

Оглавление

Предисловие	12
Благодарности	14
О книге	16
Кому адресована книга	16
Структура	16
Соглашения об оформлении кода и его загрузке	18
Форум книги	18
Об авторе	19
От издательства	19
1 Введение в глубокое обучение: зачем его изучать	20
Добро пожаловать в «Грокаем глубокое обучение»!	21
Почему вам стоит изучать глубокое обучение	21
Этому трудно учиться?	22
Почему вы должны прочитать эту книгу	23
Что нужно для начала	25
Возможно, вам потребуется знание Python	26
Итоги	26

2 Основные понятия: как учатся машины?	27
Что такое глубокое обучение?	28
Что такое машинное обучение?	29
Машинное обучение с учителем	30
Машинное обучение без учителя	31
Параметрическое и непараметрическое обучение	32
Параметрическое обучение с учителем	33
Параметрическое обучение без учителя	35
Непараметрическое обучение	37
Итоги	38
3 Введение в нейронное прогнозирование: прямое распространение	39
Шаг 1: прогнозирование	40
Простая нейронная сеть, делающая прогноз	42
Что такое нейронная сеть?	43
Что делает эта нейронная сеть?	44
Прогнозирование с несколькими входами	47
Несколько входов: что делает эта нейронная сеть?	49
Несколько входов: полный выполняемый код	54
Прогнозирование с несколькими выходами	56
Прогнозирование с несколькими входами и выходами	58
Несколько входов и выходов: как это работает?	60
Прогнозирование на основе прогнозов	62
Короткий пример использования NumPy	64
Итоги	67
4 Введение в нейронное обучение: градиентный спуск	69
Предсказание, сравнение и обучение	70
Сравнение	70
Обучение	71
Сравнение: способны ли нейронные сети делать точные прогнозы?	71
Зачем измерять ошибку?	72
Как выглядит простейшая форма нейронного обучения?	74
Обучение методом «холодно/горячо»	76
Особенности обучения методом «холодно/горячо»	77
Вычисление направления и величины из ошибки	79
Одна итерация градиентного спуска	81
Обучение просто уменьшает ошибку	83
Рассмотрим несколько циклов обучения	86
Как это работает? Что такое <code>weight_delta</code> на самом деле?	88
Узкий взгляд на одно понятие	90

Коробка со стержнями	91
Производные: второй пример	92
Что действительно необходимо знать	94
Что знать необязательно	94
Как использовать производные для обучения	95
Выглядит знакомо?	97
Ломаем градиентный спуск	98
Визуальное представление избыточной коррекции	99
Расхождение	100
Знакомьтесь: альфа-коэффициент	101
Альфа-коэффициент в коде	102
Запоминание	103

5 Корректировка сразу нескольких весов: обобщение градиентного спуска 104

Обучение методом градиентного спуска с несколькими входами	105
Градиентный спуск с несколькими входами, описание	107
Рассмотрим несколько шагов обучения	113
Замораживание одного веса: для чего?	115
Обучение методом градиентного спуска с несколькими выходами	117
Обучение методом градиентного спуска с несколькими входами и выходами . .	120
Чему обучаются эти веса?	121
Визуализация значений весов	124
Визуализация скалярных произведений (сумм весов)	125
Итоги	126

6 Создание первой глубокой нейронной сети: введение в обратное распространение 127

Задача о светофоре	128
Подготовка данных	130
Матрицы и матричные отношения	131
Создание матриц в Python	134
Создание нейронной сети	135
Обучение на полном наборе данных	137
Полный, пакетный и стохастический градиентный спуск	138
Нейронные сети изучают корреляцию	139
Повышающее и понижающее давление	140
Пограничный случай: переобучение	142
Пограничный случай: конфликт давлений	143
Определение косвенной корреляции	145
Создание корреляции	146
Объединение нейронных сетей в стек: обзор	147

Обратное распространение: определение причин ошибок на расстоянии	148
Обратное распространение: как это работает?	150
Линейность и нелинейность	151
Почему составная нейронная сеть не работает	152
Тайна эпизодической корреляции	153
Короткий перерыв	154
Ваша первая глубокая нейронная сеть	155
Обратное распространение в коде	156
Одна итерация обратного распространения	159
Объединяем все вместе	161
Почему глубокие сети важны для нас?	162
7 Как изобразить нейронную сеть: в голове и на бумаге	164
Время упрощать	165
Обобщение корреляции	166
Прежняя усложненная визуализация	167
Упрощенная визуализация	169
Еще более упрощенная визуализация	170
Посмотрим, как эта сеть получает прогноз	171
Визуализация с использованием букв вместо картинок	173
Связывание переменных	174
Сравнение разных способов визуализации	175
Важность инструментов визуализации	175
8 Усиление сигнала и игнорирование шума: введение в регуляризацию и группировку	177
Трехслойная сеть для классификации набора данных MNIST	178
Это было просто	180
Запоминание и обобщение	181
Переобучение нейронных сетей	182
Причины переобучения	184
Простейшая регуляризация: ранняя остановка	185
Стандартный способ регуляризации: прореживание (дропаут)	186
Как работает прореживание: в работе участвуют ансамбли	187
Прореживание в коде	188
Влияние прореживания на модель MNIST	191
Пакетный градиентный спуск	192
Итоги	194
9 Моделирование случайности и нелинейности: функции активации . . .	195
Что такое функция активации?	196
Стандартные функции активации для скрытых слоев	200

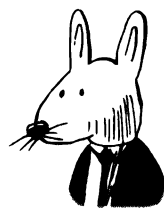
Стандартные функции активации для выходного слоя	201
Главная проблема: входные данные могут быть схожи между собой	204
Вычисление softmax	205
Инструкции по внедрению функций активации	207
Умножение разности на производную	209
Преобразование выхода в наклон (производную)	211
Усовершенствование сети MNIST	212
10 Края и углы нейронного обучения: введение в сверточные нейронные сети	215
Повторное использование весов в нескольких местах	216
Сверточный слой	217
Простая реализация в NumPy	220
Итоги	224
11 Нейронные сети, понимающие человеческий язык: король – мужчина + женщина == ?	226
Что значит понимать человеческий язык?	227
Обработка естественного языка (NLP)	228
Обработка естественного языка с учителем	229
Набор данных IMDB с обзорами фильмов	230
Выявление корреляции слов во входных данных	231
Прогнозирование обзоров фильмов	232
Введение в слой с векторным представлением	234
Интерпретация результата	236
Нейронная архитектура	237
Сравнение векторных представлений слов	240
В чем заключается смысл нейрона?	241
Подстановка пропущенных слов	242
Смысл определяется потерями	244
Король – мужчина + женщина $\approx$ королева	248
Словесные аналогии	249
Итоги	251
12 Нейронные сети, которые пишут как Шекспир: рекуррентные слои для данных переменной длины	252
Проблема произвольной длины	253
Действительно ли сравнение имеет значение?	254
Удивительная мощь усредненных векторов слов	255
Как векторные представления хранят информацию?	257
Как нейронная сеть использует векторные представления?	258
Ограничение векторов в модели «мешок слов»	259

Объединение векторных представлений слов с использованием единичной матрицы	261
Матрицы, которые ничего не меняют	262
Определение переходных матриц	264
Обучение созданию векторов предложений	265
Прямое распространение на Python	266
Как добавить сюда обратное распространение?	267
Обучим ее!	268
Подготовка	269
Прямое распространение с данными произвольной длины	271
Обратное распространение с данными произвольной длины	272
Корректировка весов с данными произвольной длины	273
Запуск и анализ результатов	274
Итоги	277

13 Введение в автоматическую оптимизацию: создание фреймворка глубокого обучения 278

Что такое фреймворк глубокого обучения?	279
Введение в тензоры	280
Введение в автоматическое вычисление градиента (autograd)	281
Контрольная точка	283
Тензоры, используемые многократно	284
Добавление поддержки тензоров многократного использования в реализацию autograd	286
Как работает сложение в обратном распространении?	288
Добавление поддержки отрицания	289
Добавление поддержки других операций	290
Использование autograd в обучении нейронной сети	295
Добавление автоматической оптимизации	297
Добавление поддержки слоев разных типов	298
Слои, содержащие другие слои	299
Слои с функцией потерь	300
Как научиться пользоваться фреймворком	301
Нелинейные слои	302
Слой с векторным представлением	304
Добавление индексирования в autograd	305
Слой с векторным представлением (повтор)	306
Слой с перекрестной энтропией	307
Рекуррентный слой	309
Итоги	313

14 Обучаем сеть писать как Шекспир: долгая краткосрочная память	314
Моделирование языка символов	315
Необходимо усеченное обратное распространение	316
Усеченное обратное распространение	317
Образец вывода	321
Затухающие и взрывные градиенты	322
Упрощенный пример обратного распространения в RNN	323
Ячейки долгой краткосрочной памяти (LSTM)	324
Аналогия, помогающая понять идею вентилях LSTM	325
Слой долгой краткосрочной памяти	326
Усовершенствование модели языка символов	328
Обучение LSTM-модели языка символов	329
Настройка LSTM-модели языка символов	330
Итоги	331
15 Глубокое обучение на конфиденциальных данных: введение в федеративное обучение	332
Проблема конфиденциальности в глубоком обучении	333
Федеративное обучение	334
Обучаем выявлять спам	335
Сделаем модель федеративной	337
Взламываем федеративную модель	338
Безопасное агрегирование	340
Гомоморфное шифрование	341
Федеративное обучение с гомоморфным шифрованием	342
Итоги	343
16 Куда пойти дальше: краткий путеводитель	345
Поздравляю!	346
Шаг 1: начните изучать PyTorch	346
Шаг 2: начните изучать следующий курс по глубокому обучению	347
Шаг 3: купите учебник по математике глубокого обучения	347
Шаг 4: заведите блог и рассказывайте в нем о глубоком обучении	348
Шаг 5: Twitter	349
Шаг 6: напишите руководство на основе академической статьи	350
Шаг 7: получите доступ к GPU	350
Шаг 8: найдите оплачиваемую работу, связанную с глубоким обучением	351
Шаг 9: присоединитесь к открытому проекту	351
Шаг 10: ищите единомышленников	352



Предисловие

«Грокаем глубокое обучение» — это результат трехлетнего напряженного труда. Чтобы создать книгу, которую вы держите в руках, мне пришлось написать вдвое больше страниц, чем вы видите. Полдесятка глав три или четыре раза были переписаны заново, и только после этого я решил, что они готовы к публикации. Кроме этого, попутно были добавлены новые важные главы, отсутствовавшие в изначальном плане.

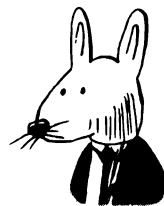
Что еще более важно, я в самом начале принял два решения, делающие мою книгу особенно ценной: эта книга не требует от читателя специальной математической подготовки, кроме знания основ арифметики, и не опирается на высокоуровневые библиотеки, которые могут скрывать происходящее в коде. Иначе говоря, любой сможет прочитать эту книгу и понять, как в действительности работает глубокое обучение. Для этого мне пришлось придумать новые способы описания и разъяснения основных идей и приемов, не прибегая к сложному математическому аппарату или замысловатому программному коду, написанному кем-то другим.

Работая над «Грокаем глубокое обучение», я преследовал цель максимально уменьшить порог входа в практику глубокого обучения. Вы не просто прочитаете теорию — вы откроете ее заново. А чтобы помочь вам в этом, я написал много кода и постарался выстроить объяснения в правильном порядке, чтобы фрагменты кода, необходимые для демонстрации, не были лишены смысла.

Эти знания, в сочетании с теорией, кодом и примерами, которые вы будете изучать в книге, помогут вам намного быстрее выполнять все наши эксперименты. Вы быстро добьетесь успеха и расширите свои практические навыки, а кроме того, вам проще будет освоить более сложные понятия глубокого обучения.

В последние три года я не только писал эту книгу, но еще и поступил в аспирантуру Оксфордского университета, присоединился к команде Google и стал одним из инициаторов проекта OpenMined — децентрализованной платформы искусственного интеллекта. Эта книга является кульминацией многих лет размышлений, обучения и преподавания.

Есть много источников знаний о глубоком обучении. И я рад, что вы выбрали именно этот.



Благодарности

Я чрезвычайно благодарен всем, кто принял участие в работе над книгой. Прежде всего я хочу поблагодарить удивительную команду издательства Manning: Берта Бейтса (Bert Bates), научившего меня писать; Кристину Тейлор (Christina Taylor), терпеливо поддерживавшую меня в течение трех лет; Майкла Стивенса (Michael Stephens), чье творческое мышление позволило книге стать успешной еще до публикации; Марьяна Бейса (Marjan Vase), чья поддержка была определяющей во время задержек.

«Грожаем глубокое обучение» не получилась бы такой, какой получилась, без значительного вклада первых ее читателей, приславших отзывы по электронной почте, в Twitter и GitHub. Я очень признателен Яше Суишеру (Jascha Swisher), Варуну Судхакару (Varun Sudhakar), Франсуа Шолле (Francois Chollet), Фредерику Виторино (Frederico Vitorino), Коди Хаммонду (Cody Hammond), Маурисио Марото Арриете (Mauricio Maroto Arrieta), Александару Драгосавлевицу (Aleksandar Dragosavljevic), Алану Картеру (Alan Carter), Френку Хинеку (Frank Hinek), Николасу Бенджамину Хокеру (Nicolas Benjamin Hocker), Хенку Мейссе (Hank Meisse), Уотеру Хибме (Wouter Hibma), Йоргу Розенкранцу (Joerg Rosenkranz), Алексу Виейре (Alex Vieira) и Чарли Харрингтону (Charlie Harrington) за их помощь в улучшении текста и кода в онлайн-репозитории.

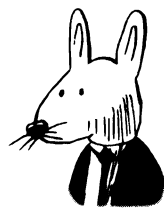
Хочу также поблагодарить рецензентов, выкроивших время, чтобы прочитать рукопись на разных этапах работы: Александра А. Мыльцева (Alexander

А. Myltsev), Амита Ламба (Amit Lamba), Ананда Саха (Anand Saha), Эндрю Хамора (Andrew Hamor), Кристиана Барриентоса (Cristian Barrientos), Монтойя (Montoya), Еремея Валетова (Eremey Valetov), Джеральда Мака (Gerald Mack), Яна Стирка (Ian Stirk), Каляна Редди (Kalyan Reddy), Камала Раджа (Kamal Raj), Кельвина Д. Микса (Kelvin D. Meeks), Марко Пауло душ Сантуш Ногейра (Marco Paulo dos Santos Nogueira), Мартина Бира (Martin Beer), Массимо Иларио (Massimo Ilario), Ненси У. Греди (Nancy W. Grady), Питера Хемптона (Peter Hampton), Себастьяна Мальдонада (Sebastian Maldonado), Шашанка Гупты (Shashank Gupta), Тимотеуша Воложко (Tymoteusz Wołodźko), Кумара Унникришнана (Kumar Unnikrishnan), Випула Гупты (Vipul Gupta), Уилла Фугера (Will Fuger) и Уильяма Уилера (William Wheeler).

Я чрезвычайно благодарен Мэту (Mat) и Нико (Niko) из Udacity, которые включили книгу в свой курс обучения Deep Learning Nanodegree, что очень помогло популяризации книги среди молодых специалистов по глубокому обучению.

Я должен поблагодарить доктора Уильяма Хупера (Dr. William Hooper), позволившего мне зайти в его кабинет и поспорить по вопросам информатики, сделавшего исключение и давшего мне возможность попасть на его курс (где уже не было мест) по программированию и вдохновившего меня на карьеру в области глубокого обучения. Я чрезвычайно благодарен за терпение, проявленное ко мне с самого начала. Вы были безмерно щедры ко мне.

Наконец, я хочу сказать спасибо моей жене за то, что терпела, когда я ночи напролет работал над книгой. Спасибо, что много раз исправляла ошибки в тексте, а также создавала и настраивала репозиторий кода на GitHub.



О книге

Книга «Грокаем глубокое обучение» закладывает фундамент для дальнейшего овладения технологией глубокого обучения. Она начинается с описания основ нейронных сетей и затем подробно рассматривает дополнительные уровни и архитектуры.

Кому адресована книга

Я специально писал книгу с намерением обеспечить минимально возможный порог входа. Вам не требуются знания линейной алгебры, численных методов, выпуклых оптимизаций и даже машинного обучения. Все, что потребуется для понимания глубокого обучения, будет разъясняться по ходу дела. Если вы окончили среднюю школу и владеете языком Python, этого будет вполне достаточно для чтения книги.

Структура

Книга состоит из 16 глав:

- ❑ Глава 1 рассказывает, зачем необходимо изучать глубокое обучение и что вам потребуется на начальном этапе.
- ❑ Глава 2 начинает погружение в основные понятия, такие как машинное обучение, параметрические и непараметрические модели и обучение с учителем

и без учителя. Она также знакомит с парадигмой «предсказание, сравнение, обучение», рассмотрение которой будет продолжено в следующих главах.

- ❑ Глава 3 показывает примеры использования простых сетей для предсказания, а также впервые знакомит с нейронными сетями.
- ❑ Глава 4 научит оценивать прогнозы, сделанные в главе 3, и выявлять ошибки, что позволит продолжить обучение моделям на следующем этапе.
- ❑ Глава 5 акцентирует внимание на части «обучение» в парадигме «предсказание, сравнение, обучение». Эта глава рассматривает процесс обучения на более обширном примере.
- ❑ В главе 6 вы создадите свою первую «глубокую» нейронную сеть, напишете код и сделаете все остальное.
- ❑ Глава 7 содержит общий обзор нейронных сетей, что поможет вам сформировать свое представление.
- ❑ Глава 8 познакомит вас с такими понятиями, как переобучение, регуляризация и пакетный градиентный спуск, а также научит, как классифицировать набор своих данных в новой, только что созданной сети.
- ❑ Глава 9 расскажет о функциях активации и как их использовать при моделировании вероятностей.
- ❑ Глава 10 знакомит со сверточными нейронными сетями, акцентируя внимание на приемах для предотвращения переобучения.
- ❑ Глава 11 посвящена обработке естественного языка (natural language processing, NLP) и определяет базовый словарь и понятия в области глубокого обучения.
- ❑ Глава 12 рассматривает рекуррентные нейронные сети, современный метод глубокого обучения, используемый почти во всех областях моделирования последовательностей, один из самых популярных инструментов в отрасли.
- ❑ Глава 13 кратко рассказывает, как создать свой фреймворк глубокого обучения с нуля и стать опытным пользователем таких фреймворков.
- ❑ В главе 14 вы используете свою рекуррентную нейронную сеть для решения более сложной задачи: моделирования текста на естественном языке.
- ❑ Глава 15 затрагивает вопросы конфиденциальности данных, знакомит с базовыми понятиями конфиденциальности, такими как федеративное

обучение, гомоморфное шифрование, и идеями, имеющими отношение к дифференцированной конфиденциальности и безопасности многосторонних вычислений.

- ❑ Глава 16 познакомит вас с инструментами и ресурсами, необходимыми для дальнейшего путешествия по миру глубокого обучения.

Соглашения об оформлении кода и его загрузке

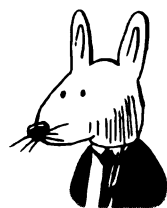
Весь программный код в книге набран моноширинным шрифтом, как здесь, чтобы его проще было отличить от обычного текста. Некоторые листинги кода сопровождаются комментариями, подчеркивающими важные понятия.

Код примеров в книге можно загрузить с сайта издательства: www.manning.com/books/grokking-deep-learning или из репозитория GitHub: <https://github.com/iamtrask/grokking-deep-learning>.

Форум книги

Приобретая книгу «Грокаем глубокое обучение», вы получаете бесплатный доступ на частный веб-форум на английском языке издательства Manning Publications, где сможете оставлять отзывы о книге, задавать вопросы и получать помощь от авторов и других пользователей. Чтобы получить доступ к форуму и зарегистрироваться на нем, откройте в браузере страницу <https://forums.manning.com/forums/grokking-deep-learning>. Кроме того, на странице <https://forums.manning.com/forums/about> вы можете узнать больше подробностей о форумах Manning и правилах поведения на них.

Издательство Manning обязуется предоставить своим читателям место встречи, где может состояться содержательный диалог между отдельными читателями и между читателями и автором. Но со стороны автора отсутствуют какие-либо обязательства уделять форуму какое-то определенное внимание — его присутствие на форуме остается добровольным (и неоплачиваемым). Мы предлагаем задавать автору стимулирующие вопросы, чтобы его интерес не угасал! Форум и архивы предыдущих дискуссий будут оставаться доступными, пока книга продолжает издаваться.



Об авторе

Эндрю Траск (Andrew Trask) — один из основателей лаборатории машинного обучения в Digital Reasoning, где изучаются методы глубокого обучения и их применение к обработке естественного языка, распознаванию образов и преобразованию речи в текст. В течение нескольких месяцев Эндрю и его коллегам удалось превзойти лучшие опубликованные результаты в области классификации эмоциональной окраски и маркировки частей речи. Он обучил крупнейшую нейронную сеть более чем со 160 миллиардами параметров и вместе со своим соавтором представил результаты на международной конференции по машинному обучению. Эти результаты были опубликованы в журнале *Journal of Machine Learning*. В настоящее время Эндрю руководит направлением анализа текста и речи в Digital Reasoning и отвечает за разработку планов анализа для платформы когнитивных вычислений Synthesys, в которой глубокое обучение является ключевой основой.

От издательства

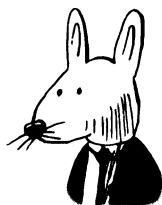
Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.

1

Введение в глубокое обучение: зачем его изучать



В этой главе

- ✓ Почему вам стоит изучать глубокое обучение.
 - ✓ Почему вы должны прочитать эту книгу.
 - ✓ Что потребуется для начала.
-

Не беспокойтесь о ваших сложностях с математикой.
Могу вас заверить, что мои сложности куда больше.

Альберт Эйнштейн

Добро пожаловать в «Грокаем глубокое обучение»!

Вы приступаете к овладению одним из самых ценных навыков века!

Рад приветствовать вас! Думаю, вы тоже рады! Глубокое обучение — это захватывающее пересечение машинного обучения и искусственного интеллекта, а также значимый прорыв для общества и промышленности. Методы, рассматриваемые в этой книге, меняют окружающий мир. Глубокое обучение используется везде: от оптимизации двигателя вашего автомобиля до выбора контента для просмотра в социальных сетях. Оно открывает широкие возможности, и, к счастью, его изучение доставляет массу удовольствия!

Почему вам стоит изучать глубокое обучение

Это мощный инструмент для постепенной автоматизации интеллектуальных задач

С незапамятных времен люди создавали все более качественные инструменты, помогающие понять окружающий мир и управлять им. Глубокое обучение является новейшей главой в этой истории инноваций.

Особенно захватывающей эту главу делает тот факт, что данная область лежит в сфере *умственных* инноваций, а не *механических*. Глубокое обучение, как и родственные ему области машинного обучения, стремится *автоматизировать интеллект* постепенно. В последние несколько лет были достигнуты большие успехи в этой сфере, превосходящие предыдущие достижения в области компьютерного зрения, распознавания речи, машинного перевода и в решении многих других задач.

Самое необычное, что для достижения всех этих успехов системы глубокого обучения используют почти *тот же алгоритм, по которому работает мозг* (нейронные сети). Даже при том, что глубокое обучение все еще остается сферой активных исследований с множеством проблем, последние разработки вызвали большое волнение: похоже, что мы нашли не просто хороший инструмент, а окно в наши собственные умы.

У глубокого обучения хороший потенциал для автоматизации умственного труда

О потенциальном влиянии глубокого обучения, если оно будет развиваться с той или иной скоростью, было сделано много шокирующих предсказаний. Многие из предсказателей явно переусердствовали, но я полагаю, что одно из них заслуживает вашего внимания: сокращение рабочих мест. Я думаю, что в отличие от остальных это утверждение имеет под собой реальную основу, потому что даже если развитие глубокого обучения прекратится прямо *сегодня*, то оно *уже* оказало большое влияние на квалификацию труда по всему миру. Операторы информационно-справочных служб, водители такси и бизнес-аналитики низшего звена являются убедительными примерами, когда глубокое обучение способно обеспечить недорогую альтернативу.

К счастью, экономика не способна в одночасье совершить крутой разворот; но во многих сферах мы уже столкнулись с неприятностями, обусловленными развитием технологий. Надеюсь, что вы (и ваши знакомые) сможете с помощью этой книги перейти из одной отрасли, столкнувшейся с потрясениями, в другую, где наблюдается рост и процветание, имя которому глубокое обучение.

Это весело и интересно. Пытаясь симитировать творчество и интеллект, вы многое узнаете о том, что значит быть человеком

Лично я занялся глубоким обучением потому, что это очень увлекательно. Это удивительное пересечение человека и машины. Процесс познания, означающий мысли, рассуждения и творчество, оказался для меня поучительным, увлекательным и вдохновляющим. Только представьте, что вам удалось собрать коллекцию картин, когда-либо нарисованных человеком, и теперь вы можете научить машину рисовать как Моне. Невероятно, но возможно. А наблюдение за тем, как это происходит, вызывает непередаваемые эмоции.

Этому трудно учиться?

Насколько усердно придется поработать, прежде чем наступит «весело и интересно»?

Это мой любимый вопрос. В моем понимании «весело и интересно» — это опыт переживания чего-то, чему я *научился*. Есть что-то удивительное, когда видишь, как твое творение делает что-то необычное. Если вам знакомо это чувство, тог-

да ответ прост. На нескольких страницах в главе 3 вы создадите свою первую нейронную сеть. Единственное, над чем вам придется потрудиться, — прочитать страницы, отделяющие вас от этого.

Следующий забавный этап после главы 3, если вам интересно узнать, наступит в главе 4, когда вы после знакомства с небольшим фрагментом кода доберетесь до середины главы. Примерно так построены все главы: знакомство с небольшим фрагментом кода в предыдущей главе, чтение следующей главы и новый всплеск интереса от знакомства с новой нейронной сетью.

Почему вы должны прочитать эту книгу

Устанавливает очень низкий порог входа

Причина, по которой вы должны прочитать эту книгу, совпадает с причиной, по которой я ее написал. Я не знаю другого источника знаний (книги, курса, серии статей в блоге), который рассказывал бы о глубоком обучении, *не предполагая наличия у читателя углубленных знаний математики* (которые даются, например, в университете).

Не поймите меня неправильно: есть более чем веские причины для преподавания этого предмета с привлечением математики. В конце концов, математика — это язык, и намного *эффективнее* изучать глубокое обучение с использованием этого языка, однако я не считаю, что углубленные знания математики абсолютно необходимы, чтобы стать опытным и знающим практиком, который четко понимает, *как* происходит глубокое обучение.

Итак, почему вы должны изучать глубокое обучение с помощью этой книги? Я предполагаю, что у вас за плечами лишь школьный курс математики (и кое-что уже позабылось), и собираюсь *объяснять все остальное по ходу дела*. Помните таблицу умножения? Помните графики с системой координат x/y (такие квадраты с линиями на них)? Отлично! Вам этого будет достаточно.

Помогает понять, что находится внутри фреймворка (Torch, TensorFlow и других)

Есть две разновидности учебных материалов по глубокому обучению (книги или курсы). В материалах из одной группы рассказывается, как использовать популярные фреймворки и библиотеки, такие как Torch, TensorFlow, Keras и другие. А материалы из другой группы описывают собственно принципы глубокого обучения, то есть *теорию*, на которой эти фреймворки основываются.

Важно знать *и то и другое*. Если бы вы захотели стать пилотом NASCAR, то вам было бы нужно знать не только конкретную модель болида со всеми ее тонкостями (фреймворк), но и уметь управлять им (теория/навыки). Простое изучение фреймворков напоминает изучение достоинств и недостатков Chevrolet SS шестого поколения до знакомства с рычагом переключения передач. Эта книга познакомит вас с глубоким обучением и подготовит к изучению фреймворков.

Все связанное с математикой будет подкрепляться простыми и понятными аналогиями

Всякий раз, встречаясь с математической формулой в дикой природе, я использую двухэтапный подход. Сначала подбираю понятную *аналогию* из реального мира. Я почти никогда не принимаю формулы на веру и всегда разбиваю их на *части*, каждую со своей историей. Этот же подход я использовал и здесь. Каждый раз, подходя к какому-то математическому понятию, я буду предлагать аналогию тому, что фактически делает формула.

*Все должно быть простым, насколько возможно,
но не проще.*

Приписывается Альберту Эйнштейну

Все после вступительных глав основано на понятии проекта

Если и есть что-то, что мне не нравится при изучении чего-то нового, так это недосказанность в отношении полезности или уместности изучаемого. Если кто-то во всех подробностях рассказывает мне об устройстве молотка, но не берет меня за руку и не учит забивать гвозди, я считаю, что он не учит меня владению молотком. Я знаю, что встречу что-то, не связанное между собой, и если окажусь в реальном мире с молотком, коробкой гвоздей и парой досок, мне придется кое о чем догадываться самому.

Эта книга *сначала* даст вам доски, гвозди и молоток, а потом расскажет, что с ними делать. Каждый урок описывает, какие инструменты выбрать и как с их помощью что-то сконструировать, а также объясняет, как все это работает. При таком подходе вы не просто будете иметь список фактов о разных инструментах глубокого обучения, но сможете их использовать для решения задач. Кроме того, вы будете знать самое важное: когда, почему и для каких задач следует использовать каждый инструмент. Благодаря этому знанию вы сможете продолжить карьеру в области исследований или в промышленности.

Что нужно для начала

Установите Jupyter Notebook и библиотеку NumPy для Python

Моим самым любимым рабочим инструментом является Jupyter Notebook. Я считаю, что при освоении глубокого обучения очень важно иметь возможность остановить процесс обучения сети и разобрать ее на составные части, чтобы увидеть, как она устроена. А для этого нет удобнее инструмента, чем Jupyter Notebook.

Ничто не скроется от нашего внимания, так как в книге используется единственная библиотека — библиотека матричных вычислений NumPy. Благодаря такому подходу вы узнаете саму *суть* глубокого обучения, а не только список функций фреймворка и порядок их вызова. Эта книга рассказывает о глубоком обучении всё: от начала и до конца.

Инструкции по установке этих инструментов можно найти по адресам: <http://jupyter.org> (для Jupyter) и <http://numpy.org> (для NumPy). Все примеры я создавал в Python 2.7, но также проверил их в Python 3. Чтобы Reddy упростить установку, рекомендую использовать фреймворк Anaconda: <https://docs.continuum.io/anaconda/install>.

Повторите школьный курс математики

Как уже отмечалось, эта книга не предполагает наличие у читателя специальной математической подготовки и моя цель — поведать вам о глубоком обучении, предполагая лишь базовые знания школьной алгебры.

Найдите задачу, которая интересна лично вам

Это условие кажется «необязательным» для начала. Может быть и так, но я вполне серьезно советую найти такую задачу. У всех, кого я знаю и кто добился успеха на этом поприще, была своя задача, которую они пытались решить. Изучение глубокого обучения было лишь «зависимостью», необходимой для решения некоторой другой интересной задачи.

Моей задачей было использование Twitter для предсказания котировок на фондовой бирже. Я просто посчитал такую задачу увлекательной. Именно она заставила меня сесть, прочитать следующую главу и создать прототип.

И, как выяснилось, эта область *настолько нова* и меняется *настолько быстро*, что если вы потратите пару следующих лет, стараясь реализовать один проект

с помощью этих инструментов, то быстро войдете в число ведущих экспертов по этой *конкретной теме*, намного быстрее, чем кажется сейчас. В моем случае такая погоня за идеей всего за 18 месяцев превратила меня, человека почти ничего не знающего о программировании, в опытного специалиста, который был удостоен гранта хедж-фонда! Для освоения глубокого обучения важно иметь задачу, увлекающую вас и предусматривающую использование одного набора данных для прогнозирования другого. Обязательно найдите такую задачу!

Возможно, вам потребуется знание Python

Python — мой любимый язык для обучения, но я покажу примеры на некоторых других интерактивных языках

Python — удивительно простой и понятный язык. Более того, я считаю его самым популярным и понятным из всех языков, созданных когда-либо. Сообщество пользователей Python испытывает неодолимую страсть к простоте. По этим причинам я написал все примеры на Python (точнее на Python 2.7). В исходном коде примеров, доступном для загрузки на сайте издательства www.manning.com/books/grokking-deep-learning, а также в репозитории на GitHub <https://github.com/iamtrask/Grokking-Deep-Learning>, я представлю примеры на некоторых других интерактивных языках.

Насколько большой опыт программирования требуется?

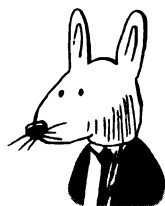
Загляните на страницу курса Python Codecademy (www.codecademy.com/learn/python). Если вы прочитали оглавление и все упомянутые там термины вам знакомы, значит, всё в порядке! Если нет, то пройдите курс до конца и возвращайтесь к книге, когда закончите. Он ориентирован на начинающих и очень хорошо продуман.

Итоги

Если вы установили Jupyter Notebook и знакомы с основами программирования на Python, тогда вы готовы перейти к следующей главе. Глава 2 — это последняя глава в этой книге, где мы с вами будем просто беседовать, ничего не создавая. Ее цель — познакомить вас с терминологией и с базовыми понятиями в области искусственного интеллекта, машинного обучения и, самое главное, глубокого обучения.

2

Основные понятия: как учатся машины?



В этой главе

- ✓ Что такое глубокое обучение, машинное обучение и искусственный интеллект.
- ✓ Что такое параметрические модели и непараметрические модели.
- ✓ Что такое обучение с учителем и обучение без учителя.
- ✓ Как учатся машины.

Через пять лет машинное обучение обеспечит успех
каждого первичного размещения акций.

Эрик Шмидт (Eric Schmidt),
председатель правления Google,
из вступительной речи на конференции
Cloud Computing Platform conference в 2016 году

Что такое глубокое обучение?

Глубокое обучение — это подмножество методов машинного обучения

Глубокое обучение (deep learning) — это подмножество методов машинного обучения, области изучения и создания машин, которые могут обучаться (иногда с целью достичь уровня искусственного интеллекта).

Глубокое обучение используется в промышленности для решения практических задач в самых разных областях, таких как компьютерное зрение (изображения), обработка естественного языка (текст) и автоматическое распознавание речи. Проще говоря, глубокое обучение — это подмножество *методов* машинного обучения, главным образом основанных на применении *искусственных нейронных сетей*, которые представляют класс алгоритмов, подражающих человеческому мозгу.



Обратите внимание, что глубокое обучение, как показано на этом рисунке, не полностью входит в область искусственного интеллекта (разумные машины, как в кино). Эта технология часто используется для решения широкого круга практических задач. Цель этой книги — познакомить с основами глубокого обучения, на которые опираются и передовые исследования, и практические применения, и подготовить вас к приложению своих сил в том или ином направлении.

Что такое машинное обучение?

Область исследований, которая дает компьютерам способность обучаться без непосредственного программирования.

Приписывается Артуру Сэмюэлю (Arthur Samuel)

Если глубокое обучение является подмножеством методов машинного обучения, тогда что такое машинное обучение? В общем и целом это именно то, что подразумевается. Машинное обучение — это область computer science, в которой *машины учатся* решать задачи, для которых они *не были запрограммированы непосредственно*. Проще говоря, машины наблюдают закономерности и пытаются прямо или косвенно некоторым способом имитировать их.

Машинное обучение $\sim$ Обезьяна видит,
обезьяна делает

Я упомянул прямую и косвенную имитации как параллель с двумя основными видам машинного обучения: *с учителем* и *без учителя*. Машинное обучение с учителем — это прямая имитация закономерностей, имеющих место между двумя наборами данных. В нем всегда входной набор данных преобразуется в выходной. Часто это невероятно мощный и полезный метод. Рассмотрим следующие примеры (**входные** данные выделены жирным шрифтом, а *выходные* — курсивом):

- ❑ Использование **пикселей** изображения для определения *присутствия* или *отсутствия* кота.
- ❑ Использование списка **понравившихся фильмов** для выбора *фильмов, которые могут понравиться*.
- ❑ Использование **слов в сообщении**, чтобы предсказать, *счастливы* ли их автор или *расстроен*.
- ❑ Использование **данных** с метеорологических приборов для предсказания *вероятности дождя*.
- ❑ Использование **датчиков** автомобильного двигателя для определения *оптимальных настроек*.
- ❑ Использование **новостей** для предсказания завтрашних *котировок* на бирже.

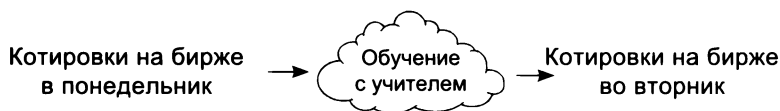
- Использование входного **числа** для предсказания удвоенного *числа*.
- Использование **аудиофайла** для получения *транскрипции* речи, содержащейся в нем.

Все это — задачи машинного обучения с учителем. Во всех случаях алгоритм машинного обучения пытается выявить такие закономерности между двумя наборами данных, чтобы *по одному можно было спрогнозировать другой*. А теперь представьте, что для любого из этих примеров вы получили возможность предсказать *результат*, имея только **входной** набор данных. Значимость такой возможности трудно было бы переоценить.

Машинное обучение с учителем

Машинное обучение с учителем преобразует наборы данных

Машинное обучение с учителем — это метод преобразования одного набора данных в другой. Например, если представить, что имеется один набор данных «Котировки на бирже в понедельник», в котором записаны все котировки, имевшие место в каждый понедельник в течение последних 10 лет, и второй набор «Котировки на бирже во вторник» с котировками за тот же период, то алгоритм машинного обучения с учителем может попытаться использовать первый, чтобы предсказать второй.



Если вам удастся успешно обучить алгоритм машинного обучения с учителем на 10-летних наборах данных с котировками по понедельникам и по вторникам, то вы сможете предсказывать котировки в любой вторник в будущем, имея котировки за предшествующий понедельник. А теперь давайте остановимся и немного поразмышляем.

Машинное обучение с учителем лежит в основе прикладного искусственного интеллекта (также известного как ограниченный ИИ). Его удобно использовать, когда на входе имеется *нечто известное* и требуется быстро преобразовать его в то, *что хотелось бы знать*. Это позволяет алгоритмам машинного обучения с учителем расширять человеческий интеллект едва ли не до бесконечности.

Основным результатом машинного обучения является обученный классификатор некоторого типа. Даже машинное обучение без учителя (с которым мы познакомимся чуть ниже) обычно выполняется для разработки точного алгоритма машинного обучения с учителем.



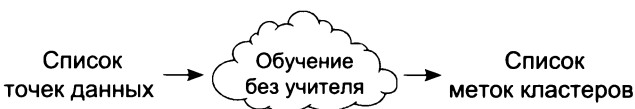
Далее мы будем создавать алгоритмы, принимающие входные данные, которые можно наблюдать и записывать, то есть *знать*, и преобразующие их в выходные данные, требующие логической оценки. В этом сила машинного обучения с учителем.

Машинное обучение без учителя

Обучение без учителя группирует данные

Обучение без учителя и обучение с учителем обладают одним общим свойством: они оба преобразуют один набор данных в другой. Но в обучении без учителя набор данных, в который происходит преобразование, *прежде не был известен*. В отличие от обучения с учителем, здесь нет «правильного ответа», который модель должна воспроизвести. Вы просто даете команду алгоритму «найти закономерности в этих данных и сообщить о них».

Например, *кластеризация набора данных на группы* — это разновидность обучения без учителя. Кластеризация преобразует последовательность *точек данных* в последовательность *меток кластеров*. Часто роль меток играют последовательные целые числа, например, в случае 10 кластерами будут созданы метки от 1 до 10. Каждая точка данных получит метку, в зависимости от того, к какому кластеру она будет отнесена. Набор данных, состоящий из точек, превратится в набор меток. Почему в качестве меток часто выбираются числа? Алгоритм ничего не может сказать о природе кластеров, он лишь сообщает вам: «Я обнаружил тут некоторые закономерности. Похоже, что ваши данные делятся на группы. Вот они!»



Могу вас обрадовать! Эту идею кластеризации с полным правом можно считать определением обучения без учителя. Несмотря на большое разнообразие форм обучения без учителя, *их все можно рассматривать как разновидности кластеризации*. Далее в книге мы еще не раз будем рассматривать эту тему.



Взгляните на этот пример. Сможете ли вы понять, по какому принципу алгоритм объединил слова, несмотря на то что он ничего не сообщает о природе кластеров? (Ответ: 1 == няшки и 2 == вкусняшки.) Позже мы выясним, что другие формы обучения без учителя тоже являются лишь разновидностями кластеризации, и узнаем, как эти кластеры могут пригодиться в обучении с учителем.

Параметрическое и непараметрическое обучение

Упрощенно: обучение методом проб и ошибок
и вычисления и вероятность

На предыдущих двух страницах мы выяснили, что все алгоритмы машинного обучения делятся на две группы: с учителем и без учителя. Теперь обсудим деление на две группы по другим признакам: параметрические и непараметрические. Можно представить, что облако алгоритмов машинного обучения имеет два переключателя:



Как видите, в действительности существует четыре разных типа алгоритмов. Алгоритмы бывают с учителем или без, а также параметрические или непараметрические. Если, как говорилось выше, наличие или отсутствие учителя (обучающей выборки данных) определяет *тип выявляемых закономерностей*, то *параметричность* задает способ *хранения* результатов обучения и зачастую *метод обучения*. Для начала рассмотрим формальное определение параметрических и непараметрических моделей. Справедливости ради следует отметить, что споры, касающиеся точных отличий, продолжаются до сих пор.

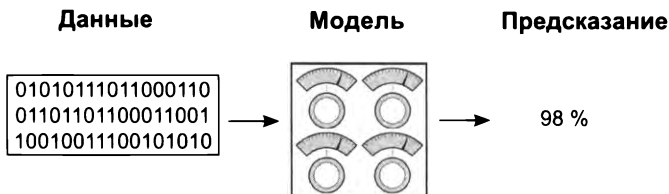
Параметрическая модель характеризуется наличием фиксированного числа параметров, тогда как непараметрическая модель имеет *бесконечное* число параметров (определяется данными).

В качестве примера возьмем задачу, в которой требуется вставить колышек с квадратным сечением в правильное (квадратное) отверстие. Некоторые люди (например, маленькие дети) просто пытаются вставить колышек во все отверстия, пока он не встанет (параметрический подход). Ребенок постарше уже может подсчитать число сторон (четыре) у колышка и найти отверстие с тем же числом сторон (непараметрический подход). Параметрические модели обычно используют метод проб и ошибок, тогда как непараметрические модели обычно основаны на вычислениях. А теперь рассмотрим эти модели подробнее.

Параметрическое обучение с учителем

Упрощенно: обучение методом проб и ошибок с использованием регуляторов

Параметрические модели обучения с учителем — это модели, имеющие фиксированное число регуляторов (это параметрическая часть таких моделей), обучение которых происходит путем поворота регуляторов. Входные данные обрабатываются согласно углу поворота регуляторов и преобразуются в *предсказание*.



Обучение осуществляется поворотом регуляторов на разные углы. Если вы попытаетесь предсказать вероятность выигрыша мировой серии бейсбольной командой Red Sox, тогда эта модель сначала примет исходные данные (такие, как статистика побед/поражений или среднее число игр, сыгранных игроками) и сделает прогноз (например, вероятность 98 %). Затем модель проверит, действительно ли команда Red Sox выиграла мировую серию. Затем, зная результат, алгоритм обучения *повернет регуляторы*, чтобы в следующий раз, когда он получит *те же или похожие исходные данные*, можно было дать более точный прогноз.

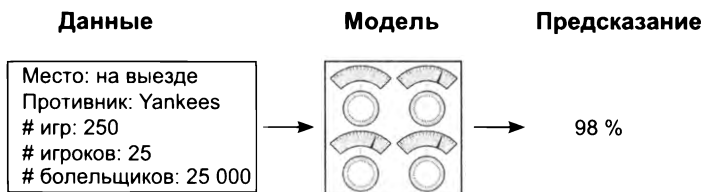
Возможно, он «повернет» в сторону увеличения регулятор «учета побед/поражений», если победы и поражения команды окажутся хорошим прогнозирующим признаком. И наоборот, он может «повернуть» в сторону уменьшения регулятор «среднего числа игр на игрока», если этот признак окажется слабо влияющим на прогноз. Именно так обучаются параметрические модели!

Обратите внимание, что результат обучения модели в любой момент можно зафиксировать по положениям регуляторов. Этот способ обучения модели также можно представить как алгоритм поиска. Вы стараетесь «найти» подходящие положения регуляторов, настраивая их и повторяя попытки.

Отметьте также, что понятие «метод проб и ошибок» не является формальным определением, но считается общим свойством параметрических моделей (за некоторым исключением). Когда имеется некоторое фиксированное число регуляторов, для определения оптимальной конфигурации требуется выполнить некоторый поиск. Непараметрические модели, напротив, часто основаны на вычислениях (в той или иной степени) и добавляют новые регуляторы, когда обнаруживается что-то новое, пригодное для использования в вычислениях. Давайте разобьем параметрическое обучение с учителем на три этапа.

Этап 1: прогноз

Для иллюстрации параметрического обучения с учителем продолжим аналогию со спортивными состязаниями и попыткой предсказать победу Red Sox в мировой серии. На первом шаге, как уже упоминалось, производится сбор статистики, ввод ее в машину и прогнозирование вероятности победы Red Sox.



Этап 2: сравнение с истиной

Второй этап — сравнение прогноза (98 %) с истиной (выиграла ли команд Red Sox на самом деле). К сожалению, они проиграли, поэтому

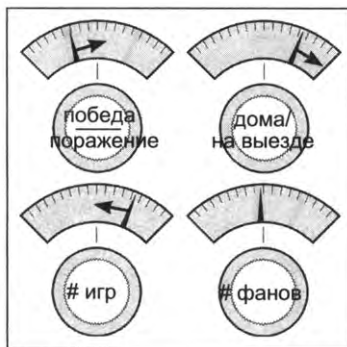
Прогноз : 98 % > Истина : 0 %

На этом этапе выясняется, что если бы модель выдала прогноз 0 %, то отлично предсказала бы грядущий проигрыш команды. Нам нужно повысить точность прогноза, поэтому переходим к этапу 3.

Этап 3: обучение

На этом этапе модель поворачивает регуляторы, учитывая величину ошибки (98 %) и исходные данные на момент прогноза (статистика состязаний), чтобы уточнить прогноз по заданным исходным данным.

Корректировка чувствительности поворотом регуляторов



Теоретически, когда эта модель увидит ту же статистику состязаний, она вернет прогноз меньше 98 %. Обратите внимание, что каждый регулятор представляет *чувствительность прогноза к разным типам входных данных*. Именно они меняются при «обучении».

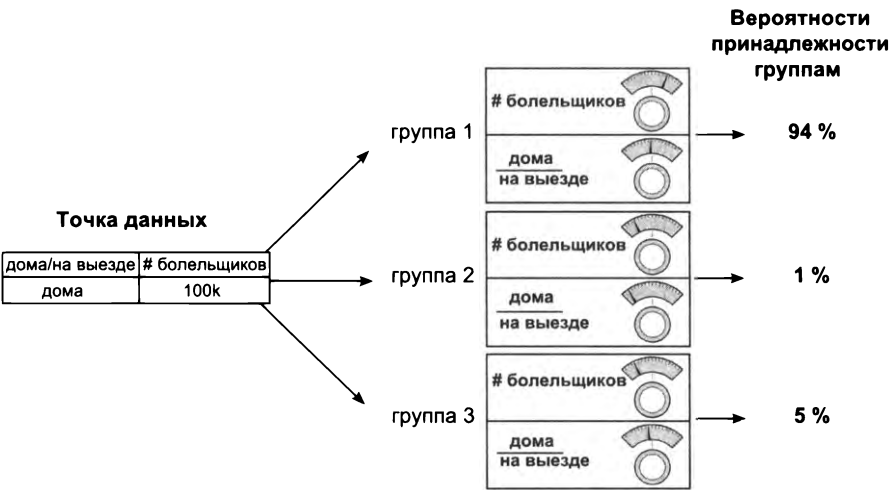
Параметрическое обучение без учителя

В параметрическом обучении без учителя используется схожий подход. Давайте рассмотрим в общих чертах этапы такого обучения. Не забывайте, что, по сути, обучение без учителя осуществляет группировку данных. В *параме-*

трическом обучении без учителя регуляторы используются для группировки данных. В этом случае обычно имеется несколько регуляторов по числу групп, каждый из которых отражает близость входных данных к конкретной группе (с некоторыми исключениями и нюансами не забывайте, что это всего лишь обобщенное описание). Рассмотрим пример, предполагающий деление данных на три группы.

Дома или на выезде	Болельщиков
Дома	100k
На выезде	50k
Дома	100k
Дома	99k
На выезде	50k
На выезде	10k
На выезде	11k

В этом наборе данных я выделил разным шрифтом три кластера — **группа 1**, группа 2 и группа 3, — которые должна выявить параметрическая модель. Передадим первую точку данных модели, обученной без учителя, как показано ниже. Обратите внимание, что она наиболее близка **группе 1**.



Для каждой группы модель пытается преобразовать входные данные в число от 0 до 1, сообщая *вероятность принадлежности данных к этой группе*. Модели могут обучаться самыми разными способами и получать самые разные свой-

параметрические алгоритмы в некоторой степени зависят от числа классов в данных, даже при том, что они явно не подсчитывают имеющиеся закономерности.

Из вышесказанного вытекает, что *параметры* — это обобщенный термин, относящийся лишь к множеству чисел, используемых для моделирования закономерностей (без каких-либо ограничений в отношении использования этих чисел). Счетчики — это параметры. Веса — это параметры. Нормализованные значения счетчиков или весов — это параметры. Коэффициенты корреляции тоже могут быть параметрами. Этот термин обозначает набор чисел, используемых для моделирования. Кстати, глубокое обучение является классом параметрических моделей. В этой книге мы больше не будем возвращаться к непараметрическим моделям, но имейте в виду, что они представляют интересный и обширный класс алгоритмов.

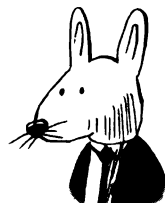
Итоги

Мы рассмотрели некоторые виды машинного обучения. Вы узнали, что машинное обучение может быть с учителем или без учителя, а также параметрическим или непараметрическим. Мы рассмотрели отличительные черты этих четырех групп алгоритмов. Вы узнали, что машинное обучение с учителем — это класс алгоритмов, обучающихся предсказанию одного набора данных по другому, а обучение без учителя фактически сводится к делению набора данных на группы. Вы также узнали, что параметрические алгоритмы имеют фиксированное число *параметров*, а непараметрические алгоритмы выбирают число параметров, основываясь на данных.

Для обучения с учителем и без учителя в глубоком обучении используются нейронные сети. До сих пор мы оставались на концептуальном уровне и пытались сориентироваться и понять, где находимся. В следующей главе мы создадим свою первую нейронную сеть, и все последующие главы будут сопровождаться *разработкой проектов*. Итак, доставайте свой блокнот Jupyter Notebook и приступим!

3

Введение в нейронное прогнозирование: прямое распространение



В этой главе

- ✓ Простая сеть, делающая прогноз.
- ✓ Что такое нейронная сеть и что она делает.
- ✓ Прогнозирование с несколькими входами.
- ✓ Прогнозирование с несколькими выходами.
- ✓ Прогнозирование с несколькими входами и выходами.
- ✓ Прогнозирование на основе прогнозов.

Я стараюсь не давать прогнозов. Это самый простой способ поставить себя в идиотское положение.

*Уоррен Эллис (Warren Ellis),
автор комиксов, прозаик и сценарист*