Содержание

Веб-сайт

www.deeplearning.book

Книгу сопровождает указанный выше сайт, где представлены упражнения, слайды, исправления ошибок и другие материалы, полезные читателям и преподавателям.

Обозначения

В этом разделе приведен краткий перечень обозначений, используемых в книге. Большая часть соответствующего математического аппарата описана в главах 2–4.

Числа и массивы

a	скаляр (целый или вещественный)
$\boldsymbol{a}$	вектор
$\boldsymbol{A}$	матрица
$\mathbf{A}$	тензор
$\boldsymbol{I}_n$	единичная матрица с n строками и n столбцами
$\boldsymbol{I}$	единичная матрица, размер которой определяется контекстом
$e^{(i)}$	стандартный базисный вектор [0, ..., 0, 1, 0..., 0], содержащий 1 в i-й позиции
$\text{diag}(\boldsymbol{a})$	квадратная диагональная матрица, на диагонали которой находятся элементы вектора $\boldsymbol{a}$
a	случайная скалярная величина
$\mathbf{a}$	случайный вектор
$\mathbf{A}$	случайная матрица

Множества и графы

$\mathbb{A}$	множество
$\mathbb{R}$	множество вещественных чисел
$\{0, 1\}$	множество из двух элементов: 0 и 1
$\{0, 1, ..., n\}$	множество целых чисел от 0 до n включительно
$[a, b]$	замкнутый интервал вещественной прямой от a до b, включающий границы
$(a, b]$	интервал вещественной прямой, не включающий a, но включающий b
$\mathbb{A} \setminus \mathbb{B}$	разность множеств, т. е. множество, содержащее все элементы $\mathbb{A}$, не являющиеся элементами $\mathbb{B}$
$\mathcal{G}$	граф
$Pa_{\mathcal{G}}(x_i)$	родители x_i в $\mathcal{G}$

Индексирование

a_i	i-й элемент вектора a, индексирование начинается с 1
a_{-i}	все элементы вектора a, кроме i-го
$A_{i,j}$	элемент матрицы $\boldsymbol{A}$ в позиции (i, j)
$\boldsymbol{A}_{i,:}$	i-я строка матрицы $\boldsymbol{A}$
$\boldsymbol{A}_{:,i}$	i-й столбец матрицы $\boldsymbol{A}$
$\mathbf{A}_{i,j,k}$	элемент трехмерного тензора $\mathbf{A}$ в позиции (i, j, k)
$\mathbf{A}_{:,:,i}$	двумерная срезка трехмерного тензора $\mathbf{A}$
a_i	i-й элемент случайного вектора $\mathbf{a}$

Операции линейной алгебры

$\boldsymbol{A}^\top$	матрица, транспонированная к $\boldsymbol{A}$
$\boldsymbol{A}^+$	псевдообратная матрица Мура-Пенроуза

$\boldsymbol{A} \odot \boldsymbol{B}$	поэлементное произведение матриц $\boldsymbol{A}$ и $\boldsymbol{B}$ (произведение Адамара)
$\det(\boldsymbol{A})$	определитель $\boldsymbol{A}$

Математический анализ

$\frac{dy}{dx}$	производная y по x
$\frac{\partial y}{\partial x}$	частная производная y по x
$\nabla_{\boldsymbol{x}} y$	градиент y по $\boldsymbol{x}$
$\nabla_{\boldsymbol{X}} y$	матрица производных y относительно $\boldsymbol{X}$
$\nabla_{\mathbf{X}} y$	тензор производных y относительно $\mathbf{X}$
$\frac{\partial f}{\partial x}$	матрица Якоби $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ функции $f: \mathbb{R}^n \to \mathbb{R}^m$
$\nabla_x^2 f(x)$ или $\boldsymbol{H}(f)(\boldsymbol{x})$	гессиан функции f в точке $\boldsymbol{x}$
$\int f(\boldsymbol{x}) d\boldsymbol{x}$	определенный интеграл по всей области определения $\boldsymbol{x}$
$\int_{\mathbb{S}} f(\boldsymbol{x}) d\boldsymbol{x}$	определенный интеграл по множеству $\mathbb{S}$

Теория вероятности и теория информации

$\mathrm{a} \perp \mathrm{b}$	случайные величины a и b независимы
$\mathrm{a} \perp \mathrm{b} \mid \mathrm{c}$	они условно независимы при условии c
$P(\mathrm{a})$	распределение вероятности дискретной случайной величины
$p(\mathrm{a})$	распределение вероятности непрерывной случайной величины или величины, тип которой не задан
$\mathrm{a} \sim P$	случайная величина a имеет распределение P
$\mathbb{E}_{x \sim P}[f(x)]$ или $\mathbb{E} f(x)$	математическое ожидание $f(x)$ при заданном распределении $P(x)$
$\mathrm{Var}(f(x))$	дисперсия $f(x)$ при заданном распределении $P(x)$
$\mathrm{Cov}(f(x), g(x))$	ковариация $f(x)$ и $g(x)$ при заданном распределении $P(x)$
$H(\mathrm{x})$	энтропия Шеннона случайной величины x
$D_{\mathrm{KL}}(P \Vert Q)$	расстояние Кульбака–Лейблера между P и Q
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	нормальное распределение случайной величины $\boldsymbol{x}$ со средним $\boldsymbol{\mu}$ и ковариацией $\boldsymbol{\Sigma}$

Функции

$f: \mathbb{A} \to \mathbb{B}$	функция f с областью определения $\mathbb{A}$ и областью значений $\mathbb{B}$
$f \circ g$	композиция функций f и g
$f(\boldsymbol{x}; \theta)$	функция от x, параметризованная θ (иногда, чтобы не утяжелять формулы, мы пишем $f(x)$, опуская аргумент θ)
$\log x$	натуральный логарифм x
$\sigma(x)$	логистическая сигмоида, $1 / (1 + \exp(-x))$
$\zeta(x)$	функция $\log(1 + \exp(x))$
$\Vert \boldsymbol{x} \Vert_p$	норма L^p вектора $\boldsymbol{x}$
$\Vert \boldsymbol{x} \Vert$	норма L^2 вектора $\boldsymbol{x}$
x^+	положительная часть x, т. е. $\max(0, x)$
$\mathbf{1}_{\mathrm{condition}}$	1, если условие *condition* истинно, иначе 0

Иногда мы применяем функцию f со скалярным аргументом к вектору, матрице или тензору: $f(\boldsymbol{x})$, $f(\boldsymbol{X})$ или $f(\mathbf{X})$. Это означает, что функция f применяется к каждому элементу массива. Например, запись $\mathbf{C} = \sigma(\mathbf{X})$ означает, что $C_{i,j,k} = \sigma(X_{i,j,k})$ для всех i, j, k.

Наборы данных и распределения

p_{data}	распределение, порождающее данные
$\hat{p}_{\text{data}}$	эмпирическое распределение, определенное обучающим набором
$\mathbb{X}$	обучающий набор примеров
$\boldsymbol{x}^{(i)}$	i-й пример из входного набора данных
$y^{(i)}$ или $\boldsymbol{y}^{(i}$	метка, ассоциированная с $\boldsymbol{x}^{(i)}$ при обучении с учителем
$\boldsymbol{X}$	матрица $m \times n$, в строке $\boldsymbol{X}_{i,:}$ которой находится входной пример $\boldsymbol{x}^{(i)}$

Глава 1

Введение

Изобретатели давно мечтали создать думающую машину. Эти мечты восходят еще к Древней Греции. Персонажей мифов – Пигмалиона, Дедала, Гефеста – можно было бы назвать легендарными изобретателями, а их творения – Галатею, Талоса и Пандору – искусственной жизнью (Ovid and Martin, 2004; Sparkes, 1996; Tandy, 1997).

Впервые задумавшись о программируемых вычислительных машинах, человек задался вопросом, смогут ли они стать разумными, – за сотню с лишним лет до построения компьютера (Lovelace, 1842). Сегодня **искусственный интеллект** (ИИ) – бурно развивающаяся дисциплина, имеющая многочисленные приложения. Мы хотим иметь интеллектуальные программы, которые могли бы автоматизировать рутинный труд, понимали речь и изображения, ставили медицинские диагнозы и поддерживали научные исследования.

Когда наука об искусственном интеллекте только зарождалась, были быстро исследованы и решены некоторые задачи, трудные для человека, но относительно простые для компьютеров – описываемые с помощью списка формальных математических правил. Настоящим испытанием для искусственного интеллекта стали задачи, которые легко решаются человеком, но с трудом поддаются формализации, – задачи, которые мы решаем интуитивно, как бы автоматически: распознавание устной речи или лиц на картинке.

Эта книга посвящена решению таких интуитивных задач. Цель заключается в том, чтобы компьютер мог учиться на опыте и понимать мир в терминах иерархии понятий, каждое из которых определено через более простые понятия. Благодаря приобретению знаний опытным путем этот подход позволяет исключить этап формального описания человеком всех необходимых компьютеру знаний. Иерархическая организация дает компьютеру возможность учиться более сложным понятиям путем построения их из более простых. Граф, описывающий эту иерархию, будет глубоким – содержащим много уровней. Поэтому такой подход к ИИ называется **глубоким обучением**.

Ранние успехи ИИ в большинстве своем были достигнуты в относительно стерильной формальной среде, где от компьютера не требовались обширные знания о мире. Взять, к примеру, созданную IBM шахматную программу Deep Blue, которая в 1997 году обыграла чемпиона мира Гарри Каспарова (Hsu, 2002). Шахматы – это очень простой мир, состоящий всего из 64 клеток и 32 фигур, которые могут ходить лишь строго определенным образом. Разработка успешной стратегии игры в шахматы – огромное достижение, но трудность задачи – не в описании множества фигур и допустимых ходов на языке, понятном компьютеру. Для полного описания игры достаточно очень короткого списка формальных правил, который заранее составляется программистом.

Забавно, что абстрактные, формально поставленные задачи, требующие значительных умственных усилий от человека, для компьютера как раз наиболее просты. Компьютеры давно уже способны обыграть в шахматы сильнейших гроссмейстеров, но лишь в последние годы стали сопоставимы с человеком в части распознавания объектов или речи. В повседневной жизни человеку необходим гигантский объем знаний о мире. Знания эти субъективны и представлены на интуитивном уровне, поэтому выразить их формально затруднительно. Но чтобы вести себя «разумно», компьютерам нужны такие же знания. Одна из основных проблем искусственного интеллекта – как заложить эти неформальные знания в компьютер.

Авторы нескольких проектов в области ИИ пытались представить знания о мире с помощью формальных языков. Компьютер может автоматически рассуждать о предложениях на таком языке, применяя правила логического вывода. В основе таких подходов лежит **база знаний**. Ни один из этих проектов не привел к существенному успеху. Одним из самых известных был проект Cyc (Lenat and Guha, 1989) – машина логического вывода и база утверждений на языке CycL. За ввод утверждений отвечал штат учителей-людей. Процесс оказывается крайне громоздким. Люди изо всех сил пытаются придумать формальные правила, достаточно сложные для точного описания мира. Например, Cyc не сумел понять рассказ о человеке по имени Фред, который бреется по утрам (Linde, 1992). Его машина вывода обнаружила в рассказе противоречие: он знал, что в людях нет электрических деталей, но, поскольку Фред держал электрическую бритву, система решила, что объект «БреющийсяФред» содержит электрические детали. И задала вопрос: является ли Фред по-прежнему человеком, когда бреется.

Трудности, с которыми сталкиваются системы, опирающиеся на «зашитые в код» знания, наводят на мысль, что система с искусственным интеллектом должна уметь самостоятельно накапливать знания, отыскивая закономерности в исходных данных. Это умение называется **машинным обучением**. С появлением машинного обучения перед компьютерами открылась возможность подступиться к задачам, требующим знаний о реальном мире, и принимать решения, кажущиеся субъективными. Простой алгоритм машинного обучения – **логистическая регрессия** – может решить, следует ли рекомендовать кесарево сечение (Mor-Yosef et al., 1990). Другой простой алгоритм – **наивный байесовский классификатор** – умеет отделять нормальную электронную почту от спама.

Качество этих простых алгоритмов сильно зависит от представления исходных данных. Так, система ИИ, выдающая рекомендации о показанности кесарева сечения, не осматривает пациента. Вместо этого врач сообщает системе относящуюся к делу информацию, например о наличии или отсутствии рубца на матке. Каждый отдельный элемент информации, включаемый в представление о пациенте, называется **признаком**. Алгоритм логистической регрессии анализирует, как признаки пациента коррелируют с различными результатами. Но он не может никаким образом повлиять на определение признаков. Если алгоритму предложить снимок МРТ, а не формализованные врачом сведения, то он не сможет выдать полезную рекомендацию. Отдельные пиксели снимка практически не коррелированы с осложнениями, которые могут возникнуть во время родов.

Эта зависимость от представления является общим явлением, проявляющимся как в информатике, так и в повседневной жизни. Если говорить об информатике, то такие операции, как поиск в коллекции данных, будут производиться многократно

быстрее, если коллекция структурирована и подходящим образом индексирована. Люди же легко выполняют арифметические операции с числами, записанными арабскими цифрами, но тратят куда больше времени, если используются римские цифры. Неудивительно, что выбор представления оказывает огромное влияние на качество и производительность алгоритмов машинного обучения. На рис. 1.1 приведен простой наглядный пример.

Многие задачи ИИ можно решить, если правильно подобрать признаки, а затем предъявить их алгоритму машинного обучения. Например, в задаче идентификации говорящего по звукам речи полезным признаком является речевой тракт. Он позволяет с большой точностью определить, кто говорит: мужчина, женщина или ребенок.

Но во многих задачах нелегко понять, какие признаки выделять. Допустим, к примеру, что мы пишем программу обнаружения автомобилей на фотографиях. Мы знаем, что у автомобилей есть колеса, поэтому могли бы счесть присутствие колеса признаком. К сожалению, на уровне пикселей трудно описать, как выглядит колесо. Колесо имеет простую геометрическую форму, но распознавание его изображения может быть осложнено отбрасыванием теней, блеском солнца на металлических деталях автомобиля, наличием щитка, защищающего колесо от грязи, или объектов на переднем плане, частично загораживающих колесо, и т. д.

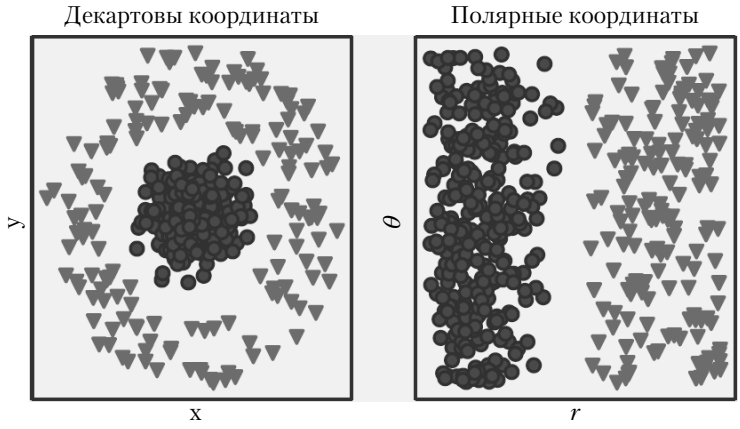

Рис. 1.1 ❖ Пример различных представлений: предположим, что требуется разделить две категории данных, проведя прямую на диаграмме рассеяния. На левом рисунке данные представлены в декартовых координатах, и задача неразрешима. На правом рисунке те же данные представлены в полярных координатах и разделяются вертикальной прямой (рисунок подготовлен совместно с Дэвидом Уорд-Фарли)

Одно из решений этой проблемы – воспользоваться машинным обучением не только для того, чтобы найти отображение представления на результат, но и чтобы определить само представление. Такой подход называется **обучением представлений**. На представлениях, полученных в ходе обучения, часто удается добиться гораздо более высокого качества, чем на представлениях, созданных вручную. К тому же это позволяет системам ИИ быстро адаптироваться к новым задачам при минималь-

ном вмешательстве человека. Для простой задачи алгоритм обучения представлений может найти хороший набор признаков за несколько минут, для сложных – за время от нескольких часов до нескольких месяцев. Проектирование признаков вручную для сложной задачи требует много времени и труда, на это могут уйти десятилетия работы всего сообщества исследователей.

Квинтэссенцией алгоритма обучения представлений является **автокодировщик**. Это комбинация функции **кодирования**, которая преобразует входные данные в другое представление, и функции **декодирования**, которая преобразует новое представление в исходный формат. Обучение автокодировщиков устроено так, чтобы при кодировании и обратном декодировании сохранялось максимально много информации, но чтобы при этом новое представление обладало различными полезными свойствами. Различные автокодировщики ориентированы на получение различных свойств.

При проектировании признаков или алгоритмов обучения признаков нашей целью обычно является выделение **факторов вариативности**, которые объясняют наблюдаемые данные. В этом контексте слово «фактор» означает просто источник влияния, а не «сомножитель». Зачастую факторы – это величины, не наблюдаемые непосредственно. Это могут быть ненаблюдаемые объекты или силы в физическом мире, оказывающие влияние на наблюдаемые величины. Это могут быть также умозрительные конструкции, дающие полезные упрощающие объяснения или логически выведенные причины наблюдаемых данных. Их можно представлять себе как концепции или абстракции, помогающие извлечь смысл из данных, характеризуемых высокой вариативностью. В случае анализа записи речи к факторам вариативности относятся возраст и пол говорящего, акцент и произносимые слова. В случае анализа изображения автомобиля факторами вариативности являются положение машины, ее цвет, а также высота солнца над горизонтом и его яркость.

Источник трудностей в целом ряде практических приложений искусственного интеллекта – тот факт, что многие факторы вариативности оказывают влияние абсолютно на все данные, доступные нашему наблюдению. Отдельные пиксели изображения красного автомобиля ночью могут быть очень близки к черному цвету. Форма силуэта автомобиля зависит от угла зрения. В большинстве приложений требуется *разделить* факторы вариативности и отбросить те, что нам не интересны.

Разумеется, может оказаться очень трудно выделить такие высокоуровневые абстрактные признаки из исходных данных. Многие факторы вариативности, к примеру акцент говорящего, можно идентифицировать, только если наличествует очень глубокое, приближающееся к человеческому понимание природы данных. Но раз получить представление почти так же трудно, как решить исходную задачу, то, на первый взгляд, обучение представлений ничем не поможет.

Глубокое обучение решает эту центральную проблему обучения представлений, вводя представления, выражаемые в терминах других, более простых представлений. Глубокое обучение позволяет компьютеру строить сложные концепции из более простых. На рис. 1.2 показано, как в системе глубокого обучения можно представить концепцию изображения человека в виде комбинации более простых концепций – углов и контуров, – которые, в свою очередь, определены в терминах границ.

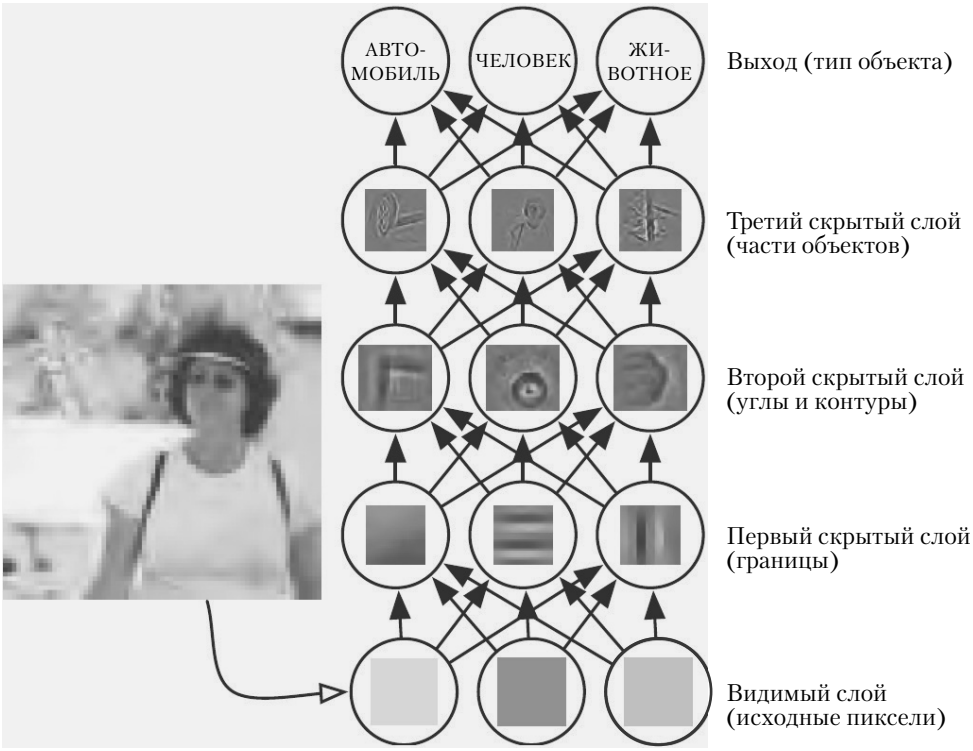

Рис. 1.2 ❖ Иллюстрация модели глубокого обучения. Компьютеру трудно понять смысл исходных данных, полученных от сенсоров, таких, например, как изображение, представленное в виде набора значений пикселей. Функция, отображающая множество пикселей в распознанный объект, очень сложна. Если подходить к задаче вычисления или обучения этого отображения в лоб, то она выглядит безнадежной. Глубокое обучение разрешает эту проблему, разбивая искомое сложное отображение на ряд более простых вложенных, каждое из которых описывается отдельным слоем модели. Входные данные представлены видимым слоем, он называется так, потому что содержит переменные, доступные наблюдению. За ним идет ряд скрытых слоев, которые извлекают из изображения все более и более абстрактные признаки. Слово «скрытый» означает, что значения, вырабатываемые этими слоями, не присутствуют в данных; сама модель должна определить, какие концепции полезны для объяснения связей в наблюдаемых данных. На рисунке показаны признаки, представленные каждым скрытым слоем. Зная исходные пиксели, первый слой легко может найти границы, для этого нужно лишь сравнить яркости соседних пикселей. Имея описание границ, выработанное первым скрытым слоем, второй скрытый слой находит углы и контуры в виде наборов границ. По этому описанию третий скрытый слой может распознать части конкретных объектов, представленные совокупностями контуров и углов определенного вида. Наконец, по описанию изображения в терминах частей объектов можно распознать сами объекты. Изображения взяты из работы Zeiler and Fergus (2014) с разрешения авторов

Типичным примером модели глубокого обучения является глубокая сеть прямого распространения, или **многослойный перцептрон** (МСП). Многослойный перцептрон – это просто математическая функция, отображающая множество входных значений на множество выходных. Эта функция является композицией нескольких более простых функций. Каждое применение одной математической функции можно рассматривать как новое представление входных данных.

Идея нахождения подходящего представления данных путем обучения – это лишь один взгляд на глубокое обучение. Другой взгляд состоит в том, что глубина позволяет обучать многошаговую компьютерную программу. Каждый слой представления можно мыслить себе как состояние памяти компьютера после параллельного выполнения очередного набора инструкций. Чем больше глубина сети, тем больше инструкций она может выполнить последовательно. Последовательное выполнение инструкций расширяет возможности, поскольку более поздние инструкции могут обращаться к результатам выполнения предыдущих.

При таком взгляде на глубокое обучение не всякая информация, используемая для активации слоев, обязательно кодирует факторы вариативности, объясняющие входные данные. В представлении хранится также вспомогательная информация о состоянии, помогающая выполнить программу, способную извлекать смысл из данных. Эту информацию можно уподобить счетчику или указателю в традиционной компьютерной программе. Она не имеет никакого отношения к содержанию входных данных, но помогает модели в организации их обработки.

Есть два основных способа измерить глубину модели. Первый оценивает архитектуру на основе числа последовательных инструкций, которые необходимо выполнить. Можно считать, что это длина самого длинного пути в графе, описывающем вычисление каждого выхода модели по ее входам. Как у двух эквивалентных компьютерных программ могут быть разные длины пути в зависимости от языка, на котором они написаны, так и одна и та же функциональность может быть изображена графами с разной длиной пути в зависимости от того, какие функции допускаются в качестве шагов. На рис. 1.3 показано, как выбор языка может дать разные результаты измерений для одной и той же архитектуры.

При другом подходе, используемом в глубоких вероятностных моделях, глубиной модели считается не глубина графа вычислений, а глубина графа, описывающего связи концепций. В этом случае граф вычислений, выполняемых для вычисления представления каждой концепции, может быть гораздо глубже, чем граф самих концепций. Связано это с тем, что понятие системы о простых концепциях можно уточнять, располагая информацией о более сложных. Например, система ИИ, наблюдающая изображение лица, на котором один глаз находится в тени, первоначально может распознать только один глаз. Но, обнаружив присутствие лица, система может заключить, что должен быть и второй глаз. В таком случае граф концепций содержит только два слоя – для глаз и для лиц, тогда как граф вычислений содержит $2n$ слоев, если мы n раз уточняем оценку каждой концепции при известной информации о второй.

Поскольку не всегда ясно, какой из двух подходов – глубина графа вычислений или глубина графа вероятностной модели – более релевантен, и поскольку разные

люди по-разному выбирают наборы примитивных элементов, из которых строятся графы, не существует единственно правильного значения глубины архитектуры, как не существует единственно правильной длины компьютерной программы. И нет общего мнения о том, какой должна быть глубина, чтобы модель можно было считать «глубокой». Однако можно все-таки сказать, что глубокое обучение – это наука о моделях, в которых уровень композиции обученных функций или обученных концепций выше, чем в традиционном машинном обучении.

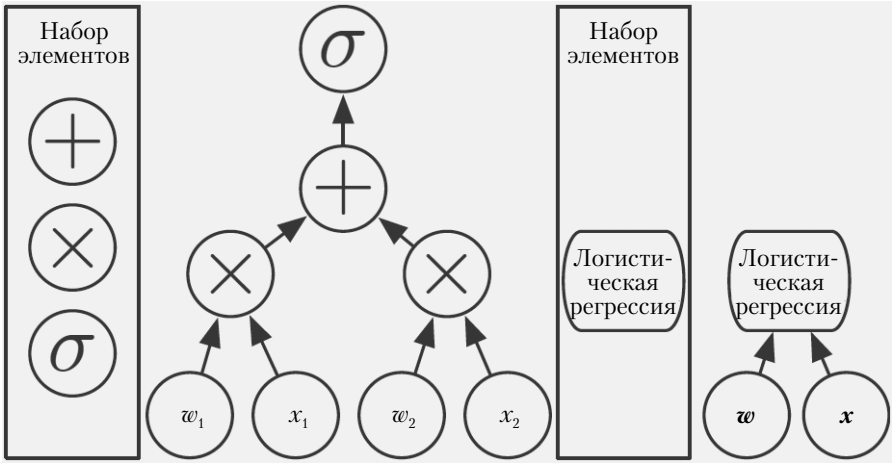

Рис. 1.3 ❖ Иллюстрация графов вычислений, переводящих вход в выход; в каждом узле выполняется некоторая операция. Глубиной считается длина самого длинного пути от входа к выходу, она зависит от определения допустимого шага вычисления. Оба графа описывают выход модели логистической регрессии $\sigma(\boldsymbol{w}^\top \boldsymbol{x})$, где σ – логистическая сигмоида. Если в качестве элементов языка используются сложение, умножение и логистические сигмоиды, то глубина модели равна 3. Если же логистическая регрессия сама считается элементом языка, то глубина равна 1

Итак, глубокое обучение – тема этой книги – один из подходов к ИИ. Конкретно, это вид машинного обучения – методики, которая позволяет компьютерной системе совершенствоваться по мере накопления опыта и данных. Мы твердо убеждены, что машинное обучение – единственный жизнеспособный подход к построению систем ИИ, которые могут функционировать в сложных окружающих условиях. Глубокое обучение – это частный случай машинного обучения, позволяющий достичь большей эффективности и гибкости за счет представления мира в виде иерархии вложенных концепций, в которой каждая концепция определяется в терминах более простых концепций, а более абстрактные представления вычисляются в терминах менее абстрактных. На рис. 1.4 показано соотношение между разными отраслями ИИ, а на рис. 1.5 – высокоуровневое описание каждого подхода.

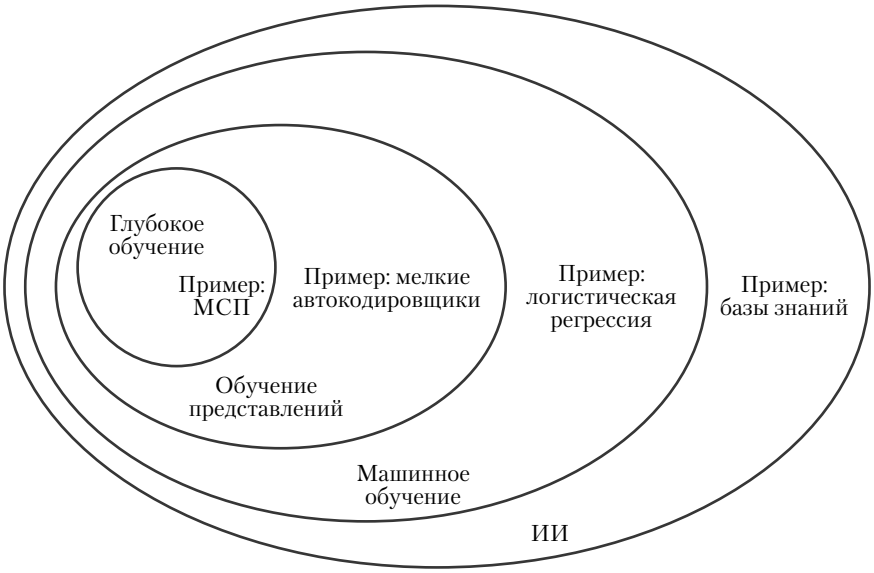

Рис. 1.4 ❖ На этой диаграмме Венна показано, что глубокое обучение – частный случай обучения представлений, которое, в свою очередь, является частным случаем машинного обучения, используемого во многих, но не во всех подходах к ИИ. На каждой части диаграммы Венна приведен пример технологии ИИ

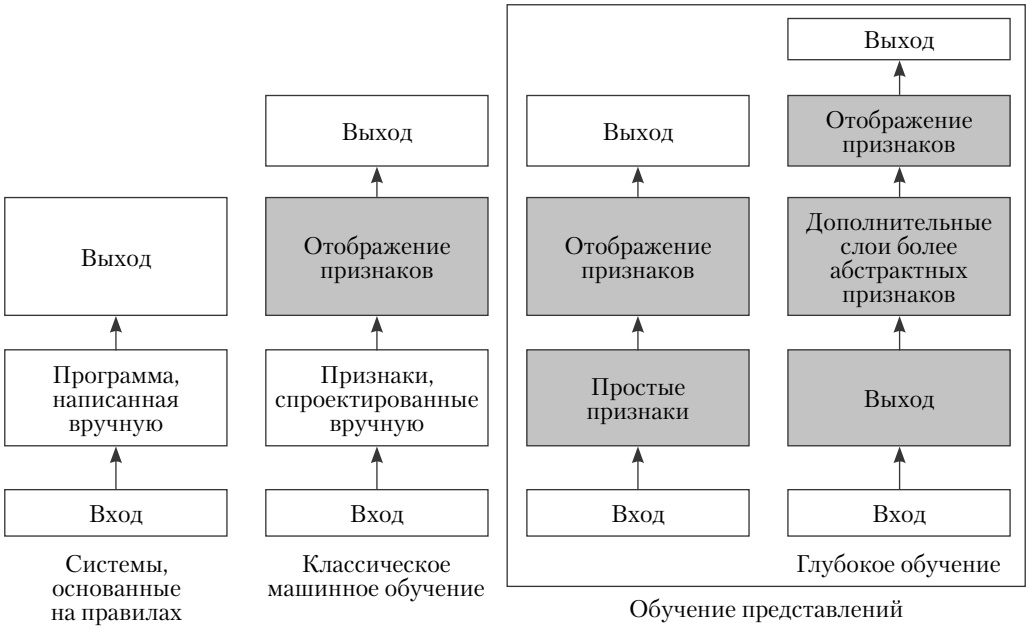

Рис. 1.5 ❖ Связь различных частей системы ИИ между собой в рамках разных подходов к ИИ. Серым цветом показаны компоненты, способные обучаться на данных

1.1. На кого ориентирована эта книга

Книга будет полезна разным читателям, но мы писали ее, имея в виду две целевые аудитории. Во-первых, это студенты университетов (как младших, так и старших курсов), изучающие машинное обучение, в том числе и те, кто начинает строить карьеру в области машинного обучения и искусственного интеллекта. Во-вторых, это разработчики ПО, которые не изучали машинное обучение или статистику, но хотят быстро освоить эти дисциплины и приступить к использованию глубокого обучения в своем продукте или платформе. Глубокое обучение доказало полезность во многих направлениях ПО, в т. ч. компьютерном зрении, распознавании речи и аудиозаписей, обработке естественных языков, робототехнике, биоинформатике и химии, видеоиграх, поисковых системах, интернет-рекламе и финансах.

Книга разделена на три части, чтобы лучше удовлетворить потребности разных категорий читателей. В части I излагается базовый математический аппарат и дается введение в концепции машинного обучения. В части II описаны самые известные алгоритмы глубокого обучения, которые можно считать сформировавшимися технологиями. Часть III посвящена более спорным идеям, которые многие считают важными для будущих исследований в области глубокого обучения.

Читатель может без ущерба для понимания пропускать части, не отвечающие его интересам или подготовке. Например, знакомые с линейной алгеброй, теорией вероятности и фундаментальными концепциями машинного обучения могут пропустить часть I, а желающим реализовать работающую систему нет необходимости читать дальше части II. Чтобы читателю было проще понять, какие главы ему интересны, на рис. 1.6 показано, как устроена книга в целом.

Мы предполагаем, что читатели имеют подготовку в области информатики: знакомы с программированием, понимают, что такое производительность вычислений и теория сложности, знают начала математического анализа и некоторые положения и термины теории графов.

1.2. Исторические тенденции в машинном обучении

Понять глубокое обучение проще всего в историческом контексте. Мы не станем детально описывать историю глубокого обучения, а выделим несколько ключевых тенденций:

- у глубокого обучения долгая и богатая история, но оно фигурировало под разными названиями, отражающими различные философские воззрения, в связи с чем его популярность то усиливалась, то ослабевала;
- полезность глубокого обучения возросла, когда увеличился объем доступных обучающих данных;
- размер моделей глубокого обучения увеличивался по мере того, как совершенствовалась компьютерная инфраструктура (программная и аппаратная);
- со временем с помощью глубокого обучения удавалось решать все более сложные задачи со все большей точностью.

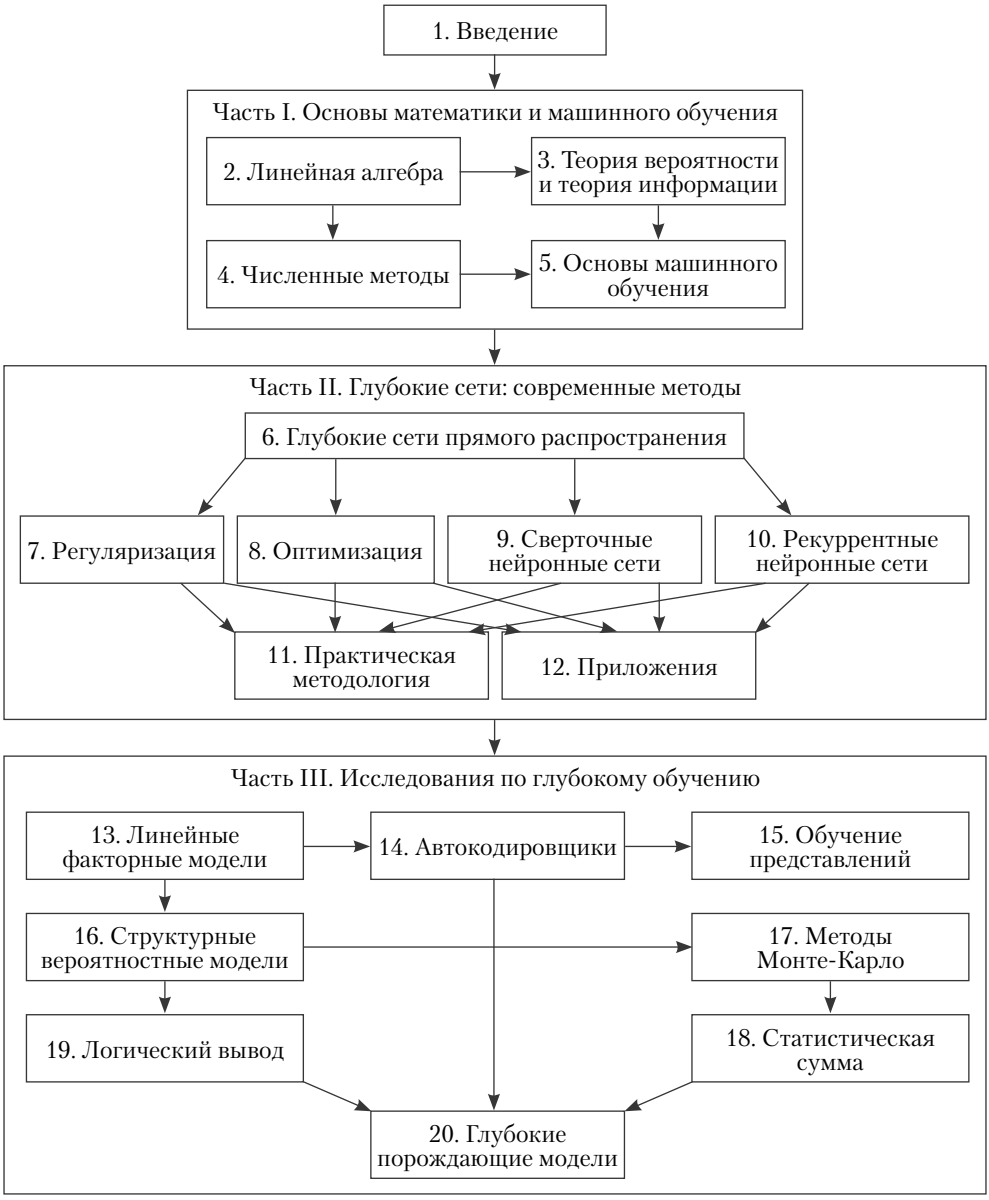

Рис. 1.6 ❖ Структура книги. Стрелки означают, что одна глава содержит материал, необходимый для понимания другой

1.2.1. Нейронные сети: разные названия и переменчивая фортуна

Мы полагаем, что многие читатели слышали о глубоком обучении как о сулящей чудеса новой технологии и удивлены, встретив слово «история» в применении к только зарождающейся дисциплине. Но в действительности глубокое обучение возникло еще в 1940-х годах. Оно кажется новым лишь потому, что в течение нескольких

лет, предшествующих нынешнему всплеску популярности, прозябало в тени, а также потому, что названия менялись, и только недавно эта дисциплина стала называться «глубоким обучением». Прежние названия отражали вес разных исследователей в научных кругах и разные точки зрения на предмет.

Изложение полной истории глубокого обучения выходит за рамки этого учебника. Но кое-какие базовые сведения помогут лучше понять его смысл. Если отвлечься от деталей, то было три волны разработок: в 1940–1960-х годах глубокое обучение было известно под названием **кибернетики**, в 1980–1990-х – как **коннекционизм**, а в современной инкарнации – под нынешним названием – оно возродилось в 2006 году. Количественная картина показана на рис. 1.7.

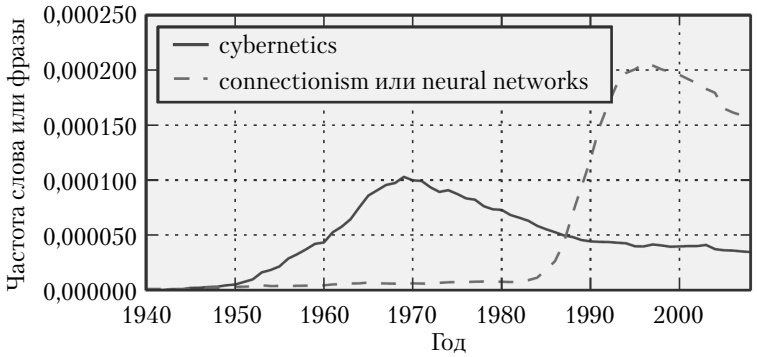

Рис. 1.7 ❖ Две из трех исторических волн исследований по искусственным нейронным сетям, оцененные по частоте фраз «cybernetics» и «connectionism или neural networks» согласно Google Books (третья волна началась недавно и еще не отражена). Первая волна, связанная с кибернетикой, приходится на 1940–1960-е годы, когда разрабатывались теории биологического обучения (McCulloch and Pitts, 1943; Hebb, 1949) и были реализованы первые модели, в частности перцептрон (Rosenblatt, 1958), позволявшие обучить один нейрон. Вторая волна периода 1980–1995 гг. связана с коннекционистским подходом, когда метод обратного распространения (Rumelhart et al., 1986a) был применен к обучению нейронной сети с одним или двумя скрытыми слоями. Третья волна, глубокое обучение, началась примерно в 2006 году (Hinton et al., 2006; Bengio et al., 2007; Ranzato et al., 2007a) и только теперь – в 2016 году – описывается в виде книги. Книги, посвященные двум другим волнам, также вышли гораздо позже активизации исследований в соответствующей области

Некоторые из самых ранних алгоритмов обучения сегодня мы назвали бы компьютерными моделями биологического обучения, т. е. процессов, которые происходят или могли бы происходить в мозге. Поэтому одно из прежних названий глубокого обучения – искусственные нейронные сети (ИНС). Им соответствует взгляд на модели глубокого обучения как на инженерные системы, устроенные по образцу биологического мозга (человека или животного). Но хотя были попытки использовать нейронные сети, применяемые в машинном обучении, чтобы понять, как функционирует мозг (Hinton and Shallice, 1991), в общем случае при их проектировании не ставилась задача создать реалистическую модель биологической функции. Нейронный подход к глубокому обучению основан на двух главных идеях. Во-первых, мозг – доказатель-

ный пример того, что разумное поведение возможно. Поэтому концептуально прямой путь к построению искусственного интеллекта состоит в том, чтобы проанализировать с вычислительной точки зрения принципы работы мозга и воспроизвести его функциональность. Во-вторых, вообще было бы очень интересно понять, как работает мозг и какие принципы лежат в основе разума человека, поэтому модели машинного обучения, проливающие свет на эти фундаментальные научные вопросы, полезны вне зависимости от их применимости к конкретным инженерным задачам.

Современный термин «глубокое обучение» выходит за рамки нейробиологического взгляда на модели машинного обучения. В нем заложен более общий принцип обучения нескольких уровней композиции, применимый к системам машинного обучения, не обязательно устроенным по примеру нейронов.

Предтечами современного глубокого обучения были простые линейные модели на основе нейробиологических аналогий. Они принимали множество n входных значений x_1, ..., x_n и ассоциировали с ними выход y. В ходе обучения модель должна была найти веса w_1, ..., w_n и вычислить выход в виде $f(\boldsymbol{x}, \boldsymbol{w}) = x_1 w_1 + ... + x_n w_n$. Эта первая волна нейронных сетей известна под названием кибернетики (рис. 1.7).

Нейрон Маккаллока–Питтса (McCulloch and Pitts, 1943) был ранней моделью функционирования мозга. Эта линейная модель могла распознавать две категории выходов, проверяя, является значение $f(\boldsymbol{x}, \boldsymbol{w})$ положительным или отрицательным. Конечно, чтобы модель соответствовала желаемому определению категорий, нужно было правильно подобрать веса. Веса задавал человек. В 1950-е годы был изобретен перцептрон (Rosenblatt, 1958, 1962) – первая модель, которая могла в процессе обучения находить веса, определяющие категории, имея примеры входных данных из каждой категории. Модель **адаптивного линейного элемента** (ADALINE), относящаяся примерно к тому же времени, просто возвращала само значение $f(\boldsymbol{x})$ для предсказания вещественного числа (Widrow and Hoff, 1960) и также могла обучаться предсказанию чисел на данных.

Эти простые алгоритмы обучения оказали заметное влияние на современный ландшафт машинного обучения. Алгоритм обучения, использованный для подбора весов в модели ADALINE, был частным случаем алгоритма **стохастического градиентного спуска**. Его немного модифицированные варианты и по сей день остаются основными алгоритмами моделей глубокого обучения.

Модели на основе функции $f(\boldsymbol{x}, \boldsymbol{w})$, используемые в перцептроне и ADALINE, называются **линейными моделями**. Они до сих пор относятся к числу наиболее распространенных моделей машинного обучения, хотя часто обучаются иначе, чем было принято раньше.

У линейных моделей много ограничений. Самое известное состоит в невозможности обучить функцию XOR, для которой $f([0, 1], \boldsymbol{w}) = 1$ и $f([1, 0], \boldsymbol{w}) = 1$, но $f([1, 1], \boldsymbol{w}) = 0$ и $f([0, 0], \boldsymbol{w})$. Критики, отмечавшие эти изъяны линейных моделей, вообще возражали против обучения, основанного на биологических аналогиях (Minsky and Papert, 1969). Это стало первым серьезным ударом по популярности нейронных сетей. В настоящее время нейробиология рассматривается как важный источник идей для исследований в области глубокого обучения, но уже не занимает доминирующих позиций.

Главная причина снижения роли нейробиологии в исследованиях по глубокому обучению – тот факт, что у нас попросту не хватает информации о мозге, чтобы использовать ее в качестве образца и руководства к действию. Чтобы по-настоящему

понять алгоритмы работы мозга, нужно было бы одновременно наблюдать за активностью тысяч (как минимум) взаимосвязанных нейронов. Не имея такой возможности, мы далеки от понимания даже самых простых и хорошо изученных областей мозга (Olshausen and Field, 2005).

Нейробиология дала надежду на то, что один алгоритм глубокого обучения сможет решить много разных задач. Нейробиологи выяснили, что хорьки могут «видеть» с помощью области мозга, отвечающей за обработку слуховой информации, если перенаправить нервы из глаз в слуховую кору (Von Melchner et al., 2000). Это наводит на мысль, что значительная часть мозга млекопитающих, возможно, использует единый алгоритм для решения большинства задач. До появления этой гипотезы исследования по машинному обучению были более фрагментированы: обработкой естественных языков, компьютерным зрением, планированием движения и распознаванием речи занимались разные группы ученых. Эти сообщества и по сей день разделены, но ученые, работающие в области глубокого обучения, зачастую занимаются многими или даже всеми этими предметами.

Мы можем почерпнуть из нейробиологии кое-какие соображения. Основная идея, навеянная осмыслением работы мозга, – наличие большого числа вычислительных блоков, которые обретают разум только в результате взаимодействий. Неокогнитрон (Fukushima, 1980) предложил архитектуру модели, эффективную для обработки изображений. Идея сложилась под влиянием структуры зрительной системы млекопитающих и впоследствии легла в основу современной сверточной сети (LeCun et al., 1998b), с которой мы познакомимся в разделе 9.10. Большинство современных нейронных сетей основано на модели нейрона, которая называется **блоком линейной ректификации**. Оригинальная модель когнитрона (Fukushima, 1975) была более сложной, основанной на наших знаниях о работе мозга. В современной упрощенной модели объединены различные точки зрения; в работах Nair and Hinton (2010) и Glorot et al. (2011a) отмечено влияние нейробиологии, а в работе Jarrett et al. – скорее, инженерных дисциплин. Каким бы важным источником идей ни была нейробиология, считать, что от нее нельзя отклониться ни на шаг, вовсе необязательно. Мы знаем, что настоящие нейроны вычисляют совсем не те функции, что блоки линейной ректификации, но большее приближение к реальности пока не привело к повышению качества машинного обучения. Кроме того, хотя нейробиология легла в основу нескольких успешных архитектур нейронных сетей, мы до сих пор знаем о биологическом обучении недостаточно, чтобы полнее использовать нейробиологию для построения *алгоритмов обучения* этих архитектур.

В социальных сетях часто подчеркивают сходство между глубоким обучением и мозгом. Да, действительно, исследователи, занимающиеся глубоким обучением, чаще говорят о влиянии аналогий с мозгом на свою работу, чем те, кто занимается другими методами машинного обучения, например ядерными методами или байесовской статистикой. Но не следует думать, что цель глубокого обучения – попытка имитировать мозг. Современное глубокое обучение черпает идеи из разных дисциплин, особенно из таких отраслей прикладной математики, как линейная алгебра, теория вероятности, теория информации и численная оптимизация. Некоторые ученые считают нейробиологию важным источником идей, другие не упоминают ее вовсе.

Отметим, что попытки понять, как работает мозг на алгоритмическом уровне, не прекращаются. Эта область исследований, известная под названием «вычислительная нейробиология», не совпадает с глубоким обучением. Нередко ученые

переходят из одной области в другую. Предмет глубокого обучения – построение компьютерных систем, способных успешно решать задачи, требующие интеллекта, а предмет вычислительной нейробиологии – построение более точных моделей работы мозга.

В 1980-е годы поднялась вторая волна исследований по нейронным сетям, вызванная главным образом движением под названием **коннекционизм**, или **параллельная распределенная обработка** (Rumelhart et al., 1986c; McClelland et al., 1995). Коннекционизм возник в контексте когнитивистики – междисциплинарного подхода к пониманию процесса познания, объединяющего несколько разных уровней анализа. В начале 1980-х годов большинство когнитивистов изучало модели принятия решений путем манипулирования символами (symbolic reasoning). Несмотря на популярность символических моделей, трудно было объяснить, как мозг мог бы реализовать их с помощью нейронов. Коннекционисты начали изучать модели познания, которые допускали реализацию на основе нейронов (Touretzky and Minton, 1985), возродив многие идеи психолога Дональда Хебба, высказанные в 1940-х годах (Hebb, 1949).

Центральная идея коннекционизма состоит в том, что при наличии большого количества вычислительных блоков, объединенных в сеть, удастся достичь разумного поведения. Эта идея относится в равной мере к нейронам в биологических нервных системах и к скрытым блокам в компьютерных моделях.

Движение коннекционизма породило несколько ключевых концепций, которые и по сей день играют важнейшую роль в глубоком обучении.

Одна из них – **распределенное представление** (Hinton et al., 1986). Идея в том, что каждый вход системы следует представлять многими признаками, а каждый признак должен участвовать в представлении многих возможных входов. Пусть, например, имеется зрительная система, способная распознавать легковые автомобили, грузовики и птиц, причем объекты могут быть красного, зеленого или синего цвета. Один из способов представления таких входов – завести отдельный нейрон или скрытый блок для активации каждой из девяти возможных комбинаций: красный грузовик, красная легковушка, красная птица, зеленый грузовик и т. д. Тогда потребуется девять нейронов, и каждый нейрон необходимо независимо обучить концепциям цвета и типа объекта. Улучшить ситуацию можно, воспользовавшись распределенным представлением, в котором три нейрона описывают цвет, а еще три – тип объекта. Тогда понадобится всего шесть нейронов, и нейрон, отвечающий за красное, можно обучить на изображениях легковушек, грузовиков и птиц, а не только на изображениях объектов одного типа. Концепция распределенного представления является центральной в этой книге и подробно рассматривается в главе 15.

Еще одним крупным достижением коннекционистов стали успешное использование обратного распространения для обучения глубоких нейронных сетей с внутренними представлениями и популяризация алгоритма обратного распространения (Rumelhart et al., 1986a; LeCun, 1987). Его популярность то возрастала, то убывала, но на данный момент это преобладающий подход к обучению глубоких моделей.

1990-е годы стали временем важных достижений в моделировании последовательностей с помощью нейронных сетей. В работах Hochreiter (1991) и Bengio et al. (1994) сформулирован ряд фундаментальных математических трудностей моделирования длинных последовательностей (см. раздел 10.7). В работе Hochreiter and Schmidhuber (1997) введено понятие сетей с долгой краткосрочной памятью (long short-term me-

mory – LSTM) для разрешения некоторых из описанных трудностей. Сегодня LSTM-сети широко используются во многих задачах моделирования последовательностей, в т. ч. для обработки естественных языков в Google.

Вторая волна работ по нейронным сетям продолжалась до середины 1990-х годов. Но компании, специализирующиеся на нейронных сетях и других технологиях ИИ, стали давать чрезмерно амбициозные обещания в попытках привлечь инвестиции. Когда ИИ не оправдал этих неразумных надежд, инвесторы испытали разочарование. В то же время имел место заметный прогресс в других областях машинного обучения. Ядерные методы (Boser et al., 1992; Cortes and Vapnik, 1995; Schölkopf et al., 1999) и графические модели (Jordan, 1998) позволили достичь хороших результатов при решении многих важных задач. В совокупности эти два фактора привели к спаду интереса к нейронным сетям, который продолжался до 2007 года.

В это время нейронные сети по-прежнему показывали впечатляющее качество на некоторых задачах (LeCun et al., 1998b; Bengio et al., 2001). Канадский институт перспективных исследований (Canadian Institute for Advanced Research – CIFAR) помог нейронным сетям остаться на плаву, профинансировав исследовательскую программу нейронных вычислений и адаптивного восприятия (Neural Computation and Adaptive Perception – NCAP). В рамках этой программы объединились группы Джеффри Хинтона из Торонтского университета, Иошуа Бенджио из Монреальского университета и Янна Лекуна из Нью-Йоркского университета. Мультидисциплинарная исследовательская программа CIFAR NCAP включала также нейробиологов и специалистов по человеческому и компьютерному зрению.

Тогда сложилось общее мнение, что обучить глубокие сети очень трудно. Теперь мы знаем, что алгоритмы, существовавшие начиная с 1980-х годов, работают отлично, но это не было очевидно до 2006 года. Причина, наверное, в том, что с вычислительной точки зрения эти алгоритмы очень накладны, поэтому было невозможно экспериментировать с ними на имевшемся тогда оборудовании.

Третья волна работ по нейронным сетям началась с прорыва в 2006 году. Джеффри Хинтон показал, что так называемые **глубокие сети доверия** можно эффективно обучать с помощью стратегии жадного послойного предобучения (Hinton et al., 2006), которую мы подробно опишем в разделе 15.1. Другие аффилированные с CIFAR исследовательские группы быстро показали, что ту же стратегию можно использовать для обучения многих других видов глубоких сетей (Bengio et al., 2007; Ranzato et al., 2007a), и систематически улучшали степень обобщения на тестовых примерах. Благодаря этой волне исследований в обиход вошел термин «глубокое обучение», подчеркивающий, что теперь можно обучать более глубокие нейронные сети, чем раньше, и привлекающий внимание к теоретической важности глубины (Bengio and LeCun, 2007; Delalleau and Bengio, 2011; Pascanu et al., 2014a; Montufar et al., 2014). В то время глубокие нейронные сети превзошли конкурирующие системы ИИ – как основанные на других технологиях машинного обучения, так и спроектированные вручную. Третья волна популярности нейронных сетей продолжается и во время написания этой книги, хотя фокус исследований значительно сместился. Поначалу в центре внимания находились методы обучения без учителя и способность глубоких моделей, обученных на небольших наборах данных, к обобщению. А теперь больший интерес вызывают гораздо более старые алгоритмы обучения с учителем и возможность задействовать большие размеченные наборы данных при обучении глубоких моделей.

1.2.2. Увеличение размера набора данных

Может возникнуть вопрос, почему лишь недавно была осознана роль глубокого обучения как ключевой технологии, хотя первые эксперименты с искусственными нейронными сетями были проведены еще в 1950-х годах. Глубинное обучение успешно применяется в коммерческих приложениях, начиная с 1990-х, но часто его рассматривали не как технологию, а как искусство – как нечто такое, что подвластно только экспертам. Так было до недавнего времени. Действительно, чтобы добиться хорошего качества от алгоритма глубокого обучения, нужен некоторый навык. Но, к счастью, потребность в таком навыке снижается по мере увеличения объема обучающих данных. Алгоритмы обучения, по качеству приближающиеся к возможностям человека при решении сложных задач, остались почти такими же, как алгоритмы, с трудом справлявшиеся с игрушечными задачами в 1980-х, но модели, которые мы с их помощью обучаем, претерпели изменения, позволившие упростить обучение очень глубоких архитектур. Самое важное новшество заключается в том, что сегодня мы можем предоставить этим алгоритмам необходимые для успеха ресурсы. На рис. 1.8 показано изменение эталонных наборов данных со временем. Эта тенденция обусловлена возрастающей цифровизацией общества. Чем активнее применяются компьютеры, тем больше записей о том, что мы делаем. А поскольку компьютеры объединяются в сети, становится проще централизованно хранить эти записи и построить из них набор данных, подходящий для машинного обучения. С наступлением эры «больших данных» машинное обучение значительно упростилось, поскольку ключевая проблема статистического оценивания – высокое качество обобщения на новые данные после обучения на небольшом количестве примеров – теперь далеко не так актуальна. В 2016 году действует грубое эвристическое правило: алгоритм глубокого обучения с учителем достигает приемлемого качества при наличии примерно 5000 помеченных примеров на категорию и оказывается сопоставим или даже превосходит человека, если обучается на наборе данных, содержащем не менее 10 миллионов помеченных примеров. Как добиться успеха при работе с наборами данных меньшего размера – важная область исследований, и акцент в ней делается на том, чтобы воспользоваться преимуществами большого количества непомеченных примеров, применяя обучение без учителя или с частичным привлечением учителя.

1.2.3. Увеличение размера моделей

Еще одна причина нынешнего роста популярности нейронных сетей после сравнительно скромных успехов в 1980-е годы – наличие вычислительных ресурсов, достаточных для работы с гораздо более крупными моделями. Один из главных выводов коннекционизма состоит в том, что животные проявляют интеллект, когда много нейронов работает совместно. Один нейрон или небольшой их набор не принесет особой пользы.

Плотность соединений между биологическими нейронами не слишком велика. Как видно по рис. 1.10, в наших моделях машинного обучения число соединений в расчете на один нейрон примерно на порядок выше, чем даже в мозгу млекопитающих, и такое положение существует уже несколько десятков лет.

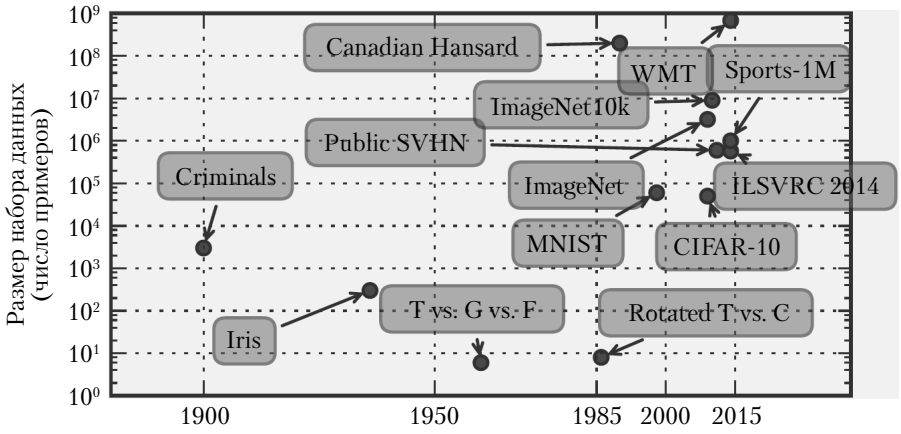

Рис. 1.8 ❖ Увеличение размера набора данных со временем. В начале 1900-х годов статистики изучали наборы данных, содержащие от сотен до тысяч вручную подготовленных измерений (Garson, 1900; Gosset, 1908; Anderson, 1935; Fisher, 1936). В период между 1950-ми и 1980-ми пионеры биокомпьютерного зрения зачастую работали с небольшими синтетическими наборами данных, например растровыми изображениями букв низкого разрешения, специально спроектированными так, чтобы снизить стоимость вычислений и продемонстрировать, что нейронные сети можно обучить функциям специального вида (Widrow and Hoff, 1960; Rumelhart et al., 1986b). В 1980-е и 1990-е машинное обучение стало в большей степени статистическим, а наборы данных уже насчитывали десятки тысяч примеров, как, например, набор MNIST (рис. 1.9) отсканированных рукописных цифр (LeCun et al., 1998b). В первом десятилетии XXI века продолжали создавать более изощренные наборы данных того же размера, например CIFAR-10 (Krizhevsky and Hinton, 2009). В конце этого периода и в первой половине 2010-х появление гораздо больших наборов данных, содержащих от сотен тысяч до десятков миллионов примеров, полностью изменило представление о возможностях глубокого обучения. К таким наборам относится общедоступный набор номеров домов (Street View House Numbers) (Netzer et al., 2011), различные варианты набора ImageNet (Deng et al., 2009, 2010a; Russakovsky et al., 2014a) и набор Sports-1M (Karpathy et al., 2014). В верхней части диаграммы мы видим, что наборы переведенных предложений, например набор, построенный IBM по официальным отчетам о заседаниях канадского парламента (Brown et al., 1990), и набор WMT 2014 переводов с английского на французский (Schwenk, 2014), по размеру намного превосходят большинство остальных наборов

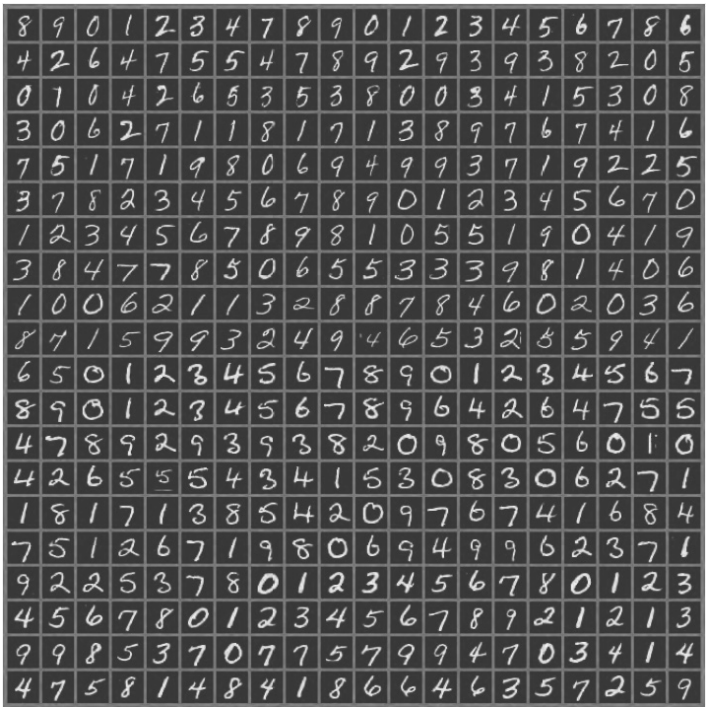

Рис. 1.9 ❖ Несколько примеров из набора данных MNIST. Акроним «NIST» означает «National Institute of Standards and Technology» (Национальный институт стандартов и технологий) – учреждение, первоначально собравшее эти данные. А буква «M» означает «модифицированный», поскольку данные были подвергнуты предварительной обработке, чтобы упростить применение алгоритмов машинного обучения. Набор данных MNIST содержит отсканированные изображения рукописных цифр и ассоциированные с ними метки, описывающие, какая цифра от 0 до 9 изображена. Это одна из самых простых задач классификации, поэтому набор широко используется для тестирования в исследованиях по глубокому обучению. Он сохраняет популярность, хотя для современных методов задача не представляет труда. Джеффри Хинтон назвал его «дрозофилой машинного обучения», имея в виду, что он позволяет специалистам по машинному обучению исследовать свои алгоритмы в контролируемых лабораторных условиях, как биологи изучают фруктовых мушек

Если говорить об общем числе нейронов, то до недавнего времени нейронные сети были на удивление малы (рис. 1.11). После добавления скрытых блоков размер искусственных нейронных сетей удваивался в среднем каждые 2,4 года, чему способствовало появление более быстрых компьютеров с большим объемом памяти, а также наличие крупных наборов данных. Чем больше сеть, тем выше ее точность на более сложных задачах. Эта тенденция прослеживается на протяжении десятилетий. Если темпы масштабирования не ускорятся в результате внедрения новых технологий, то искусственная нейронная сеть сравняется с человеческим мозгом по числу нейронов

не раньше 2050-х годов. Биологические нейроны могут представлять более сложные функции, чем современные искусственные, поэтому биологические нейронные сети могут оказаться даже больше, чем показано на диаграмме.

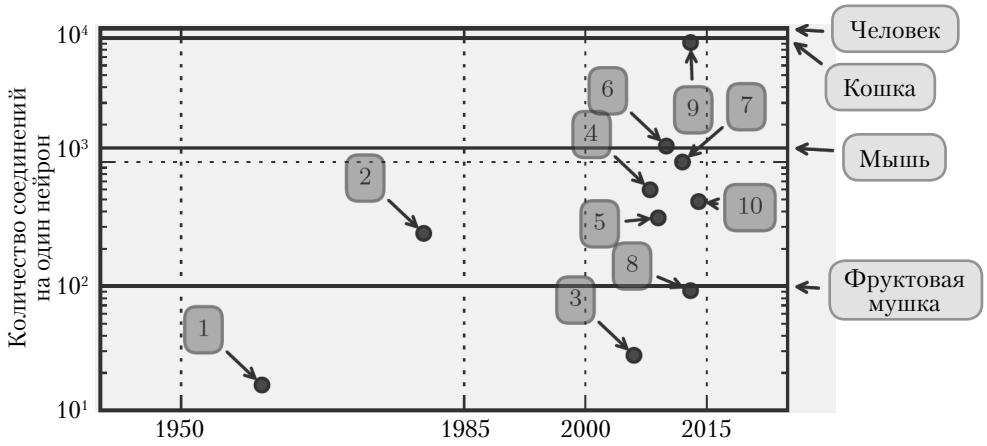

Рис. 1.10 ❖ Рост количества соединений в расчете на один нейрон со временем. Первоначально число соединений между нейронами в искусственной нейронной сети было ограничено возможностями оборудования. Сегодня оно в основном является параметром проектирования. В некоторых нейронных сетях число соединений почти такое же, как у кошки, а сети с числом соединений, как у мелких млекопитающих типа мыши, встречаются сплошь и рядом. Даже в мозге человека число соединений на нейрон нельзя назвать заоблачным. Размеры биологических нейронных сетей взяты из Википедии (2015)

1. Адаптивный линейный элемент (Widrow and Hoff, 1960)
2. Неокогнитрон (Fukushima, 1980)
3. GPU-ускоренная сверточная сеть (Chellapilla et al., 2006)
4. Глубокая машина Больцмана (Salakhutdinov and Hinton, 2009a)
5. Сверточная сеть без учителя (Jarrett et al., 2009)
6. GPU-ускоренный многослойный перцептрон (Ciresan et al., 2010)
7. Распределенный автокодировщик (Le et al., 2012)
8. Сверточная сеть с несколькими GPU (Krizhevsky et al., 2012)
9. Сверточная сеть без учителя на компьютерах типа COTS HPC (Coates et al., 2013)
10. GoogLeNet (Szegedy et al., 2014a)

В ретроспективе не кажется удивительным, что нейронные сети с числом нейронов меньше, чем у пиявки, не могли справиться с решением сложных задач искусственного интеллекта. Даже нынешние сети, которые мы считаем очень большими с вычислительной точки зрения, меньше нервной системы сравнительно примитивных позвоночных животных, например лягушек.

Увеличение размера моделей вследствие доступности более быстрых процессоров, GPU общего назначения (см. раздел 12.1.2), более быстрых сетей и более совершенной инфраструктуры распределенных вычислений – одна из самых важных тенденций в истории глубокого обучения. Ожидается, что она продолжится и в будущем.

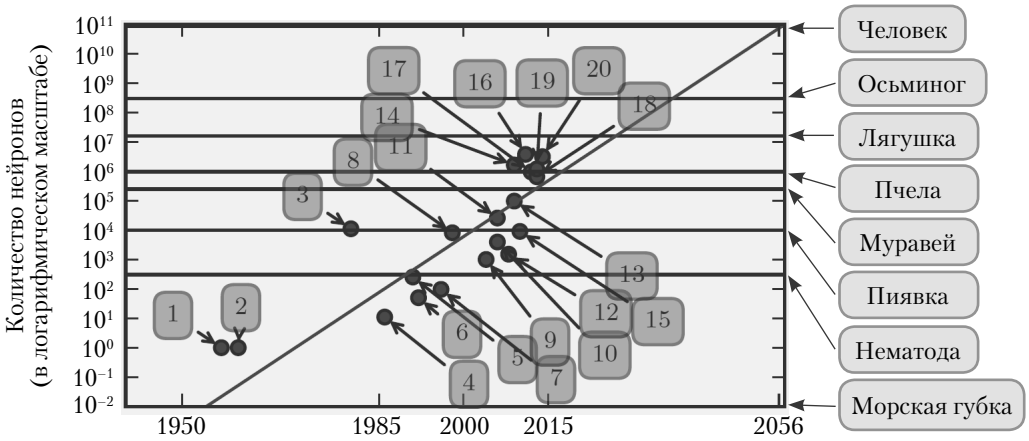

Рис. 1.11 ❖ Рост размера нейронной сети со временем. После добавления скрытых блоков размер нейронных сетей удваивался примерно каждые 2,4 года. Размеры биологических нейронных сетей взяты из Википедии (2015)

1. Перцептрон (Rosenblatt, 1958, 1962)
2. Адаптивный линейный элемент (Widrow and Hoff, 1960)
3. Неокогнитрон (Fukushima, 1980)
4. Ранняя сеть с обратным распространением (Rumelhart et al., 1986b)
5. Рекуррентная нейронная сеть для распознавания речи (Robinson and Fallside, 1991)
6. Многослойный перцептрон для распознавания речи (Bengio et al., 1991)
7. Сигмоидальная сеть доверия со средним полем (Saul et al., 1996)
8. LeNet-5 (LeCun et al., 1998b)
9. Нейронная эхо-сеть (Jaeger and Haas, 2004)
10. Глубокая сеть доверия (Hinton et al., 2006)
11. GPU-ускоренная сверточная сеть (Chellapilla et al., 2006)
12. Глубокая машина Больцмана (Salakhutdinov and Hinton, 2009a)
13. GPU-ускоренная сеть глубокого доверия (Raina et al., 2009)
14. Сверточная сеть без учителя (Jarrett et al., 2009)
15. GPU-ускоренный многослойный перцептрон (Ciresan et al., 2010)
16. Сеть OMP-1 (Coates and Ng, 2011)
17. Распределенный автокодировщик (Le et al., 2012)
18. Сверточная сеть с несколькими GPU (Krizhevsky et al., 2012)
19. Сверточная сеть без учителя на компьютерах типа COTS HPC (Coates et al., 2013)
20. GoogLeNet (Szegedy et al., 2014a)

1.2.4. Повышение точности и сложности и расширение круга задач

Начиная с 1980-х годов точность распознавания и прогнозирования глубоких моделей постоянно росла. И вместе с тем все разнообразнее становились задачи, которые удавалось решать с их помощью.

Самые первые глубокие модели использовались для распознавания отдельных объектов в кадрированных изображениях совсем небольшого размера (Rumelhart et al., 1986a). С тех пор размер изображений, которые можно было обработать с помощью нейронной сети, постепенно увеличивался. Современные сети распознавания объектов обрабатывают фотографии с высоким разрешением и не требуют кадрирования фотографии по месту расположения объекта (Krizhevsky et al., 2012). Кроме того, ранние сети умели распознавать только два вида объектов (а в некоторых случаях при-

сутствие или отсутствие объектов одного вида), тогда как типичная современная сеть распознает не менее 1000 категорий объектов. Самый крупный конкурс по распознаванию объектов – ImageNet Large Scale Visual Recognition Challenge (ILSVRC) – проводится каждый год. Переломным моментом, ознаменовавшим стремительный взлет глубокого обучения, стала победа с большим отрывом сверточной сети, которая участвовала впервые и сразу уменьшила частоту непопадания в первые пять (top-5 error rate) с 26,1 до 15,3% (Krizhevsky et al., 2012). Смысл этого показателя следующий: сверточная сеть порождала для каждого изображения ранжированный список возможных категорий, и правильная категория отсутствовала среди первых пяти элементов этого списка только в 15,3% тестовых примеров. С тех пор подобные конкурсы неизменно выигрывали сверточные сети, и на данный момент прогресс глубокого обучения позволил довести частоту непопадания в первые пять до 3,6% (рис. 1.12).

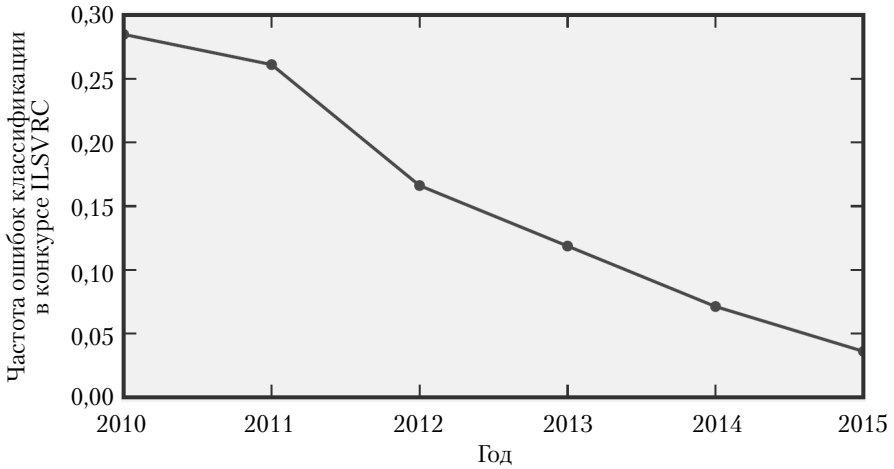

Рис. 1.12 ❖ Уменьшение частоты ошибок со временем. С тех пор как глубокие сети достигли масштаба, необходимого для участия в конкурсе ImageNet Large Scale Visual Recognition Challenge, они неизменно выигрывают его, с каждым разом демонстрируя все меньшую и меньшую частоту ошибок. Данные взяты из работ Russakovsky et al. (2014b) и He et al. (2015)

Глубокое обучение также оказало огромное влияние на распознавание речи. После прогресса, достигнутого на протяжении 1990-х годов, в качестве распознавания речи наступил застой. Применение глубокого обучения (Dahl et al., 2010; Deng et al., 2010b; Seide et al., 2011; Hinton et al., 2012a) привело к резкому уменьшению частоты ошибок, иногда аж наполовину. Мы вернемся к этой теме в разделе 12.3.

Глубокие сети добились также впечатляющих успехов в обнаружении пешеходов и сегментации изображений (Sermanet et al., 2013; Farabet et al., 2013; Couprie et al., 2013), а в задаче классификации дорожных знаков показали себя лучше человека (Ciresan et al., 2012).

Одновременно с повышением размера и точности глубоких сетей росла и сложность решаемых с их помощью задач. В работе Goodfellow et al. (2014d) показано, что нейронные сети можно научить распознаванию целых последовательностей символов в изображении, а не только идентификации одиночного объекта. Ранее считалось, что для такого обучения необходимо помечать отдельные элементы последовательности (Gülçehre

and Bengio, 2013). Рекуррентные нейронные сети, в частности вышеупомянутая модель LSTM, теперь применяются для моделирования связей *последовательностей* с другими *последовательностями*, а не только с фиксированными входами. Такое обучение типа «последовательность в последовательность», похоже, привело к революции в другом приложении: машинном переводе (Sutskever et al., 2014; Bahdanau et al., 2015).

Эта тенденция к увеличению сложности была доведена до логического завершения с вводом в рассмотрение нейронных машин Тьюринга (Graves et al., 2014a), которые обучаются читать из ячеек памяти и писать произвольные данные в ячейки памяти. Такие нейронные сети могут обучаться простым программам на примерах желаемого поведения. Например, они могут научиться сортировать списки чисел, если им предъявить примеры отсортированных и неотсортированных последовательностей. Эта технология самопрограммирования пока находится в зачаточной стадии, но в будущем теоретически может быть применена почти к любой задаче.

Еще одним венцом глубокого обучения является обобщение на **обучение с подкреплением**. В этом контексте автономный агент должен научиться выполнять некоторое задание методом проб и ошибок, без какой-либо подсказки со стороны человека. Компания DeepMind продемонстрировала систему обучения с подкреплением, основанную на технологиях глубокого обучения, которая способна научиться играть в видеоигры Atari, достигая во многих случаях уровня, сравнимого с человеческим (Mnih et al., 2015). Глубокое обучение позволило также значительно усовершенствовать качество обучения с подкреплением в робототехнике (Finn et al., 2015).

Многие приложения глубокого обучения приносят солидную прибыль. Эти технологии используются в таких крупных компаниях, как Google, Microsoft, Facebook, IBM, Baidu, Apple, Adobe, Netflix, NVIDIA и NEC.

Прогресс в глубоком обучении сильно зависит от достижений в области программной инфраструктуры. Библиотеки Theano (Bergstra et al., 2010; Bastien et al., 2012), PyLearn2 (Goodfellow et al., 2013c), Torch (Collobert et al., 2011b), DistBelief (Dean et al., 2012), Caffe (Jia, 2013), MXNet (Chen et al., 2015) и TensorFlow (Abadi et al., 2015) применяются в важных исследовательских проектах и в коммерческих продуктах.

Глубокое обучение внесло вклад и в другие науки. Современные сверточные сети для распознавания объектов дают модель визуальной обработки, изучаемую в нейробиологии (DiCarlo, 2013). Кроме того, глубокое обучение предоставляет полезные инструменты для обработки больших массивов данных и прогнозирования в различных научных дисциплинах. Оно было успешно применено к прогнозированию взаимодействия молекул в интересах фармацевтических компаний, занимающихся поиском новых лекарств (Dahl et al., 2014), к поиску субатомных частиц (Baldi et al., 2014) и к автоматическому распознаванию сделанных микроскопом изображений для построения трехмерной карты человеческого мозга (Knowles-Barley et al., 2014). В будущем мы ожидаем проникновения глубокого обучения во все новые и новые отрасли науки.

Резюмируя, можно сказать, что глубокое обучение – это развивавшийся в течение нескольких десятилетий подход к машинному обучению, основанный главным образом на наших знаниях о человеческом мозге, на статистике и прикладной математике. В последние годы глубокое обучение переживает стремительный рост популярности и полезности благодаря в первую очередь появлению более мощных компьютеров, больших наборов данных и методов обучения более глубоких сетей. Будущее сулит нам новые проблемы и возможности дальнейшего совершенствования глубокого обучения с выходом на новые рубежи.