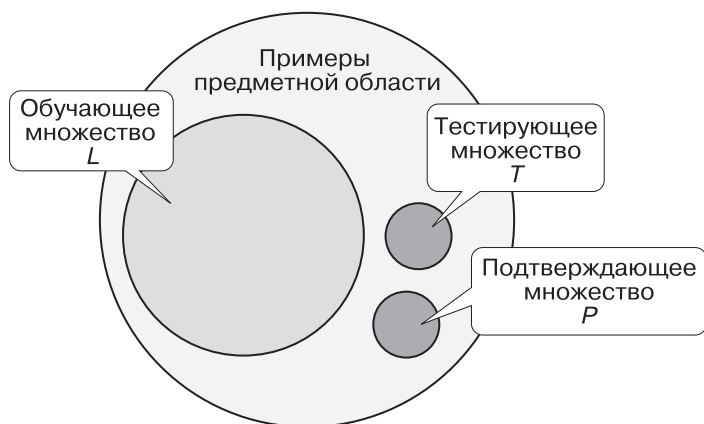


Рис. 3.26. Деление примеров предметной области на обучающее множество L , тестирующее множество T и подтверждающее множество P

с некоторого оптимального количества нейронов скрытого слоя $N = N_0$) вновь возрастает. Характерные кривые зависимостей ошибок обучения и обобщения от количества нейронов скрытых слоев персептрона приведены на рис. 3.27. Свойство же нейронных сетей терять способность к обобщению при чрезмерном увеличении числа нейронов скрытых слоев называют *переобучением*, или *гиперразмерностью*.

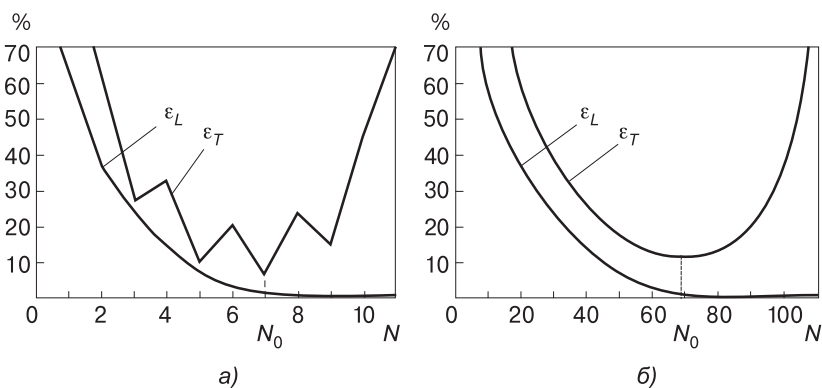


Рис. 3.27. Характерные зависимости ошибки обучения ε_L и ошибки обобщения (тестирования) ε_T от количества нейронов скрытых слоев персептрона N

Вооружившись этими теоретическими знаниями, приведем некоторые практические рекомендации по проектированию персептронов — по выбору количества входных и выходных нейронов, скрытых слоев, нейронов в скрытых слоях, а также вида активационных функций.

1. Количество входных нейронов персептрона должно совпадать с количеством входных параметров, т. е. с размерностью вектора $\mathbf{X}$, который определен условиями решаемой задачи.
2. Количество нейронов выходного слоя должно совпадать с количеством выходных параметров, т. е. с размерностью выходного вектора $\mathbf{D}$, что также определено условиями задачи.
3. Количество скрытых слоев персептрона, согласно теоремам Арнольда — Колмогорова — Хехт-Нильсена, должно быть не менее 1, причем нейроны в скрытых слоях должны иметь сигмоидную активационную функцию.
4. Количество нейронов в скрытых слоях можно приблизительно оценить по формулам (3.23), (3.24), однако его желательно уточнять для каждой конкретной задачи путем построения кривых, аналогичных показанным на рис. 3.27, и определения с их помощью оптимального количества нейронов скрытых слоев N_0 .

В наиболее ответственных случаях, наряду с обучающим множеством $\mathbf{L}$ и тестирующим множеством $\mathbf{T}$, в рассмотрение вводят еще одно множество примеров, называемое *подтверждающим*, и обозначают его буквой $\mathbf{P}$ (см. рис. 3.27). Примеры этого множества также относятся к рассматриваемой предметной области, но ни в обучении, ни в тестировании сети не участвуют, поэтому *ошибка прогнозирования* ε_P , вычисленная на примерах подтверждающего множества $\mathbf{P}$, является наиболее объективной оценкой качества нейросетевой математической модели.



Для проектирования нейронных сетей и работы с ними имеется множество специальных программ, называемых *нейропакетами*. С одной из таких программ вы познакомитесь при выполнении **лабораторной работы № 8** из состава лабораторного практикума, размещенного на сайте <http://www.LbAI.ru>.



Коротко о главном

Количество входных и выходных нейронов персептрона определяется условиями решаемой задачи. Количество скрытых слоев должно быть не менее 1. Количество нейронов в скрытых слоях можно приблизительно оценить по формуле Арнольда — Колмогорова — Хехт-Нильсена (3.23), (3.24), однако его рекомендуется каждый раз уточнять, добиваясь наилучших обобщающих свойств персептрона, путем построения кривых, аналогичных показанным на рис. 3.27, и определения с помощью этих кривых оптимального количества нейронов скрытых слоев. (При построении кривых, аналогичных рис. 3.27, используются примеры тестирующего множества.)



Вопросы и задания

1. В чем заключается процесс проектирования персептронов?
2. Как определяется количество нейронов входного слоя персептрона?
3. Как определяется количество нейронов выходного слоя персептрона?
4. Как определяется количество нейронов в скрытых слоях персептрона?
5. От каких параметров задачи зависит оптимальное количество нейронов скрытых слоев персептрона?
6. Как задается вид активационных функций нейронов?

7. Чем ошибка обучения отличается от ошибки обобщения?
8. Нарисуйте примерные графики зависимости обеих указанных ошибок от количества нейронов скрытых слоев персептрона.
9. К какому нежелательному последствию может привести чрезмерное уменьшение количества нейронов в скрытых слоях персептрона?
10. К какому нежелательному последствию может привести чрезмерное увеличение количества нейронов в скрытых слоях персептрона?
11. Как на практике выбирается количество нейронов скрытых слоев персептрона?
12. Чем подтверждающее множество примеров отличается от тестирующего множества?
13. Почему ошибка, вычисленная на примерах подтверждающего множества, является наиболее объективной оценкой качества нейронной сети?

3.3.3. Алгоритм применения метода нейросетевого математического моделирования

Завершая изучение метода математического моделирования на основе персептронов, в качестве итога приведем примерный алгоритм его применения (рис. 3.28).

Этап 1. Постановка задачи

На этом этапе определяются цели моделирования, устанавливаются входные и выходные параметры модели, выбирается структура (состав и длина) входного вектора X и выходного вектора D .

В качестве компонентов входного вектора X важно выбрать *значимые* параметры — те, которые оказывают существенное влияние на результат. Если же имеются сомнения в значимости того или иного входного параметра, то его лучше включить во входной вектор, рассчитывая, что позже с помощью создаваемой нейросети можно будет оценить степень влияния этого параметра на результат и, если

это влияние окажется слабым, то в последующей работе исключить этот параметр.

Выходной вектор D формируется так, чтобы его компоненты давали возможность получить ответы на все поставленные в задаче вопросы.

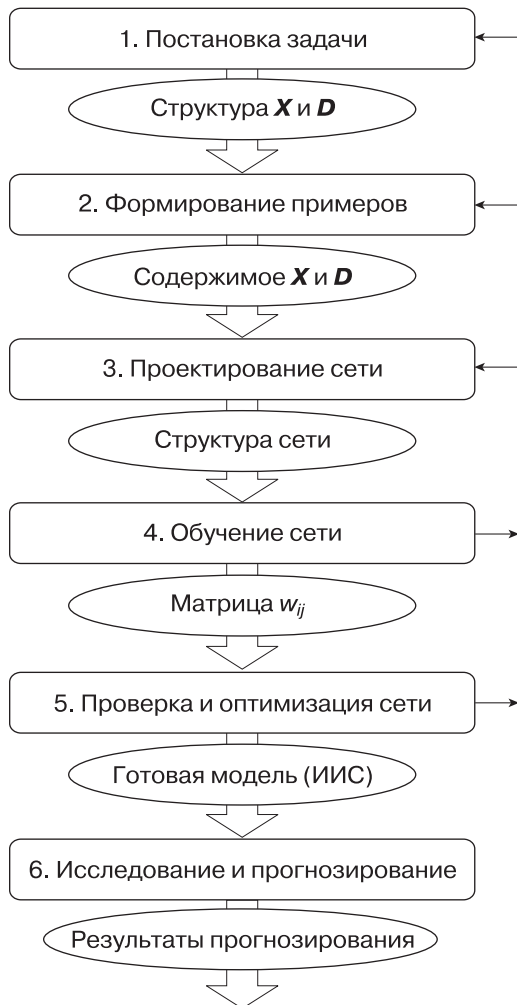


Рис. 3.28. Типовой алгоритм применения метода нейросетевого математического моделирования

Компоненты входного вектора X и выходного вектора D представляют собой числа. Это могут быть значения каких-либо величин (например, температуры тела, артериального давления, частоты пульса и др.) либо числа, кодирующие наличие или отсутствие каких-либо признаков (например, 1 — мужской пол, 2 — женский пол). В некоторых случаях, если данные нечеткие и есть сомнение в их правильности, то полезно кодировать оценку их вероятности. Например, если у врача есть сомнения в правильности выставляемого диагноза, то он может закодировать не сам диагноз, а его вероятность или степень развития болезни, применяя десяти- или стобалльную систему оценки.

Этап 2. Формирование примеров

На этом этапе формируется содержимое входных и выходных векторов. В результате создается множество пар $X_q - D_q$ ($q = 1, \dots, Q$), где каждая такая пара составляет пример, характеризующий предметную область.

Значения компонентов векторов X_q и D_q могут быть сформированы различными способами: получены путем проведения социологических опросов, анкетирования, специальных экспериментов над предметной областью, взяты из архивных материалов организаций, из средств массовой информации, из сети Интернет и других источников.

Все множество примеров разбивают на обучающее L и тестирующее T (см. рис. 3.26). Обычно объем тестирующего множества выбирают не менее 10% от обучающего. Какой именно необходим минимальный объем обучающего множества, зависит от задачи. Обычно рекомендуется не менее 50 примеров, но в практике применения нейросетевых технологий встречались и случаи, когда для решения задачи хватало 10 обучающих примеров.

В особо ответственных случаях рекомендуется, кроме обучающего множества L и тестирующего множества T , также формировать подтверждающее множество P из примеров, принадлежащих той же самой предметной области, но не входящих ни в множество L , ни в множество T (см. рис. 3.26).

Этап 3. Проектирование сети

Структура персептрона выбирается исходя из следующих соображений:

- 1) количество входных нейронов N_x должно быть равно размерности входного вектора $\mathbf{X}$;
- 2) количество выходных нейронов N_y должно быть равно размерности выходного вектора $\mathbf{D}$;
- 3) количество скрытых слоев, согласно теореме Арнольда — Колмогорова — Хехт-Нильсена (см. раздел 3.3.1), должно быть не менее 1; на последующих этапах количество скрытых слоев может корректироваться, если это позволит улучшить качество работы сети;
- 4) количество нейронов в скрытых слоях рассчитывается с помощью формул (3.23), (3.24); на последующих этапах количество нейронов в скрытых слоях также может корректироваться, если это позволит улучшить качество работы сети;
- 5) активационные функции скрытых нейронов, согласно теореме Арнольда — Колмогорова — Хехт-Нильсена, рекомендуется задать сигмоидными, однако в дальнейшем их вид может быть изменен, если это позволит улучшить качество работы сети.

При корректировке структуры персептрона также следует иметь в виду, что увеличение количества скрытых нейронов обычно позволяет добиться меньшей ошибки обучения, но чрезмерное его увеличение приводит к эффекту гиперразмерности — к потере обобщающих свойств сети, выражающейся в возрастании ошибки обобщения.

Этап 4. Обучение сети

Обучение сети — очень важный, но не окончательный этап создания нейросетевой математической модели. Цель обучения — подобрать синаптические веса w_{ij} так, чтобы на каждый входной вектор $\mathbf{X}_q$ множества обучающих примеров сеть выдавала вектор $\mathbf{Y}_q$, минимально отличающийся от заданного выходного вектора $\mathbf{D}_q$. Эта цель достигается путем использования особых *алгоритмов обучения ней-*

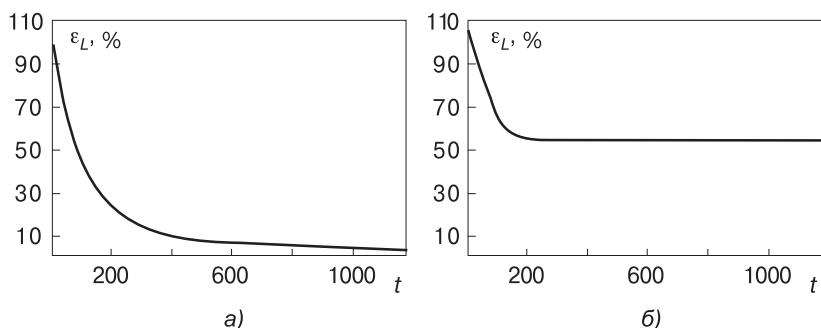


Рис. 3.29. Характерные кривые зависимости ошибок обучения от количества эпох обучения в случае, когда сеть обучается успешно (а) и когда процесс обучения не дает желаемого результата (б)

ронной сети. Характерная кривая обучения — зависимость ошибок обучения от количества эпох обучения приведена на рис. 3.29а.

Однако может случиться так, что сеть «не захочет» обучаться: ошибка обучения с увеличением количества эпох не будет стремиться к нулю (см. рис. 3.29б). Причины этого нежелательного явления могут быть следующими.

1. Недостаточное количество скрытых слоев и скрытых нейронов — тогда рекомендуется увеличить их количество.

2. Наличие в обучающем множестве противоречащих друг другу примеров, — например, когда одним и тем же симптомам соответствуют разные диагнозы заболеваний.

Обнаружить такие примеры в обучающем множестве можно, скажем, путем его визуального анализа. Затем надо разобраться в причинах возникновения противоречащих примеров: некоторые из них могут просто оказаться ошибочными, и их нужно удалить.

Возможно и то, что уже в самой структуре входного вектора отсутствуют какие-то параметры (например, возраст больного, рост, вес, цвет его глаз и др.), также оказывающие влияние на диагноз. Тогда рекомендуется вернуться на этап 1 и пересмотреть постановку задачи, увеличить размерность входного вектора X , добавив дополнительные параметры, которые своими значениями обеспечат непротиворечивость примеров обучающего множества.

3. Попадание в *локальный минимум*. Эта проблема связана с тем, что поверхность функции — ошибки персептрона (схематично изображенная на рис. 3.10) — имеет достаточно сложный характер со множеством мелких «ямочек», называемых *локальными минимумами*. Процесс обучения персептрона состоит в движении по этой поверхности небольшими шагами в сторону антиградиента (в сторону наибольшего наклона поверхности функции-ошибки). Естественно, что опускаясь таким образом по поверхности функции-ошибки, можно «застрять» в каком-либо мелком локальном минимуме, не достигнув самого глубокого минимума, называемого *глобальным*. В подобном случае рекомендуется попробовать заново начать процесс обучения из другой начальной точки.

4. Слишком большая скорость обучения — тогда рекомендуется уменьшить ее.

Этап 5. Проверка и оптимизация сети

Проверка обобщающих свойств сети (иногда данный этап называют *тестированием сети*) производится на тестирующем множестве примеров, т. е. на тех примерах, которые не были использованы при обучении сети. Результаты тестирования полезно представить графически в виде гистограммы, на которой значения желаемых выходов персептрона (D_q) сопоставлены с действительными (прогноznыми) (Y_q) — теми, которые вычислил персептрон. Пример такой гистограммы приведен на рис. 3.30.

Если разница между компонентами желаемого выходного вектора тестирующего множества примеров D_q и прогноznыми значениями Y_q окажется незначительной, то можно переходить к следующему, этапу 6, не выполняя *оптимизацию* сети. Однако чтобы лишний раз убедиться в адекватности разрабатываемой нейросетевой математической модели, полезно вернуться на этап 2 и включить те примеры, которые были тестирующими (либо часть тестирующих примеров), в обучающее множество, а часть примеров, бывших обучающими, сделать тестирующими, и затем снова повторить этапы 3, 4 и 5.

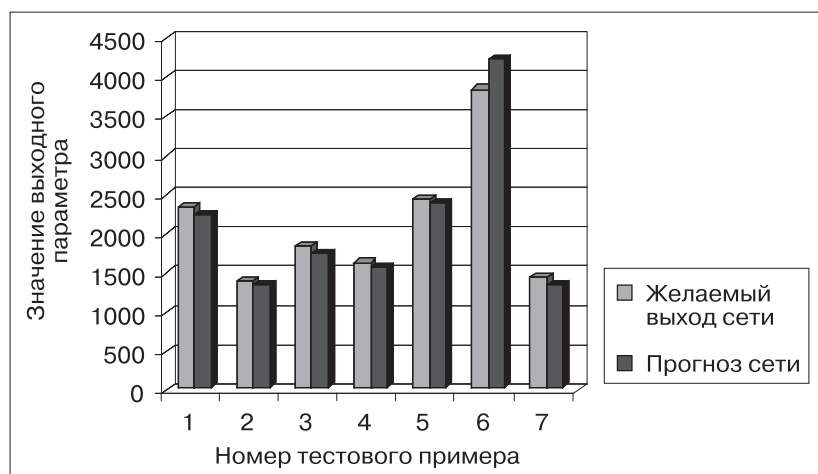


Рис. 3.30. Пример гистограммы, показывающей соотношение желаемых выходов сети с прогнозными

Если же погрешность обобщения сети окажется неприемлемо большой, то надо попытаться *оптимизировать сеть*. Эта операция состоит в подборе наиболее подходящей для данной задачи структуры сети — количества скрытых слоев, скрытых нейронов, синаптических связей, а также вида и параметров активационных функций нейронов. В некоторых нейропакетах предусмотрена автоматическая оптимизация сети, но иногда бывает полезно выполнить такую оптимизацию вручную, построив график зависимости погрешности обобщения ε_L от числа скрытых нейронов (см. рис. 3.27) и от других параметров персептрона и выбрав с помощью этих графиков структуру сети, обеспечивающую минимальную погрешность обобщения.

При выборе оптимальной структуры сети следует помнить, что цель оптимизации сети состоит в минимизации погрешности обобщения ε_T , а не погрешности обучения ε_L ! Именно по величине погрешности обобщения судят о качестве сети, о ее обобщающих, а следовательно — прогностических свойствах. Погрешность же обучения — это всего лишь промежуточный результат. Желательно, чтобы она была небольшой, но добиваться ее минимального значения

вовсе не обязательно и даже вредно, поскольку, как видно из графиков на рис. 3.27, это приводит к эффекту переобучения, т. е. к росту погрешности обобщения.

Понятно, что оптимизация нейронной сети подразумевает многократные возвраты назад на этапы 4, 3, 2 или даже 1.

После оптимизации сети ее обобщающие свойства рекомендуется проверить на примерах подтверждающего множества P . Дело в том, что в процессе оптимизации сеть могла «приспособиться» к примерам тестирующего множества, и если эти примеры по каким-либо причинам не характерны для всей предметной области, то на других примерах, которых не было ни в тестирующем, ни в обучающем множествах, нейросеть может дать неожиданно большую ошибку прогнозирования. Для исключения такого явления, а также чтобы окончательно убедиться, что сеть имеет хорошие обобщающие свойства не только на тестирующем множестве примеров, вычисляют ошибку прогноза сети ε_p на подтверждающем множестве, т. е. на тех примерах, которые не участвовали ни в обучении, ни в тестировании.

Результатом оптимизации и проверки сети является готовая к использованию нейросетевая математическая модель предметной области — *интеллектуальная информационная система (ИИС)*.

Этап 6. Исследование модели, прогнозирование

Путем проведения вычислительных экспериментов над математической нейросетевой моделью достигаются цели моделирования и находятся ответы на все поставленные вопросы. Например, могут быть решены такие задачи как оптимизация моделируемого объекта, прогнозирование его будущих свойств, выявление закономерностей предметной области и др.

Нейросетевая математическая модель, если она правильно спроектирована и обучена, «впитает в себя» закономерности моделируемой предметной области. Она реагирует на изменение входных параметров и ведет себя точно так же, как вела бы себя сама предметная область. Соответ-

ственно, надо поставить над такой моделью как можно больше экспериментов. Скажем, можно попробовать поменять некоторые из входных параметров и посмотреть, как при этом меняются значения выходного вектора Y . Например, меняя возраст кандидата в президенты, можно наблюдать за изменением его рейтинга, как это было сделано в разделе 3.2.8. Надо постараться извлечь из этих виртуальных экспериментов как можно больше полезной информации.

Часто бывает полезно графически изображать зависимости выходных параметров модели от входных. Например, на рис. 3.24 были построены зависимости рейтингов Д. Медведева и В. Жириновского от их возраста, а на рис. 3.25 — гистограмма, позволяющая давать практические рекомендации по повышению политического рейтинга И. Хакамады. Напомним, что эти интересные результаты и выводы были получены путем исследования нейросетевой математической модели политического процесса — выборов в президенты страны.

Таким образом, можно заметить, что в результате выполнения приведенного на рис. 3.28 алгоритма метода нейросетевого математического моделирования получается два вида продуктов:

- 1) после выполнения этапа 5 создается готовая к использованию интеллектуальная информационная система, являющаяся математической моделью предметной области;
- 2) после выполнения этапа 6 мы получаем результаты исследования математической модели, представленные в виде графиков, номограмм, гистограмм, полезных рекомендаций и выводов.

В заключение укажем, что алгоритм создания интеллектуальной информационной системы, приведенный на рис. 3.28, апробирован при решении многих практических задач и представляется вполне эффективным. Однако в каждом конкретном случае, а также в зависимости от опы-

та, навыков и предпочтений разработчика ИИС возможны отклонения от этого типового алгоритма. Например, если у разработчика ИИС хорошо развита интуиция или наработаны собственные алгоритмы для определения количества нейронов в скрытых слоях, то совсем не обязательным является использование формулы Арнольда — Колмогорова — Хехт-Нильсена.



Коротко о главном

Процесс нейросетевого математического моделирования состоит из нескольких этапов, включающих постановку задачи, формирование множества примеров предметной области, проектирование, обучение, проверку и оптимизацию сети, исследование готовой математической модели и выполнение с ее помощью требуемых прогнозов.

Результатом применения метода нейросетевого моделирования является интеллектуальная информационная система, представляющая собой математическую модель рассматриваемой предметной области, а также новые научные и/или практические знания, полученные путем исследования математической модели и прогнозирования.



Вопросы и задания

1. Перечислите типовые этапы создания нейросетевой математической модели предметной области.
2. Как формируется структура входного вектора X и выходного вектора D ?
3. Перечислите способы формирования содержимого пар векторов $X_q - D_q$.
4. Как производится проектирование сети?
5. В чем состоит цель обучения сети и как она достигается?

6. Назовите четыре причины, по которым сеть может не поддаваться обучению.
7. В чем состоит цель тестирования сети?
8. В чем состоит цель оптимизации сети — в подборе структуры сети, обеспечивающей минимальную ошибку обучения или обобщения?
9. Как выполняется оптимизация сети?
10. В каких случаях используется подтверждающее множество примеров и можно ли без него обойтись?
11. С какими целями проводится исследование нейросетевой математической модели?
12. Что является результатом нейросетевого математического моделирования?

3.4. Заключительные замечания и выводы

Развитие современной цивилизации свидетельствует о постоянном повышении роли компьютерного математического моделирования как в научных исследованиях для получения новых знаний, так и во многих областях практической деятельности человека. Традиционно компьютерные математические модели строятся на основе фундаментальных законов природы, зависимостей и закономерностей, которые могут быть представлены в виде алгебраических формул, алгебраических или дифференциальных уравнений, обычно решаемых с помощью компьютера численными методами. Именно таким способом ведутся многие исследования в физике, астрономии, экологии, науках о человеке, природе и обществе, выполняются долгосрочные прогнозы погоды, предсказываются землетрясения, цунами, стихийные бедствия, а также рассчитываются и проектируются самолеты, автомобили, ракеты, подводные лодки, здания и сооружения, промышленные и военные объекты.

Всегда считалось, что сам процесс открытия фундаментальных законов природы, общества, вселенной — это прерогатива исключительно человека, что построение математических моделей и их компьютерных реализаций — это высокоинтеллектуальная задача, требующая совместных усилий математиков, программистов и специалистов пред-

метных областей, в которых выполняется моделирование. Однако в результате изучения нашего курса мы понимаем, что и этот традиционный для человека вид его интеллектуальной деятельности, связанный с созданием математических моделей, в настоящее время также перекладывается на компьютер. Он теперь может сам открывать фундаментальные законы природы, выявлять связи и закономерности предметных областей и закладывать их в математические компьютерные модели, — причем во многих случаях компьютер справляется с этим лучше, чем сам человек. Компьютер может выявлять и учитывать в создаваемых им моделях новые законы, закономерности и внутренние связи предметной области, еще не известные ученым и специалистам. В результате модели, создаваемые компьютером, учитывают большее количество факторов, законов и закономерностей, влияющих на результат моделирования, что, конечно же, улучшает качество компьютерных моделей.

Следует заметить, что здесь речь идет не об обычных фон-неймановских компьютерах и не об обычных методах составления компьютерных программ, а о низкоуровневой стратегии искусственного интеллекта. Такие интеллектуальные операции может выполнять компьютер, построенный «по образу и подобию» человеческого мозга — нейрокомпьютер. Это также может быть нейронная сеть, имитирующая нейрокомпьютер на обычном компьютере.

Нейрокомпьютер, как и человеческий мозг, состоит из множества соединенных между собой элементарных ячеек — модельных нейронов, принцип действия которых во многом аналогичен принципу действия биологических нейронов, из которых состоит мозг. Как и в мозге, нейроны нейрокомпьютера соединены между собой множеством связей, через которые происходит обмен электрическими сигналами. Как и биологические, нейроны нейрокомпьютера могут переходить в возбужденное состояние, вырабатывая и посылая электрические сигналы другим нейронам, причем переход в возбужденное состояние для каждого нейрона тем вероятнее, чем большее количество сигналов он получает от других нейронов.

Согласно наиболее распространенным в настоящее время нейрофизиологическим представлениям, вся информация, хранящаяся в мозге, закодирована в виде матрицы сил синаптических связей — множества величин электропроводности проводников, соединяющих биологические нейроны, причем эти величины по мере жизнедеятельности мозга постоянно меняются, чем обеспечивается накопление и корректировка хранящейся в мозге информации. Этот механизм в нейрокомпьютере имитируется путем соответствующего изменения синаптических весовых коэффициентов, выполняющих роль электропроводности соединяющих нейроны проводников. Таким образом, в нейрокомпьютере, как и в мозге, вся информация представляется и хранится в виде матрицы сил синаптических связей нейронов.

В отличие от обычного компьютера, нейрокомпьютер не программируется, а подобно человеку обучается; при этом происходит корректировка сил синаптических связей. Обучение проводится на обучающих примерах, содержащих информацию о моделируемой предметной области. Скажем, чтобы нейрокомпьютер «открыл» теорему Пифагора, надо нарисовать несколько различных прямоугольных треугольников, измерить длины их гипотенуз и катетов и передать эту информацию нейрокомпьютеру. Если измерения выполнены правильно и треугольников было достаточно много, то можно не сомневаться, что нейрокомпьютер повторит подвиг знаменитого геометра древности! Поступая аналогичным образом и собирая информацию о соответствующих предметных областях, можно при помощи нейрокомпьютера заново открыть закон Архимеда, законы Ньютона, все фундаментальные законы физики, химии, биологии, экономики и всех других наук.

Таким образом, нейрокомпьютерные и нейросетевые технологии, по существу, открывают принципиально новый подход к самой методике построения компьютерных математических моделей. Появилась возможность строить математические модели, которые, исходя из одного только эмпирического опыта — множества обучающих примеров, сами извлекают закономерности предметной области и по-

зволяют эффективно использовать их для решения широкого круга практических задач. При этом человеку — создателю нейрокомпьютера — не обязательно задумываться над законами физики, химии, биологии, экономики и т. д., нейрокомпьютер обнаруживает их сам.

С появлением нейрокомпьютерных технологий у нас появился новый способ получения научных знаний, новый инструмент извлечения знаний из данных, позволяющий выявлять ранее не известные, еще не исследованные зависимости и закономерности и активно использовать их для решения практических задач.

Работы многих исследователей, а также наш собственный опыт убеждают нас в том, что нейроинформационные технологии являются универсальным и весьма эффективным инструментом для построения компьютерных математических моделей самых разнообразных физических, технических, химических, экономических, социальных и других объектов, процессов и явлений. Работая с этими моделями, исследуя их, мы можем решать широкий круг разнообразных практических задач: выполнять распознавание образов, оптимизировать моделируемые объекты и процессы, прогнозировать ход развития явлений и процессов и управлять ими.

Можно сказать, что человек, освоивший нейросетевые технологии, поднимается на качественно новый уровень своего развития. Фактически у него появляется дар предвидения: он может предсказывать будущие события, знает, как повлиять на эти события и что нужно сделать, чтобы эти события развивались в нужном для него направлении. Раньше таких людей называли волшебниками и колдунами, поэтому можно почти без преувеличения сказать, что изучаемый вами элективный курс учит вас искусству колдовства. Ведь, изучив этот предмет, вы сможете свободно пользоваться основными нейросетевыми технологиями и даже применять их для достижения своих личных целей. Как и каким образом? Это зависит от вашей собственной фантазии и от того, насколько глубоко вы поняли идеи и освоили методы искусственного интеллекта.

ГЛАВА 4

ИНТЕЛЛЕКТУАЛЬНЫЕ ИГРЫ¹

4.1. Понятие игры и дерева возможностей

Интеллектуальные игры — это одна из областей искусственного интеллекта, в которой оптимистические прогнозы ученых 1950-х гг. хотя и с большим опозданием, но полностью сбылись, когда в 1998 г. в Нью-Йорке в матче «Деер Blue против Гарри Каспарова» компьютер впервые победил чемпиона мира по шахматам. Этот матч состоял из шести партий и завершился со счетом 3,5 : 2,5 в пользу компьютера.

Кроме того, что интеллектуальные игры имеют эмоционально-развлекательное и философское значение, они представляют еще и практический интерес для развития самой теории искусственного интеллекта. Дело в том, что в современных программах-игроках наиболее полно удалось реализовать центральные идеи искусственного интеллекта — обучение, самообучение и самоорганизацию компьютерных программ. Да и само понятие «игра» в искусственном интеллекте имеет более широкое значение. Здесь под игрой понимается некоторая конфликтная ситуация, участники которой своими действиями не только достигают своих личных целей, но и влияют на возможности достижения целей другими участниками игры. Ясно, что при таком толковании «игрой» можно считать многие экономические, политические, военные и другие конфликты.

¹ Эта и последующие главы предлагаются заинтересованным школьникам для самостоятельного изучения.

Проблемой создания игровых программ (в частности, шахматных) занимались многие ученые-кибернетики, в том числе А. Тьюринг, К. Шеннон и Н. Нильсон. Принципы работы, предложенные каждым из разработчиков, опираются на исследования *дерева возможных продолжений игры*.

Корневая вершина такого дерева возможностей представляет собой текущее положение фигур на шахматной доске, а работа программы состоит в выборе очередного хода.

В середине партии у игрока обычно имеется около 30 возможных вариантов следующего хода. Возникающие в результате их перебора конфигурации представляются как дочерние вершины для данной корневой вершины. При этом в каждой из дочерних вершин возможно около 30 ответов противника, так что для изображения результирующих конфигураций требуется еще около 900 вершин, и т. д. Из-за этого дерево быстро разрастается (рис. 4.1), что приводит к «комбинаторному взрыву».

Все вершины дерева возможностей могут быть двух типов. В одних из них очередной ход предстоит делать компьютеру, в других — его противнику; первые называют *альфа-вершинами*, а вторые — *бета-вершинами*. Таким образом,

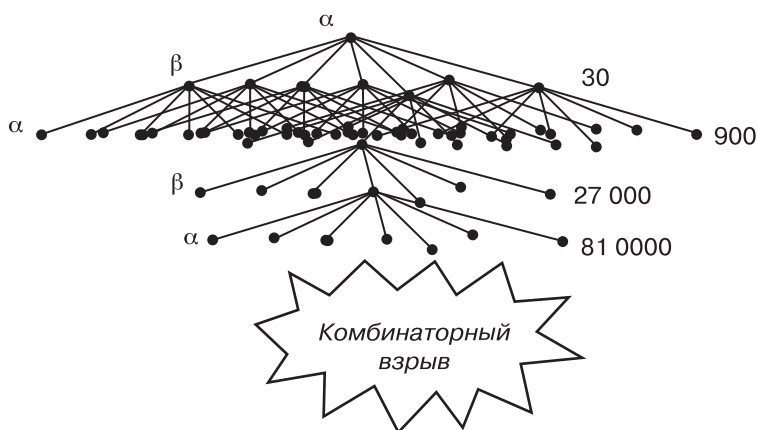


Рис. 4.1. Дерево возможных продолжений шахматной игры

дерево возможностей представляет собой *чередующиеся* слои альфа- и бета-вершин.

Если бы дерево можно было обследовать полностью, вплоть до *листьев*, представляющих собой все возможные окончания в данной игре, то имелась бы возможность выбрать ход, обеспечивающий для компьютера выигрыш независимо от действий противника. Такая возможность имеется в простейших играх, таких как крестики-нолики, калах и пр. В сложных же интеллектуальных играх типа шахмат удастся построить и проследить лишь небольшую часть дерева возможностей. В этом случае говорят, что дерево возможностей подвергается *подрезке*, а конечные вершины, ниже которых дерево отсечено, называют *терминальными вершинами*.

4.2. Методы подрезки дерева возможностей

В программах-«шахматистах» для каждой вершины обычно определяются числовые оценки силы позиций для каждого из партнеров. Такую оценку называют *оценивающим полиномом*, или *оценивающей функцией*:

$$S = k_1 a_1 + k_2 a_2 + k_3 a_3 + \dots, \quad (4.1)$$

где $k_1, k_2, k_3, \dots$ — весовые коэффициенты, а $a_1, a_2, a_3, \dots$ — некие признаки силы позиции.

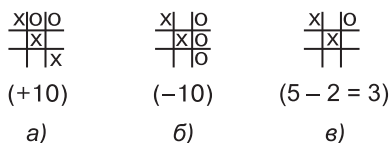
Обычно оценивающая функция равна нулю, когда позиции партнеров равноценны, положительна, когда преимущество — за компьютером, и отрицательна, когда преимущество — за его противником.

Важным компонентом любой оценивающей функции является материальное соотношение, или перевес фигур (в формуле (4.1) это параметр a_1). При этом каждой фигуре придается определенное значение ценности. Вторым важным компонентом оценивающей функции (a_2) должна быть некая мера подвижности фигур, или развитость позиции. Критерием подвижности может быть количество до-

пустимых ходов у игрока. Далее идет оценка контроля центра шахматной доски (a_3) и т. д.

После того как произведена оценка каждой терминальной вершины (конечной вершины при заданной глубине обследования дерева возможностей), выполняется перенос результатов этих оценок вверх по дереву (в направлении корня дерева). Метод, которым это достигается, называется *минимаксным переходом* и заключается в следующем: для альфа-вершин принимается значение, равное наибольшему значению из найденных оценок для дочерних вершин. Такое решение абсолютно оправданно, поскольку, опираясь на такие оценки, компьютер делает правильный для себя ход. И наоборот, для бета-вершин принимается наименьшее из значений оценки для дочерних вершин, поскольку можно предполагать, что противник сделает ход, наименее выгодный для компьютера. В итоге некоторое оценочное значение будет приписано и корневой вершине. Поскольку она являлась альфа-вершиной, это значение будет наибольшим среди значений для дочерних вершин. Ход же, который каждый раз выбирает компьютер, должен преобразовывать существующую на шахматной доске конфигурацию, представленную корневой вершиной, в конфигурацию, представленную той дочерней вершиной, из которой было взято значение оценки для данной корневой вершины.

Аналогичный подход может быть применен в программировании автоматов и для других игр. Так, Н. Нильсоном была предложена оценивающая функция для игры в крестики-нолики. Пусть компьютер ставит крестик, а его противник — нолик. Если конфигурация, которую предстоит оценить, содержит три крестика подряд (т. е. компьютер выигрывает), то оценивающая функция принимает наибольшее значение (скажем, $+10$). Если подряд стоят три нолика, то оценивающая функция принимает наименьшее значение (скажем, -10). Для ситуаций же, которые не являются выигрышными ни для одного из игроков, оценивающую функцию Нильсон предложил вычислять как число строк, столбцов и диагоналей из восьми возможных, все еще открытых для компьютера, т. е. не заблокированных

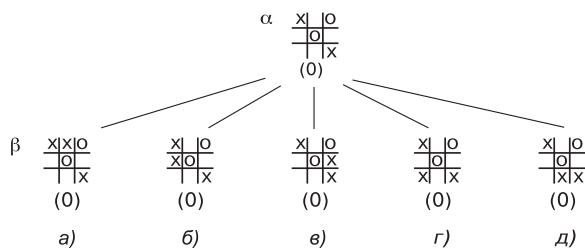
**Рис. 4.2.** Примеры оценивающих функций Нильсона:

a — выиграл компьютер; б — выиграл противник,
в — преимущество за компьютером

ноликами, минус число строк, столбцов и диагоналей, все еще открытых для противника, т. е. не заблокированных крестиками. Примеры расчета оценивающих функций Нильсона для различных ситуаций приведены на рис. 4.2.

Как следует из рис. 4.3, оценивающая функция Нильсона непригодна, если дерево имеет единичную глубину, т. е. состоит только из корневой вершины и одного слоя из ее дочерних вершин. Здесь для всех пяти возможных ходов оценивающая функция оказывается равной нулю, поэтому предлагаемый метод не позволяет выбрать единственно правильный ход (z) — в левый нижний угол. Однако, как видно из рис. 4.4, просмотр вперед на два уровня уже позволяет компьютеру выбрать хороший ход. (Под каждой конфигурацией на этих рисунках указано значение оценивающих функций, вычисленных по методике Нильсона. Аналогичные значения в скобках получены путем применения минимаксной процедуры.)

Деревья, представленные на рис. 4.3 и 4.4, были подрезаны путем их отсечения на определенной глубине. Установлено, что можно добиться более высокого качества игры,

**Рис. 4.3.** Неудачная попытка выбора хода в игре при единичной глубине дерева: компьютер «не видит» единственно правильного хода (г), тогда как а, б, в и д — неправильные ходы компьютера

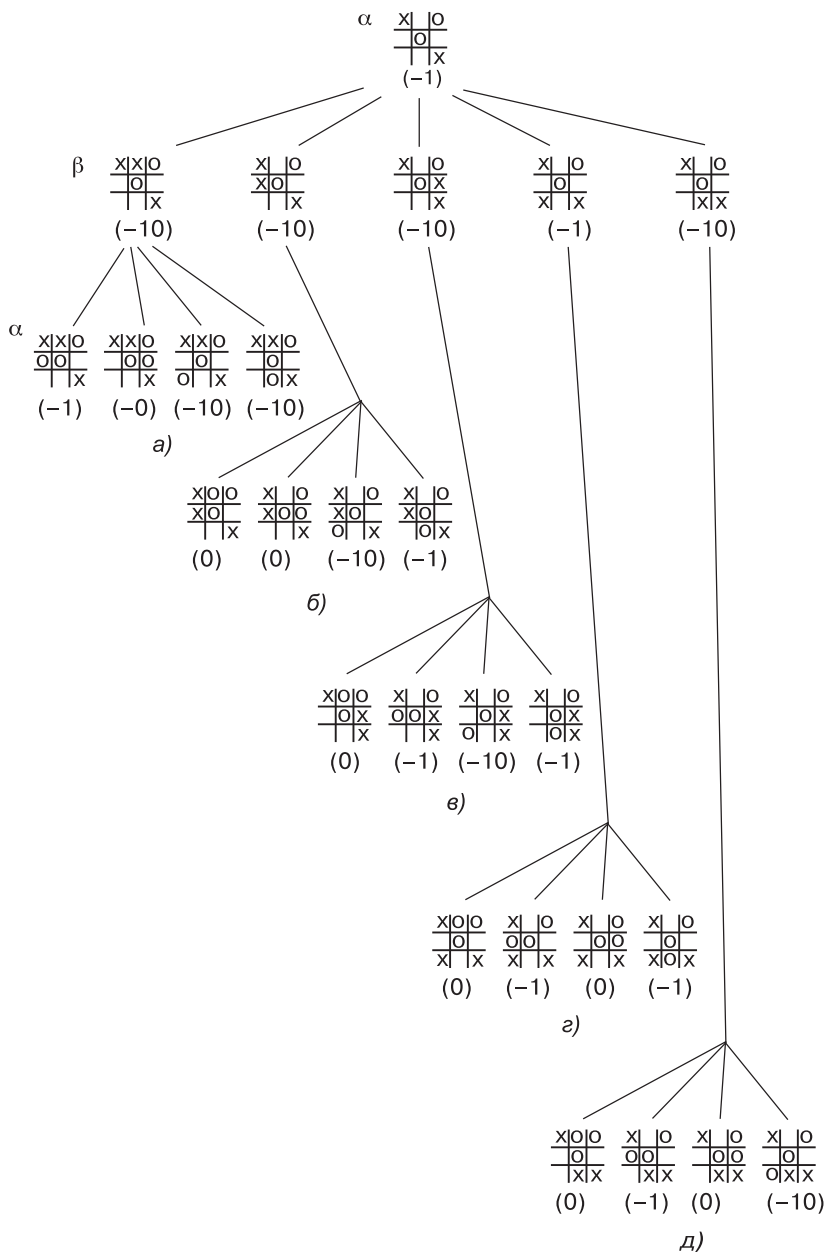


Рис. 4.4. Выбор хода в игре в крестики-нолики при глубине дерева, равной 2; варианты (а), (б), (в), (д) означают поражение компьютера и только вариант (г) может привести к выигрышу

если использовать другие методы отсечения, получившие общее название *методов прямого усечения*. Например, в одном из методов прямого усечения выделяют «мертвые», или «спокойные» позиции, тогда как другие позиции определяют как «живые», или «беспокойные», и вершина с большей вероятностью может быть принята за терминальную (конечную), если представляемая ею конфигурация классифицируется как «мертвая». Разделение же позиций на «живые» и «мертвые» осуществляется с помощью эвристических правил, например, позиция классифицируется как «живая», если существует угроза взятия фигур (для игры в шашки или шахматы).

При использовании прямого усечения дерева нельзя быть абсолютно уверенным, что результат поиска окажется таким же, как в случае, если дерево не подвергалось подрезке. Более надежен *метод обратного усечения*, который также иногда называют *процедурой альфа-бета*. Как мы уже видели, поиск по дереву включает в себя два этапа: построение дерева возможностей с последующим приписыванием его терминальным вершинам числовых значений оценочной функции, а затем — применение минимаксной процедуры для передачи значений вверх по дереву. Процедура же альфа-бета предполагает объединение этих двух этапов так, чтобы значения оценочных функций связывались с вершинами по мере формирования дерева. При этом применение минимаксной процедуры приводит к тому, что оценивающие функции для каждой альфа-вершины с ростом дерева могут только увеличиваться, а для каждой бета-вершины — только уменьшаться. Тогда наблюдение за динамикой изменения оценивающих функций (рис. 4.5) дает возможность в ходе формирования дерева понять, что некоторые из еще не построенных вершин никак не могут повлиять на конечный результат, и отказаться от целых ветвей. Таким образом, обратное усечение состоит в отказе от построения неперспективных вершин, причем распознавание таких вершин осуществляется путем изучения динамики изменения их оценивающих функций.

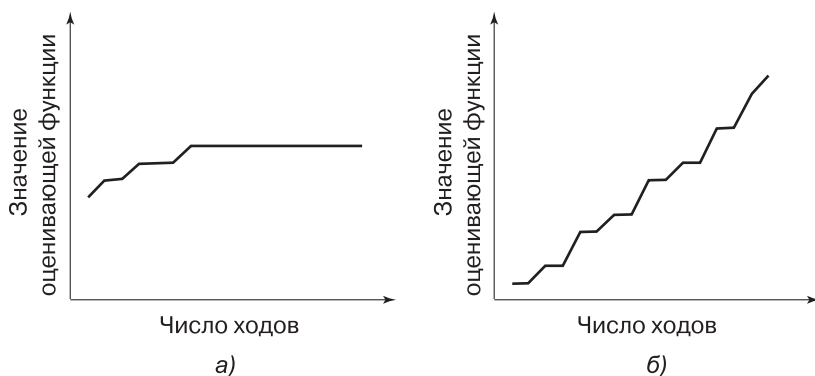


Рис. 4.5. Типичное поведение оценивающей функции в «мертвой» (а) и «живой» (б) альфа-вершине

В настоящее время эффективность шахматных программ постоянно возрастает. Существуют программы, которые легко обыгрывают шахматистов-дилетантов и даже бросают серьезный вызов профессионалам. Тем не менее практически все шахматные программы работают по принципу «грубой силы», который основан на построении и исследовании дерева возможностей. Очевидно, что такой способ выбора хода не соответствует поведению в этом случае человека. Шахматисты обычно предпринимают весьма ограниченный поиск, учитывая лишь небольшое число возможностей. Обсуждая между собой шахматные партии, шахматисты говорят не о поиске по дереву возможностей, а прибегают к таким эмоциональным понятиям, как «атака», «защита», «угроза» или «нападение». А нередко шахматист вообще не рассматривает какие-то варианты, а делает только один единственно правильный ход.

Учитывая все вышесказанное, наряду с увеличением мощности шахматных программ, развивается и новое направление, связанное с их насыщением эвристиками, заимствованными из шахматных партий, т. е. с оснащением шахматных программ базами знаний. Немаловажную роль при этом играет и применение методов обучения и самообучения игровых программ.

4.3. Идеи обучения игровых программ

Для нас представляет интерес компьютерная программа для игры в шашки, разработанная Артуром Сэмюелем. В ней Сэмюелю удалось реализовать две формы обучения: накопление и обобщение.

Накопление сводится к хранению в памяти компьютера большого числа конфигураций на шашечной доске из числа тех, которые реально (а не гипотетически) возникают в ходе шашечных игр. Вместе с каждой такой конфигурацией в памяти хранится также ее числовая оценка, полученная путем построения дерева, применения оценивающей функции к терминальным вершинам и передачи значений вверх по дереву посредством минимаксной процедуры.

Имея в памяти некоторое множество конфигураций вместе с их оценками, программа в процессе работы ищет соответствие между конфигурацией, отвечающей каждой из вершин дерева, и конфигурациями из числа запомненных. Если такое соответствие установлено, то хранимая в памяти оценка передается в эту вершину, и в результате отпадает необходимость строить какую-либо ветвь, которая могла бы возникнуть под этой вершиной. Таким образом, накопление позволяет либо экономить время, либо достичь лучшего качества игры за то же время путем использования несколько большего по размеру дерева.

Конечно, размер списка конфигураций, который может храниться в памяти и использоваться программой, ограничен сверху. Поэтому Сэмюель разработал свою программу так, что наименее употребляемые конфигурации вычеркиваются, а часто встречающиеся — остаются в памяти компьютера.

Обобщение — это другая форма обучения, использованная Сэмюелем. Она позволяет программе в ходе игры улучшать свои оценивающие функции. Обычно оценивающая функция представляет собой полином. В простейшем случае это полином первой степени, или *взвешенная сумма*:

$$S = k_1 a_1 + k_2 a_2 + k_3 a_3 + \dots,$$

где $a_1, a_2, a_3, \dots$ — значения различных вычисляемых критериев, таких как материальное соотношение, подвижность, контроль центра шахматной доски и пр., значимость которых определяется коэффициентами $k_1, k_2, k_3, \dots$. Полином может иметь и более высокую степень относительно переменных a , например:

$$S = k_1 a_1 + k_2 a_2 + k_{11} a_1^2 + k_{12} a_1 a_2 + \dots \quad (4.2)$$

Качество игры зависит от подходящего выбора весовых коэффициентов $k_1, k_2, k_3, \dots$. Обобщение же является средством их подгонки, обеспечивающей улучшение игры. Метод обобщения представляет собой пример оптимизации с использованием процедуры, которая носит название «подъем в гору». Имеется начальный набор значений $k_1, k_2, k_3, \dots$ и в каждый момент времени эти коэффициенты определяют рабочую точку. Эта рабочая точка перемещается в пределах многомерного пространства (рис. 4.6) по мере подгонки величин весовых коэффициентов в поисках положения, в котором оптимизируется определенная реакция или целевая функция.

Чтобы воспользоваться методом «подъема в гору», следовало бы дать программе возможность сыграть некоторое количество игр с определенным партнером, выбрав какое-то начальное множество коэффициентов k_i , а затем

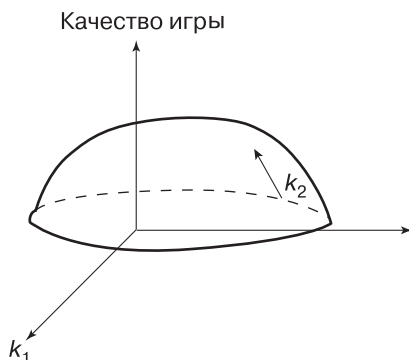


Рис. 4.6. Качество игры как многомерная функция весовых коэффициентов

сыграть еще некоторое количество игр, сделав пробные изменения в положении рабочей точки. Если программа во втором множестве игр выигрывает чаще, то принимается новое значение k_i ; в противном случае происходит возврат к старой величине и производится какое-то новое пробное изменение.

Очевидно, что указанный путь поиска k_i весьма далек от совершенства. Во-первых, этот путь предполагает очень медленное движение. А во-вторых, поскольку партнер не может играть абсолютно ровно, необходимо, чтобы два указанных множества игр были достаточно емкими.

Поэтому А. Сэмюелем был предложен другой путь нахождения весовых коэффициентов во время игры. Его метод основан на том, что качество игры растет с увеличением глубины просмотра дерева возможностей. Если может быть найдено средство для вычисления оценочной функции, обеспечивающее точное совпадение переданного назад по дереву (с большой глубиной) значения оценочной функции с результатом его прямого определения (с небольшой глубиной), то такая оценка должна быть равнозначна изучению всего полностью построенного дерева игры.

Если S — результат прямой оценки с помощью оценочной функции, а S_b — результат передачи оценки по дереву (с большой глубиной), то можно считать их разность ошибкой e , где

$$e = S - S_b. \quad (4.3)$$

Сэмюель сделал так, что в его программе вычислялась корреляция между e и a_1, a_2 и т. д. При этом положительная корреляция между e и любым значением a_i указывала, что соответствующий коэффициент k_i следует уменьшить, а отрицательная корреляция означала, что его надо увеличить.

При применении указанного метода требовалось уделить внимание обеспечению его устойчивости. Для этого Сэмюель фиксировал один из весовых коэффициентов, тогда как другие коэффициенты могли изменяться. Обычно фиксировался наиболее важный параметр, оценивающий

материальное соотношение, поскольку разумно полагать, что игроку всегда выгодно, чтобы его фигуры на доске были сохранены.

Тем самым Сэмюелем был создан алгоритм, обладающий свойством *самообучения* (обучения без учителя). Насколько нам известно, это и была самая первая в мире действующая самообучающаяся программа. Однако Сэмюель пошел еще дальше. Он включил в свою программу большой ассортимент критериев a_1, a_2 и т. д., чем тот, что допускался для использования в конкретной оценивающей функции. При этом используемое множество критериев видоизменялось во время игры: если какое-то из значений весовых множителей k_i оставалось близким к нулю в течение длительного времени, то компонент оценивающей функции, к которому относился этот коэффициент, изымался из рабочего множества, а на его место ставился другой компонент из числа ожидающих своей очереди. Изъятый же критерий добавлялся к множеству ожидающих своей очереди критериев и мог быть впоследствии заново внесен в оценивающую функцию. Такая возможность изменения множества критериев a_i придавала данному методу обучения новый характер. Теперь его можно было воспринимать как некую самоорганизующуюся систему, способную изменять в процессе обучения не только свои параметры, но и структуру.

Следующая идея А. Сэмюеля была воистину гениальной. Он замкнул игровую программу саму на себя — организовал работу программы так, что она могла вести игру и самообучаться непрерывно днем и ночью, имитируя одновременно двух игроков — «икс» и «игрек» (рис. 4.7). При этом игроку «икс» разрешалось модифицировать свою оценивающую функцию путем обобщения, тогда как игрок «игрек» пользовался фиксированной оценивающей функцией. Если «икс» выигрывал, то игрок «игрек» копировал оценивающую функцию у игрока «икс». Если же игрок «игрек» выигрывал подряд три игры, то наоборот, его оценивающая функция копировалась игроку «икс». Последнее гарантировало возможность возвращения игрока «икс» к прежнему положению в том случае, если процесс

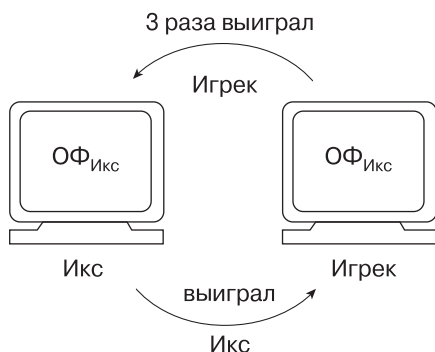


Рис. 4.7. Компьютеры обучают друг друга

подгонки параметров происходил в нежелательном направлении.

Таким образом А. Сэмюель создал программу, которая позволяла не только правильно играть в шашки, но и улучшать стратегию игры, используя опыт, накопленный в предыдущих партиях. Вначале Сэмюель с легкостью обыгрывал машину (IBM 704), но затем она начала быстро совершенствоваться и вскоре достигла такого уровня, что выигрывала у А. Сэмюеля все партии подряд.

По существу это событие, произошедшее в 1959 г., открыло новую эру в истории человечества — эру самонастраивающихся, самообучающихся, самоорганизующихся компьютеров, способных совершенствоваться самостоятельно, без помощи человека.

Укажем теперь на недостаток современных обучающихся игровых программ. В них, как правило, реализованы сразу две парадигмы обучения: с учителем и без него. Естественно, что результат обучения таких программ зависит от конкретного учителя. И очень часто вместо того чтобы учиться играть, такие программы учатся обыгрывать учителя. Яркий тому пример мы видели несколько лет назад, когда после победы DeepBlue над Гарри Каспаровым программисты IBM отказались от матчей с другими гроссмейстерами. В результате Каспаров заявил, что программа была «натаскана» именно на его партиях, изучила его

стиль и потому просто не способна конкурировать с другими гроссмейстерами.

Следует заметить в ответ, что современная шахматная программа заведомо может научиться выигрывать у любого шахматиста, но после этого ей придется некоторое время перестраиваться на другого соперника. Однако, если вдуматься, такой же недостаток имеет любой начинающий шахматист — он учится и в процессе обучения перенимает тактику и стратегию игры своего учителя.

Остается открытым лишь вопрос о способности игровых программ к творчеству. Смогут ли они вырабатывать принципиально новые решения, стратегии, стили, манеры или это навсегда останется прерогативой человека? Впрочем, компьютерное творчество — это уже тема следующей главы нашего курса.



Вопросы и задания

1. Что представляет собой дерево возможностей?
2. Какие вершины дерева возможностей называют «терминальными»?
3. По каким принципам осуществляется подрезка дерева возможностей?
4. По каким принципам формируется оценивающая функция?
5. Объясните основную идею минимаксного перехода.
6. Назовите способы улучшения оценивающих функций.
7. В чем заключаются идеи А. Сэмюеля по реализации самообучения и самоорганизации игровых программ?
8. Как происходит обучение современных шахматных программ?
9. Почему победа компьютера над чемпионом мира по шахматам 1998 г. была поставлена под сомнение? Согласны ли вы с этим?

ГЛАВА 5

МОДЕЛИРОВАНИЕ ТВОРЧЕСТВА

5.1. Философские аспекты творчества

Любое произведение искусства может быть закодировано в виде конечного количества чисел. Например, каждое слово поэмы состоит из букв, которые могут быть закодированы 33 числами, тогда вся поэма будет представлять собой одну длинную строку чисел. Аналогично обстоит дело и в живописи. Полотно картины можно разделить на мельчайшие клетки и цвет каждой клетки закодировать числами. Точно так же в музыке все музыкальное произведение от первой его ноты до последней может быть представлено одной-единственной кривой на экране осциллографа, которую можно с любой степенью точности закодировать числами.

Таким образом, любое произведение искусства любого жанра можно представить в виде набора конечного количества чисел. Количество возможных комбинаций этих чисел огромно, но не бесконечно, поэтому можно представить себе библиотеку, содержащую *все* возможные комбинации таких чисел. Подавляющее большинство их комбинаций в переводе в буквы, цвета и звуки не имеет никакого смысла, но среди этих комбинаций будут и такие, которые имеют смысл и являются тем, что мы называем произведениями искусства. Существуют ли алгоритмы, которые позволят компьютеру выбрать из множества бессмысленных вариаций те, которые являются гениальными поэмами, картинами или симфониями?

Первые попытки создания таких алгоритмов относятся к XVII в. Известен, например, пятисотстраничный трактат

немецкого иезуита Афанасиуса Кирхера «Универсальная музургия, или Великое искусство созвучий и диссонансов». А. Кирхер был учеником Р. Луллия и рассматривал музыкальную композицию как комбинаторную задачу. Его идеи были реализованы в виде устройства, напоминающего механическую экспертную систему Луллия (см. раздел 1.1), которое ныне хранится в кембриджском музее. А в начале XVIII в. вопросами механического сочинения музыкальных произведений с помощью таблиц и игральные кости занимались многие известные композиторы, такие как Бах, Гайдн и Моцарт.

Рассмотрим вопрос создания произведений искусства с использованием уже известных нам современных методов искусственного интеллекта. Мы уже сталкивались с методами математического моделирования в различных естественных науках, таких как физика, метеорология, экономика и пр. Роль математического моделирования в жизни современной цивилизации переоценить трудно, причем круг проблем, решаемых этим методом, постоянно растет. Напомним, что модель — это «черный ящик», в который вводятся входные и выводятся выходные параметры. При этом модель является намеренно упрощенной схемой некоторого реального объекта, системы или процесса, но на основе исследования модели получают рекомендации для решения реальных проблем.

Математическая модель может существовать в виде логических программ, переводимых на язык ЭВМ. Математическую модель, введенную в компьютер, называют компьютерной моделью.

Существуют общие принципы построения моделей. Приведем некоторые из них.

Для построения модели необходимо:

- 1) выявить релевантные (существенные) факторы — те, которые могут сказываться на результатах решения данной проблемы или на исходе рассматриваемого процесса;
- 2) выбрать факторы, которые могут быть описаны количественно;

- 3) объединить факторы по общим признакам и сократить их перечень, выявив инварианты (о них речь пойдет дальше);
- 4) установить количественные соотношения между выбранными факторами и инвариантами.

Факторы, которые по своей природе не могут быть выражены количественно, так же, как и несущественные факторы, исключаются из рассмотрения.

При математическом моделировании важным этапом является установление *инвариантов системы*. Идея инвариантности состоит в том, что хотя система в целом претерпевает последовательные изменения, некоторые ее свойства сохраняются неизменными. Существование инварианта в любом множестве неизбежно влечет за собой ограничение разнообразия. И поскольку любой закон природы подразумевает существование некоторого инварианта, это означает, что всякий закон природы есть ограничение разнообразия, а так как цель любой науки — поиск законов, то наука занимается поиском ограничений разнообразия.

В математике инвариантом называют функцию от преобразуемой величины, не изменяющую своего значения при изменении этой величины. Например, расстояние между двумя точками:

$$S = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

является инвариантным относительно переноса начала координат и поворота координатных осей на любой угол.

В лингвистике выделяют инварианты стихотворных форм. Например, старинная итальянская форма стиха — сонет — всегда имеет четырнадцать строк. При этом первая его часть состоит из двух катренов (четверостиший), а вторая — из двух терцетов (трехстиший). Стихотворный размер сонета — пятистопный (реже шестистопный) ямб. Форма рифмовки для катренов — две разнозвучные рифмы, а для терцетов — две рифмы, отличающиеся от рифм катре-

нов, причем третья строка первого терцета рифмуется со второй строкой второго терцета и т. д. В любом сонете легко обнаружить сохранение указанных инвариантов.

Аналогичным способом инварианты могут быть обнаружены и в других произведениях искусства. Возможность же создания тех или иных произведений искусства может быть определена (в первом приближении) как сознательная или бессознательная способность находить нужные инварианты и комбинировать их для получения желаемого эффекта. Эта способность проявляется художником-творцом в результате обобщения закономерностей всего предшествующего художественного наследия.

Таким образом, мы видим, что искусство в конечном счете преследует ту же самую цель, что и наука, — это выявление инвариантов, установление связи между ними, ограничение разнообразия.

О возможности моделирования творческой деятельности человека непрерывно идут дискуссии и высказываются различные точки зрения — как положительные, так и отрицательные. Прямое отношение к подобного рода дискуссиям имеет *теорема Мак-Каллока — Питтса* — основателей нейрокибернетики. Как мы уже знаем, этими учеными было введено понятие математического нейрона. Если нейрон является основной рабочей клеткой коры больших полушарий мозга человека, то математический нейрон — это абстрактный логический элемент, в котором формально отражены лишь те свойства живого нейрона, которые связаны с переработкой информации. Принцип действия математического нейрона и его возможности для решения практических задач мы рассматривали при изучении разделов 3.1–3.4. Смысл же теоремы Мак-Каллока — Питтса сводится к тому, что любое функционирование живой нервной ткани, которое можно представить с помощью конечного числа слов, может быть описано при помощи искусственной нейронной сети. Но тогда существует принципиальная возможность создания сети из математических нейронов, способной к творческой деятельности.

Впрочем, теорема Мак-Каллока — Питтса представляет собой *теорему существования*: она начево не говорит о том, как именно нужно создавать сеть из математических нейронов, чтобы воспроизвести творческую деятельность человека, а только утверждает, что такую сеть принципиально можно построить. Практических же успехов в области моделирования творческой деятельности удалось добиться, следуя другим, альтернативным направлением искусственного интеллекта, называемым *высокоуровневой стратегией*, или *кибернетикой «черного ящика»*.

5.2. Музыкальное творчество

Человеческий мозг — это своего рода банк данных и знаний, в котором хранится огромная информация, собранная за всю прожитую человеком жизнь. Доказано, что человеческий мозг никогда и ничего не забывает. Каждый прожитый человеком день до мельчайших подробностей, как на видео пленку, записывается в его память, и эта информация может быть определенным образом извлечена.

Точно так же и в памяти композитора существует множество различных мелодий, накопленных им в течение жизни, а может быть — даже передавшихся ему вместе с генами от его далеких предков. И естественно полагать, что фрагменты этих мелодий, отдельные музыкальные фразы, музыкальные инварианты осознанно или неосознанно используются композитором в его творческом процессе.

Учитывая все это, первое, что нужно сделать при создании модели музыкального творчества (рис. 5.1) — занести в память компьютера как можно больше музыкальных произведений (создать базу данных). Далее, как и в любой интеллектуальной системе, нужно создать базу знаний, состоящую из законов музыкальной гармонии — соотношений между музыкальными инвариантами (сольфеджио). Теперь на вход компьютера надо подать «творческое вдохновение» в виде начальной музыкальной фразы, состоящей, например, из четырех нот, и заставить компьютер отыскать такое же сочетание нот в одном из хранящихся в его памяти музы-

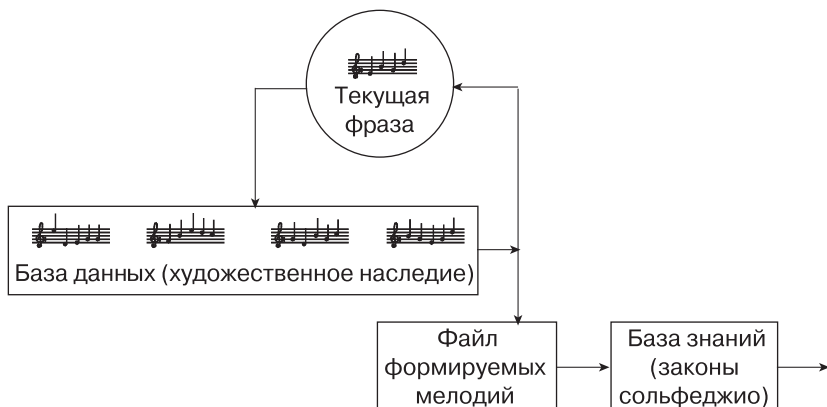


Рис. 5.1. Примерная схема музыкального творчества

кальных произведений. Затем нужно прочесть следующую за найденной фразой ноту и приписать ее к текущей музыкальной фразе, а первую ноту из этой фразы выдать в файл формируемых мелодий и вычеркнуть из текущей фразы так, чтобы в ней по-прежнему оставалось четыре ноты. Далее процесс поиска нот продолжается путем анализа следующих за найденной в памяти компьютера мелодий.

В результате в файле формируемых мелодий сформируется последовательность нот новой мелодии, которая по своему звучанию будет напоминать заложенные в памяти компьютера известные мелодии, но и чем-то отличаться от них. Например, если в память закладывались вальсы, то на выходе тоже получится вальс, если марши, то на выходе будет марш и т. д. А после завершения формирования мелодия автоматически аранжируется согласно законам музыкальной гармонии, хранящимся в базе знаний.

Описанный алгоритм выбора продолжений мелодий из базы данных можно снабдить эвристическими правилами, регулируя их силу с помощью все тех же коэффициентов доверия (см. раздел 2.3.2). Разумеется, что разным музыкальным стилям будут соответствовать разные коэффициенты доверия. Эти коэффициенты можно менять в процессе обучения, добиваясь улучшения качества сочиняемых мелодий.

Изложенная выше методика компьютерного синтеза музыки лишь в упрощенной форме отражает процесс творчества композитора и не претендует на статус полной модели его деятельности. Тем не менее эта модель в упрощенной форме отражает наши представления о творческой деятельности композитора. В ней есть база данных, содержащая художественное наследие, база знаний, состоящая из законов музыкальной гармонии, а также элемент вдохновения в виде начального приближения, которое можно задавать при помощи генератора случайных чисел. Существует также возможность совершенствования компьютерного композитора за счет его обучения — например, путем модификации коэффициентов доверия правил формирования продолжений мелодии.

Более сложные методики, отражающие также и другие стороны музыкального творчества, были предложены в 1955 г. учеными Иллинойского университета Л. Хиллером и Л. Исааксом. Они провели серию экспериментов, в которых в базу знаний последовательно закладывались законы сольфеджио (гармонизация) и вводились разнообразные музыкальные ритмы и темпы, а также был использован датчик случайных чисел.

На рис. 5.2 изображена блок-схема программы, реализованной на машине «Иллиак». На входе этой программы при помощи генератора случайных чисел задавались целые числа, соответствующие закодированным нотным знакам. Затем каждое такое число пропускалось через последовательность из четырех контрольных схем (I–IV), которые пропускали в запоминающее устройство только числа, образующие правильную (подчиняющуюся заложенным в контрольной схеме ограничениям) мелодическую линию. Завершенный период запоминался, а затем выдавался на

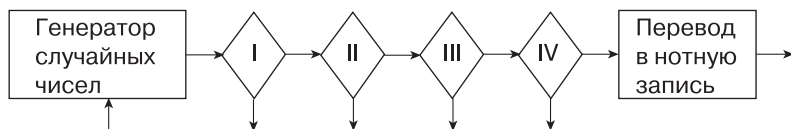


Рис. 5.2. Блок-схема программы компьютерного сочинения музыки

печать и расшифровывался в виде нот. Если же хотя бы одна из схем задерживала хотя бы одну ноту, то управление вновь передавалось генератору случайных чисел, и поиски правильной ноты продолжались. А после 50 неудачных попыток подобрать нужную ноту мелодическая линия разрушалась и начинала выстраиваться новая линия. Согласно описанному алгоритму, за 1 час работы машина «Иллиак» создавала 100 мелодий. Именно так была написана знаменитая «Иллиак-сюита» для струнного оркестра.

Широкую известность в свое время получили и музыкальные произведения, сочиненные ЭВМ «Урал-2» по алгоритмам, разработанным советским математиком Р. Х. Зариповым. В его программах также использовался генератор случайных чисел, но он выдавал не только код ноты, но также ее длительность и интервал между нотами. А далее эти данные проходили контроль на соответствие закономерностям базы знаний — законам музыкальной гармонии, полученным при анализе широкого круга музыкальных произведений. Программы Р. Х. Зарипова моделировали сочинение мелодий в мажоре или миноре и аккомпанемента к ним, а также на заданный стихотворный текст и ритм.

Принципы, разработанные первыми программистами-музыковедами, в настоящее время закладываются в схемы современных оркестровых электромузыкальных инструментов и широко используются композиторами и музыкантами. Однако этот инструментарий является вспомогательным, его применение ограничено написанием гармонии, аранжировок или сопровождений. Сами же темы сочиняет по-прежнему человек. Дело в том, что создать хорошую простую мелодию неизмеримо сложнее, чем оркестровое произведение в авангардистской манере, перегруженное случайными звуко сочетаниями и диссонансами. Когда композитор сочиняет мелодию, которая становится популярной, происходит колоссальный прорыв вперед, так как это — открытие нового, неизвестного ранее соотношения между музыкальными инвариантами.

Современные же компьютерные программы могут сочинять новые мелодии, которые приятно звучат и чем-то

напоминают ранее известные, имеют хороший стиль и манеру, но в них всегда чего-то не хватает, чтобы стать действительно популярными. Знаменитые «Yesterday» и «Лунная соната», да и не только они, сочинены не компьютером! Однако очень вероятно, что компьютер превзойдет человека и в этой области, и как ни обидно это сознавать, возможно, уже в совсем недалеком будущем молодежь будет петь и танцевать под компьютерную музыку, а великие произведения, сочиненные классиками, будут считаться экзотикой...

5.3. Поэтическое творчество

Считается, что задача моделирования стихотворчества несоизмеримо сложнее, чем моделирование сочинения музыкальных произведений. Как показали исследования русского языка, если одна буква делового языка несет 0,6 бита информации, а буква обыкновенной разговорной речи — 1 бит, то одна буква поэтической речи соответствует 1,5 бита информации.

Не вдаваясь в философские и этические проблемы, перейдем к проблеме моделирования поэтического творчества только лишь с помощью уже известных нам инструментальных средств. Например, можно воспользоваться *формализмом Бекуса — Наура*, который главным образом применяется в связи с распознаванием и обработкой текстовой информации, с машинным переводом, а также с созданием систем естественно-языкового общения. В этом формализме применяются следующие символы-операторы:

: : = — «определяется как» или «может быть переписан как»;

| (вертикальная черта) — используется для разделения различных альтернативных возможностей;

< > (угловые скобки) — используются для добавления нетерминальных символов, т. е. символов, которые должны определяться одним из правил. В отличие от них, символы, не заключенные в угловые скобки, считаются терминальными и представляют лишь самих себя.

Например, цифра в формализме Бекуса — Наура определяется следующим образом:

$\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9,$

где $\langle \text{цифра} \rangle$ — это нетерминальный символ, а каждая из цифр 0, 1, ..., 9 — конкретный терминальный символ.

Приведем некоторое множество правил упрощенного описания синтаксической структуры английского предложения:

1. $\langle \text{предложение} \rangle ::= \langle \text{существ. фрагмент} \rangle$
 $\langle \text{глагольн. фрагмент} \rangle \langle \text{существ. фрагмент} \rangle$
2. $\langle \text{существ. фрагмент} \rangle ::=$
 $\langle \text{артикль} \rangle \langle \text{определяемое существ.} \rangle$
3. $\langle \text{определяемое существ.} \rangle ::=$
 $\langle \text{существ.} \rangle \mid \langle \text{определение} \rangle \langle \text{определяемое существ.} \rangle$
4. $\langle \text{глагольн. фрагмент} \rangle ::=$
 $\langle \text{глагол} \rangle \mid \langle \text{глагольн. фрагмент} \rangle \mid \langle \text{фрагмент наречия} \rangle$
5. $\langle \text{фрагмент наречия} \rangle ::=$
 $\langle \text{наречие} \rangle \mid \langle \text{предлог} \rangle \langle \text{существ. фрагмент} \rangle$

К этим правилам добавим небольшой словарь.

6. $\langle \text{артикль} \rangle ::= a \mid the$
7. $\langle \text{существ.} \rangle ::= cat \mid mouse \mid dog \mid fox$
 (кошка, мышь, собака, лиса)
8. $\langle \text{определение} \rangle ::= quick \mid lazy \mid brown \mid black$
 (быстрый, ленивый, коричневый, черный)
9. $\langle \text{глагол} \rangle ::= goes \mid jumps \mid runs$
 (идет, прыгает, бежит)
10. $\langle \text{наречие} \rangle ::= quickly \mid easily$ (быстро, легко)
11. $\langle \text{предлог} \rangle ::= over \mid under \mid through$
 (над, под, через)

Приведенное множество синтаксических правил обычно используется для синтаксического анализа текстовой информации. Однако этот же фрагмент можно использовать (вместе с генератором случайных чисел) и для создания случайных предложений, подчиняющихся указанным

синтаксическим правилам. То есть можно как бы запустить процедуру синтаксического анализа в обратную сторону.

Чтобы построить какое-нибудь предложение, процесс генерации нужно запустить с правила 1. Здесь имеются две возможности, и для выбора какой-то одной из них надо обратиться к генератору случайных чисел. Затем процесс генерации развивается в соответствии с выбранной альтернативой, т. е. если выбрана первая из возможностей, то управление сначала должно перейти к правилу 2 для создания «существ. фрагмента», а после того как это сделано, — к правилу 4 для создания «глагольн. фрагмента». Поскольку правила 2 и 4, в свою очередь, передают управление другим правилам, необходимо иметь *стек* («магазинную память») для вспоминания незаконченных моментов в различных правилах: когда осуществляется первая передача управления от правила 1 к правилу 2, в стек помещается первый элемент и указатель на следующий по порядку элемент (в соответствии с выбранной альтернативой) и т. д. А если в процессе генерации встречается терминальный символ, то он добавляется к создаваемой выходной строке.

Указанный способ генерации предложений позволяет получить такие фразы как: «The quick brown fox jumps over the lazy dog» — «проворная коричневая лиса перепрыгнула через ленивую собаку».

В словаре нашего множества содержится всего 18 слов, выступающих в качестве терминальных символов: два артикля, четыре существительных, четыре прилагательных и т. д. Однако описанный выше метод применим и к большим словарям, содержащим яркие, выразительные слова. С их помощью можно «заставить» компьютер сочинять нерифмованные тексты, а для компьютерного написания стихов необходимо введение дополнительных правил рифмования фраз.

Однако, чтобы сочиненные компьютером произведения имели смысл, необходимо ввести в систему базу соответствующих знаний. Каждое вводимое в словарь слово должно быть увязано с другими не только синтаксически, но и семантически. Различные сочетания слов должны быть оценены при помощи некоторых оценивающих параметров, задающих

уровень смыслового соответствия. Такие параметры должны использоваться при построении фраз, так же как в экспертных системах при получении заключений используются коэффициенты доверия. Этими параметрами можно регулировать уровень осмысленности и степень абстрагизма создаваемого произведения, определять его характер, жанр и смысловую направленность. И, как и в других интеллектуальных системах, в системах стихотворчества коэффициенты доверия могут меняться в процессе работы программы, т. е. алгоритмы стихотворчества могут быть обучаемыми.

Таким образом, существующий уровень развития инструментальных средств и методов искусственного интеллекта позволяет создать более или менее приемлемые алгоритмы поэтического творчества. Это свидетельствует, что и данный вид человеческой деятельности в принципе поддается компьютерному моделированию.



Вопросы и задания

1. Перечислите общие принципы построения математических моделей.
2. Сформулируйте определение инварианта и приведите примеры инвариантов, известных вам из математики, физики и искусства.
3. Сформулируйте теорему Геделя и поясните, какое отношение она имеет к творчеству.
4. Что такое творчество с точки зрения теории инвариантов?
5. Сформулируйте теорему Мак-Каллока — Питтса и поясните, какое отношение она имеет к творчеству?
6. Если бы вам нужно было написать программу для компьютерного сочинения музыки, то какую блок-схему вы бы предложили?
7. Поясните, как можно использовать для сочинения стихов формализм Бекуса — Наура. Как, пользуясь этим алгоритмом, придавать сочиненным произведениям смысл, менять степень их абстрагизма, определять их характер, жанр и т. д.?

ГЛАВА 6

НАСТОЯЩЕЕ И БУДУЩЕЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

6.1. Искусственный интеллект и интеллектуальные информационные системы

По мнению многих исследователей, единого общепризнанного определения искусственного интеллекта в настоящее время не существует, как не существует и универсального определения человеческого интеллекта.

В ранний период возникновения и становления науки об искусственном интеллекте с этим понятием связывали большие надежды на создание машин, способных мыслить и чувствовать как человек, способных соперничать с человеческим мозгом и даже превзойти его. Эти идеи, на долгое время захватившие воображение энтузиастов, вызвали целую волну споров и критики. Они поставили ряд глобальных философских вопросов, так и оставшихся без ответа:

- можно ли создать мыслящую машину, если мы сами не знаем до конца, что такое мышление и каковы его механизмы?
- как мы можем понять и оценить, что у машины есть сознание, чувства, эмоции?
- не деградирует ли человечество, не превратится ли оно в слуг создаваемого им искусственного разума?

В результате в искусственном интеллекте, как и во множестве других наук, произошло разделение исследователей на «романтиков» — теоретиков, занимающихся глобальными вопросами создания искусственного разума, и «прагматиков» — практиков, развивающих и реализую-

щих идеи искусственного интеллекта для решения локальных практических проблем. Первое направление иногда называют «*сильным искусственным интеллектом*» (*Strong AI*), а второе — «*слабым искусственным интеллектом*» (*Weak AI*). Чтобы отделить себя от «философов», прагматики стали называть себя разработчиками *интеллектуальных информационных систем*, или *систем искусственного интеллекта*.

Впрочем, обе вышеназванные ветви искусственного интеллекта сосуществуют, взаимно дополняя и обогащая друг друга. Так, фундаментальная идея создания глобального искусственного разума в виде универсальной формулы знаний о мире, высказанная более 700 лет назад Раймундом Луллием (см. раздел 1.1), нашла блестящее воплощение в экспертных системах (см. раздел 2.3), впервые превративших искусственный интеллект в высокорентабельную сферу бизнеса. Идеи Мак-Каллока — Питтса — Розенблатта (см. раздел 3.1), предложивших способ моделирования мозга на структурном уровне, стали основой *нейроинформатики* — науки, побившей все рекорды чрезвычайным многообразием практических приложений. Попытки объяснить механизм возникновения разума как результата взаимодействия и развития совокупности относительно простых сущностей, принятые Гербертом Саймоном, Джоном Холландом и другими исследователями, привели к появлению эффективных инструментов решения широкого круга прикладных задач — таких, как генетические алгоритмы и мульти-агентные системы. С другой стороны, успехи практического применения интеллектуальных систем являются, во-первых, критерием правильности заложенных в их основу парадигм, а во-вторых, стимулом дальнейшего развития и совершенствования этих парадигм, поводом для дальнейших исследований мозга, раскрытия его новых тайн, все более полного понимания человеческого интеллекта.

По-видимому, человеческий мозг, так же как и сам человек, никогда не будет познан до конца. Люди будут постоянно решать проблемы сильного искусственного интеллекта, сталкиваясь с все новыми трудностями и преодоле-

вая их. И этот процесс, по-видимому, бесконечен. Но тогда бесконечен и другой процесс — создания и развития все более совершенных интеллектуальных информационных систем и технологий.

6.2. Нейронные сети и экспертные системы

По мнению многих исследователей, интеллект — это очень сложная область знаний, которую невозможно описать с помощью какой-то одной теории, поэтому ученые строят целую иерархию теорий, характеризующих интеллект на разных уровнях абстракции. На самом нижнем уровне этой иерархии находятся нейронные сети, генетические алгоритмы и другие формы эволюционирующих вычислений, а на более высоком уровне абстракции — экспертные системы, интеллектуальные агенты и системы понимания естественного языка.

Остановимся на двух крайних иерархических ступенях — самой верхней и самой нижней. Отметим для начала, что строительство «здания» искусственного интеллекта началось не снизу, а сверху, причем с огромным разрывом во времени. Идея создания интеллектуальной системы, основанной на явных знаниях, была высказана более 700 лет назад Раймундом Луллием, а идея математического нейрона была опубликована Мак-Каллоком и Питтсом в 1943 г. Своего апогея в области практических приложений технология экспертных систем достигла в середине 1980-х гг. Именно тогда, с конца 1980-х гг., началась и конкуренция на рынке программных приложений между экспертными системами и нейронными сетями, которая продолжается и по настоящее время.

Строя прогнозы на будущее, многие авторы отмечают наметившуюся тенденцию к сближению этих двух крайностей в виде появления интеллектуальных систем, использующих сразу обе технологии. Тем не менее очень часто при решении практических задач приходится выбирать, какой из этих двух конкурирующих технологий отдать предпочтение, поэтому укажем их плюсы и минусы.

Основным *недостатком нейронных сетей* является их невербальность. Если вывод, полученный экспертной системой, всегда можно логически обосновать, запустив «подсистему объяснений» (см. раздел 2.3.2), то в случае применения нейронной сети такая возможность отсутствует (см. раздел 3.2.9). Другой их недостаток — отсутствие законченной фундаментальной теории и, как следствие, отсутствие надежных и достаточно универсальных методик проектирования и обучения нейросетей, гарантирующих успех их применения. По этому поводу программисты шутят, что нейросетевые технологии все еще не стали в полной мере наукой и содержат элементы искусства и даже религии.

Одним из главных *преимуществ нейросетевых технологий* является сравнительно низкая трудоемкость их создания и, как следствие, низкая себестоимость программной продукции. При нейросетевом моделировании отсутствуют такие трудоемкие и дорогостоящие процедуры, как извлечение знаний и их формализация (см. раздел 3.2.1). Вместо этого предлагается целый арсенал алгоритмов обучения, осуществляющих автоматическое извлечение знаний, и стандартный способ их хранения в виде сил синаптических связей. И вместе с тем нейросети предоставляют универсальный способ решения многих нестандартных задач. Если для создания средней по сложности экспертной системы требуются месяцы и годы работы коллектива специалистов, включающего инженера-когнитолога, программиста, эксперта и пользователя (см. раздел 2.3.2), то вполне прилично работающая нейронная сеть может быть всего с нескольких попыток сгенерирована и обучена рядовым пользователем нейропакета, скачанного из сети Интернет.

К серьезному преимуществу нейросетевых технологий можно отнести и принципиальную возможность выявления новых, еще не известных экспертам знаний (см. раздел 3.2). Нейросети особенно эффективны там, где нужен аналог человеческой интуиции, где законы и закономерности предметной области неизвестны либо существуют проблемы с их выявлением и формализацией.

В противоположность нейросетям, *экспертная система* оперирует только с теми знаниями, законами и закономерностями предметных областей, которые заложены в нее экспертом. Результат работы экспертной системы всегда может быть предсказан экспертом, тогда как нейронные сети способны получать новые, совершенно неожиданные выводы, выявлять неизвестные законы и закономерности моделируемых процессов, объектов или явлений.

В целом не следует переоценивать возможности нейросетевых технологий. Обычно нейросети уступают детерминированным моделям в тех предметных областях, где знания известны и могут быть хорошо описаны и формализованы. Нейросети нет смысла применять и там, где выходные параметры не зависят или слабо зависят от входных.

Опыт показывает, что нейросети целесообразно использовать при следующих условиях, если:

- 1) решение задачи затруднительно для человека;
- 2) при решении задачи можно выделить множество входных факторов (сигналов, признаков, данных и т. п.) и множество выходных факторов;
- 3) изменения входных факторов приводят к изменению выходных.

6.3. Нейросетевые технологии и метод математического моделирования

Ранее мы уже много говорили о том, что метод математического моделирования является одним из самых мощных и перспективных современных инструментов получения научных и практических знаний, что он сыграл огромную роль в формировании нашей цивилизации и что эта роль в последнее время все более усиливается.

Следует заметить, что сам метод математического моделирования как таковой возник много веков назад одновременно с математикой. Собственно говоря, цель математики всегда и заключалась в том, чтобы отражать (т. е. моделировать!) на своем специфическом языке закономерности окружающего мира.

Грандиозный толчок развитию метода математического моделирования дало во второй половине XX в. появление компьютера и компьютерных технологий. Кроме того, что компьютер взял на себя «черновую» работу, связанную с обработкой и визуализацией научной информации, он еще и научился выполнять такие интеллектуальные этапы математического моделирования как, например, поиск решения математических уравнений.

Однако при всем этом за человеком оставался самый первый этап, связанный с построением, или, как говорят, с формулировкой математических моделей. За человеком оставалось выявление значащих (влияющих на результат моделирования) входных параметров модели, а также решающее слово о том, какие законы природы, закономерности изучаемой предметной области (объекта, процесса, явления) следует заложить в математическую модель.

И вот в конце XX — начале XXI в. мы становимся свидетелями события, значение и масштабы последствий которого в истории цивилизации мы пока не в силах до конца осознать. Мы становимся свидетелями того, как эти высокоинтеллектуальные, сугубо «человеческие» функции берет на себя компьютер. Оказывается, он способен заменить человека и здесь. Оказывается, он может (страшно даже подумать!) совершать научные открытия, может открывать законы природы и сам закладывать их в математические модели! Другими словами, компьютер теперь может не только реализовывать математические модели и интерпретировать результаты моделирования, но и создавать эти математические модели и тем самым полностью осуществлять весь процесс математического моделирования от «А» до «Я». Причем во многих случаях компьютер справляется с созданием математических моделей лучше, чем человек. Например, компьютер может выявлять и учитывать в создаваемых им моделях закономерности и внутренние связи, не известные специалистам (см. раздел 3.2). В результате модели, создаваемые компьютером, учитывают большее количество факторов, законов и закономерностей, влияющих на результат моделирования, что, конечно же, улучшает качество моделей.

Что же это за компьютер, который может не только обрабатывать математические формулы и решать математические уравнения, но и самостоятельно формулировать эти формулы и уравнения, извлекать их из анализа предметных областей?

Как вы убедились, изучив данный элективный курс, это не совсем обычный компьютер. Все, что может обычный (фон-неймановский) компьютер — это складывать и вычитать двоичные числа. Более сложные операции, такие как умножение, деление и возведение в степень, обычный компьютер сводит к сложению и вычитанию, как и все остальное, что он вообще умеет делать. Согласитесь, что это совсем не то, что происходит в мозге ученого, когда он совершает научные открытия.

Открывать законы природы способен только человек, а точнее — его мозг. Значит компьютер должен быть адекватен мозгу, он должен быть создан, аналогично сказанному в Библии, «по образу и подобию» человеческого мозга. Именно идея создания компьютеров и компьютерных программ, работающих подобно человеческому мозгу, и лежит в основе одного из самых увлекательных и плодотворных разделов искусственного интеллекта — нейроинформатики.

6.5. Философские проблемы искусственного интеллекта

Прежде чем обсуждать философские проблемы искусственного интеллекта, вспомним еще раз краткую историю его появления и развития.

Исторически первой была модель искусственного интеллекта, основанного на явных знаниях, которая безраздельно господствовала начиная с XIII в. до середины XX в. Появление персептрона в 1940-х гг. и его первые успехи были встречены с надеждами и энтузиазмом, который, однако, продержался недолго, и к середине 1980-х гг. подавляющее большинство систем искусственного интеллекта представляло собой экспертные системы, ориентированные на конкретные предметные области. Однако победа вы-

сокоуровневой стратегии была временной. Конец XX в. и начало XXI в. ознаменовались широкомасштабным наступлением нейросетевых и нейрокомпьютерных технологий, причем можно отметить как применение этих технологий в новых предметных областях, так и вытеснение экспертных систем из областей, где они традиционно применялись ранее. И вот уже в течение полувека между высоко- и низкоуровневой стратегиями идет жесточайшая конкуренция.

Третья стратегия, появившаяся на свет в середине 1970-х гг. благодаря работам профессора Мичиганского университета Дж. Холланда, навеяна дарвиновской теорией возникновения и эволюции жизни на Земле. Этот подход предполагает поиск источника интеллекта в самом процессе эволюции и взаимодействия особей. Знания в эволюционных моделях приобретаются в ходе конкурентной борьбы между особями.

На сегодня безусловным лидером как в области теоретических исследований, так и в части практических приложений являются нейросетевые и нейрокомпьютерные технологии. Попытки моделирования мозга на самом низком уровне абстрагирования — на уровне нейронной структуры показывают себя как наиболее плодотворные технологии создания интеллектуальных информационных систем.

Искусственный интеллект сегодня является лидирующей научной отраслью (см. раздел 1.3), поэтому как никакая другая дисциплина он нуждается в философском осмыслении результатов, парадигм, стратегий, методов. Как и всякая другая наука, искусственный интеллект имеет свои корни в философии. Отражение идей высокоуровневой стратегии искусственного интеллекта можно найти, например, в аристотелевской «Логике», а в полном объеме формальная аксиоматизация логических рассуждений представлена в работах таких великих философов как Готлоб Фреге, Бертран Рассел, Курт Гедель и др.

Однако работы философов прошлого, разумеется, не охватывают коренных изменений, которые искусственный интеллект претерпел совсем недавно — в конце XX — начале XXI в. К таким коренным изменениям в первую очередь

относится создание нейрокомпьютера, построенного по образу и подобию человеческого мозга. Именно идея нейрокомпьютера и феномен триумфального практического внедрения нейросетевых и нейрокомпьютерных технологий менее всего исследованы философами, но более всего в этом нуждаются. Между тем для философской оценки последствий появления нейрокомпьютера и связанных с этим событий можно обратиться к широко известному примеру из Библии — о том как Бог создал Адама и Еву «по своему образу и подобию». Создал с определенной целью, ожидая от них определенных качеств и свойств, однако созданные люди обнаружили несколько иные качества, чего Создатель от них даже и не ожидал. И, кстати, согласно той же Библии, за одно из таких непредвиденных качеств Адам и Ева были выдворены Создателем из Рая...

Обратимся теперь к истории нейрокомпьютеров. Как отмечает создатель первого нейрокомпьютера Ф. Розенблатт, его перцептрон был построен «по образу и подобию» человеческого мозга и предназначался для решения задачи узнавания букв латинского алфавита, причем только тех, которым он обучался. И эта цель Ф. Розенблаттом была успешно достигнута. Однако, наблюдая за последующими событиями, можно заключить, что со временем нейрокомпьютер стал проявлять такие свойства, которых его создатель (Ф. Розенблатт) от него не ожидал. Вот некоторые из них:

- 1) способность к обобщению (см. раздел 3.1.4) — перцептрон, обученный распознавать определенное количество образов, оказывается способен распознавать не только те образы, на которых его обучали, но и такие, которых в обучающем множестве не было, т. е. образы, которых он ни разу «не видел»;
- 2) интуиция, или «шестое чувство» (см. раздел 3.2.9) — способность делать правильные выводы и заключения, основываясь на информации, не достаточной для применения традиционных вербальных методов;
- 3) живучесть. Известны случаи, когда люди продолжали совершать разумные поступки, потеряв в результате физической травмы до 30% объема мозга. То же самое про-

исходит с нейрокомпьютерами: при выводе из строя отдельных нейрочипов нейрокомпьютеры хотя и менее качественно, но продолжают решать поставленные задачи.

Отметим еще раз, что перечисленные свойства нейросетей и нейрокомпьютеров не предусматривались их создателями заранее, а обнаружились уже в процессе их эксплуатации. Эти свойства проявились сами по себе вследствие того, что нейросети и нейрокомпьютеры являются моделями мозга. Они выполнены «по образу и подобию» человеческого мозга, поэтому напрямую «наследуют» его свойства.

Вышеприведенные рассуждения и факты свидетельствуют о том, что между нейрокомпьютером как моделью человеческого мозга и человеком как «моделью Бога» существует некая аналогия. Возможно, эту аналогию можно взять за основу при прогнозировании последствий развития искусственного интеллекта и интеллектуальных систем. Возможно, искусственная модель человека будет и дальше обнаруживать непредусмотренные создателями свойства, и между моделями и их «прототипами» — людьми — могут возникнуть непонимание и проблемы, подобные тем, которые описаны в Библии...



Вопросы и задания

1. Попробуйте сформулировать определения человеческого интеллекта и искусственного интеллекта.
2. В чем состоит различие между сильным и слабым искусственным интеллектом?
3. Перечислите уровни абстракции, на которых разрабатываются теории искусственного интеллекта.
4. Перечислите преимущества и недостатки нейросетевых технологий по сравнению с технологией экспертных систем.
5. Какие стратегии искусственного интеллекта являются лидирующими?
6. Какие свойства нейрокомпьютеры унаследовали от своего прототипа — человеческого мозга?

ТЕРМИНОЛОГИЧЕСКИЙ СЛОВАРЬ

А **Активационная функция нейрона** — функция, с помощью которой вырабатывается выходной сигнал нейрона. Аргументом активационной функции является взвешенная сумма входных сигналов.

Алгоритм — набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи.

Аппроксимация — математический прием, позволяющий построить поверхность или линию, максимально близко расположенную к заданному набору точек.

В **Входной вектор персептрона X** — вектор, компонентами которого являются параметры $x_1, x_2, \dots$, подаваемые на нейроны входного слоя персептрона.

Вербализация знаний — объяснение сути знаний с помощью слов (от англ. *verb* — «слово»).

Вероятность — численная мера степени возможности наступления случайного события.

Виртуальный эксперимент — эксперимент, выполняемый не над реальным объектом, а над его математической компьютерной моделью.

Выходной вектор персептрона (Y или D) — вектор, компонентами которого являются величины выходных сигналов нейронов выходного слоя персептрона. Выходной вектор может быть *желаемым* (D), компонентами которого являются параметры $d_1, d_2, \dots$, заданные примерами обучающего, тестирующего или подтверждающего множеств, либо *действительным* (иногда — *прогноznым*) Y , компоненты которого $y_1, y_2, \dots$ являются результатами вычислений персептрона. Разница между желаемыми и действительными векторами (параметрами) называется *погрешностью* персептрона.

Г **Гиперразмерность** — свойство нейронных сетей терять способность к обобщению при чрезмерном увеличении количества нейронов скрытых слоев.

Градиент функции — вектор, проекциями которого на оси координат являются частные производные от функции по этим координатам. Градиент функции всегда направлен в сторону ее максимального возрастания.

Д **Данные** — отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства.

З **Знания** — результат мыслительной деятельности человека. Знания обобщают его опыт, полученный в ходе практической деятельности или научных исследований.

Значащий параметр — входной параметр математической модели или нейронной сети, оказывающий существенное влияние на результат моделирования.

И **Интуиция**, или «шестое чувство» — способность человеческого мозга делать правильные выводы и принимать правильные решения, которые логическим путем из имеющейся информации получить не удастся.

Итерация (от лат. *iteratio* — «повторение») — одно из целого ряда повторений какой-либо математической операции, использующее результат предыдущей аналогичной операции.

К **Кибернетика «черного ящика»** — стратегическое направление исследований в области искусственного интеллекта, в основе которого лежит идея создания модели мозга любыми средствами, без учета его строения и протекающих в нем процессов. Синонимы: *высокоуровневая*, или *нисходящая стратегия искусственного интеллекта*.

Л **Линейно-неразделимая задача** — задача, которая не может быть решена с помощью однослойного персептрона.

М **Минимум глобальный** — точка на графике функции, в которой функция принимает наименьшее значение на рассматриваемом интервале.

Минимум локальный — точка на графике функции, в которой значение функции меньше, чем в соседних точках. Однако значение функции в точке локального минимума больше, чем в точке глобального минимума функции.

Множество обучающее (L) — состоит из примеров, на которых обучается нейросеть.

Множество подтверждающее (P) — состоит из примеров, которые не участвовали ни в обучении, ни в тестировании сети.

Множество тестирующее (T) — состоит из примеров, которые не участвовали в процессе обучения.

Н Нейроинформатика — область научных исследований, лежащая на пересечении нейронауки и информатики. Нейроинформатика представляет собой раздел искусственного интеллекта, объединяющий нейросетевые и нейрокомпьютерные технологии.

Нейрокибернетика — стратегическое направление исследований в области искусственного интеллекта, в основе которого лежит идея создания ЭВМ из искусственных нейронов. Синонимы: *низкоуровневая, или восходящая стратегия искусственного интеллекта.*

Нейроконтроллер — электронное устройство или компьютерная программа, предназначенные для управления процессом или объектом и выполненные с использованием методов нейрокибернетики.

Нейрокомпьютер — устройство, реализующее нейронную сеть на аппаратном уровне.

Нейрон биологический (естественный) — элементарная ячейка (клетка), из множества которых состоит мозг. Нейроны мозга соединены между собой нервными волокнами, через которые они обмениваются между собой электрическими сигналами.

Нейрон математический (модельный, формальный, искусственный) — узел искусственной нейронной сети, являющийся упрощенной моделью биологического нейрона.

Нейропакет — программа, предназначенная для создания нейронных сетей, их обучения, оптимизации, тестирования, прогнозирования и других видов работ с нейронными сетями.

О Ошибка обобщения (тестирования) ε_T — представляет собой разницу между прогнозным (действительным) выходным вектором Y и желаемым выходным вектором D , вычисленную на примерах тестирующего множества T .

Ошибка обучения ε_L — представляет собой разницу между прогнозным (действительным) выходным вектором $\mathbf{Y}$ и желаемым выходным вектором $\mathbf{D}$, вычисленную на примерах обучающего множества $\mathbf{L}$.

Ошибка прогнозирования ε_P — представляет собой разницу между прогнозным (действительным) выходным вектором $\mathbf{Y}$ и желаемым выходным вектором $\mathbf{D}$, вычисленную на примерах подтверждающего множества $\mathbf{P}$.

П Переобучение — свойство нейронных сетей терять способность к обобщению при чрезмерном увеличении количества нейронов скрытых слоев.

Ц Целевая функция — функция, экстремальное значение которой ищется при решении задач оптимизации.

Э Эволюционное моделирование — стратегическое направление исследований в области искусственного интеллекта, в основе которого лежит идея использования элементов эволюционной теории Ч. Дарвина.

Эвристики — строго не обоснованные правила, которые используются для поиска возможного направления решения проблемы.

ЛИТЕРАТУРА

1. Арнольд В. И. О функциях трех переменных // Доклады АН СССР. Т. 114. № 4. М.: Изд-во АН СССР, 1957.
2. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2001.
3. Горбань А. Н., Миркес Е. М. Логически прозрачные нейронные сети // Нейроинформатика и ее приложения: Тезисы докладов III всесоюзного семинара. Красноярск: Изд-во КГТУ, 1999.
4. Зибатова А. Н., Петров А. М., Сичинава З. И., Сошников А. П., Ясницкий Л. Н. Интеллектуальный полиграф // Российский полиграф. 2006. № 1. С. 76–83.
5. Информатика : Учебник / Под ред. Н. В. Макаровой. М.: Финансы и статистика, 1997.
6. Искусственный интеллект. Кн. 1. Системы общения и экспертные системы / Под ред. Э. В. Попова. М.: Радио и связь, 1990.
7. Колмогоров А. Н. О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения // Доклады АН СССР. Т. 114. М.: Изд-во АН СССР, 1957.
8. Люгер Дж. Ф. Искусственный интеллект: Стратегии и методы решения проблем / Пер. с англ. М.: Вильямс, 2003.
9. Минский М., Пайперт С. Перцептроны. М.: Мир, 1971.
10. Мкртчян С. О. Нейроны и нейронные сети. Введение в теорию формальных нейронов. М.: Энергия, 1971.
11. Мурашов Д. И., Ясницкий Л. Н. Социальный генетический алгоритм // Вестник Пермского университета. Математика. Информатика. Механика. Пермь: Изд-во Перм. ун-та, 2006. С. 53–60.
12. Обработка знаний / Под ред. С. Осуга. М.: Мир, 1989.

13. *Осовский С.* Нейронные сети для обработки информации / Пер. с польского. М.: Финансы и статистика, 2002.
14. Представление знаний / Под ред. С. Осуги, Ю. Саэки. М.: Мир, 1990.
15. Представление и использование знаний / Под ред. Х. Уэно, М. Исидзука. М.: Мир, 1987.
16. *Рассел С., Норвинг П.* Искусственный интеллект: современный подход / Пер. с англ. М.: Вильямс, 2006.
17. *Розенблатт Ф.* Принципы нейродинамики. Перцептроны и теория механизмов мозга. М.: Мир, 1965.
18. *Уоссермен Ф.* Нейрокомпьютерная техника. М.: Мир, 1992.
19. *Хайкин С.* Нейронные сети: Полный курс. 2-е издание / Пер. с англ. М.: Вильямс, 2006.
20. *Черепанов Ф. М., Ясницкий Л. Н.* Нейросетевой фильтр для исключения посторонних выбросов в статистической информации // Вестник Пермского университета. Информационные системы и технологии. Вып. 5. Пермь: Изд-во Перм. ун-та, 2007. С. 151–155.
21. *Эндрю А.* Искусственный интеллект. М.: Мир, 1985.
22. *Ясницкий Л. Н., Черепанов Ф. М.* Искусственный интеллект. Элективный курс: Методическое пособие по преподаванию. М.: БИНОМ. Лаборатория знаний, 2011.
23. *Ясницкий Л. Н.* Введение в искусственный интеллект. 2-е изд. М.: Академия, 2008.
24. *Ясницкий Л. Н.* Интеллектуальные информационные технологии и системы / Пермь: Изд-во Перм. ун-та, 2007.
25. *Ясницкий Л. Н.* Пермская научная школа искусственного интеллекта и ее инновационные проекты / Под ред. Л. Н. Ясницкого. 2-е изд. Москва-Ижевск: НИЦ «Регулярная и хаотическая динамика», 2008.
26. *Ясницкий Л. Н.* О возможностях применения методов искусственного интеллекта в политологии // Вестник Пермского университета. Политология. Пермь: Изд-во Перм. ун-та, 2008. Вып. 2 (4). С. 147–155.
27. *Ясницкий Л. Н., Данилевич Т. В.* Современные проблемы науки. М.: БИНОМ. Лаборатория знаний, 2008.
28. *Golovko V., Savitsky Ju., Gladischuk V.* A neural net for prediction problems // Proceedings of Int. Conf. On technical informatics. Timisoara: University of Timisoara, Romania, 1996. P. 49–52.

29. *Hebb D.O.* The Organization of Behavior. New York: John Wiley & Sons, 1949.
30. *Hecht-Nielsen R.* Kolmogorov's Mapping Neural Network Existence Theorem // IEEE First Annual Int. Conf. On Neural Networks. San Diego, 1987. V. 3.
31. *Holland J.H.* Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975.
32. *McCulloch W.S., Pitts W.* A Logical Calculus of Ideas Immanent in Nervous Activity // Bull. Mathematical Biophysics. 1943. Vol. 5.
33. *Rosenblatt F.* The perceptron: a probabilistic model for information storage and organization in the brain // Psychological Review. 1958. Vol. 65.
34. *Rummelhart D.E., Hilton G.E., Williams R. J.* Learning internal representations by error propagation. In McClelland *et al.*, 1986.
35. *Werbos P.* Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences // Phd Thesis, Dept. of Applied Mathematics. Cambridge: Harvard University, 1974.
36. *Widrow B., Hoff M.E.* Adaptive switching circuits // 1960 IRE WESTCON Conferenction Record. New York, 1960.