

# Содержание

## ТОМ 1. ОСНОВЫ

Глава 1. Введение в машинное обучение и глубокое обучение

Глава 2. Хаотичность и базовая статистика

Глава 3. Вероятность

Глава 4. Правило Байеса

Глава 5. Кривые и поверхности

Глава 6. Теория информации

Глава 7. Классификация

Глава 8. Обучение и тестирование

Глава 9. Избыточное обучение и недостаточное обучение

Глава 10. Нейроны

Глава 11. Обучение и мышление

Глава 12. Подготовка данных

Глава 13. Классификаторы

Глава 14. Ансамбли

Глава 15. Библиотека Scikit-learn

Глава 16. Нейронные сети прямого распространения

Глава 17. Функции активации

Глава 18. Обратное распространение

Глава 19. Оптимизаторы

## ТОМ 2. ПРАКТИКА

Предисловие. Добро пожаловать!.....12

Глава 20. Глубокое обучение.....19

20.1. Зачем здесь эта глава.....19

20.2. Обзор глубокого обучения.....19

20.2.1. Тензоры.....21

20.3. Вход и выход слоев.....22

20.3.1. Входной слой.....22

20.3.2. Выходной слой.....23

20.4. Обзор слоев глубокого обучения.....23

20.4.1. Полносвязные слои .....	24
20.4.2. Функции активации .....	25
20.4.3. Выпадающий слой .....	26
20.4.4. Групповая нормализация .....	27
20.4.5. Свертка .....	29
20.4.6. Объединение слоев .....	31
20.4.7. Рекуррентные слои .....	32
20.4.8. Другие применяемые слои .....	33
20.5. Обзор схематических символов слоев .....	34
20.6. Некоторые примеры .....	35
20.7. Построение глубокого обучающегося .....	43
20.7.1. Начало .....	45
20.8. Интерпретация результатов .....	46
20.8.1. Удовлетворительное объяснение .....	50
Справочные материалы .....	52
Заимствованные рисунки .....	53
<b>Глава 21. Нейронные сети свертки .....</b>	<b>54</b>
21.1. Зачем здесь эта глава .....	54
21.2. Введение .....	54
21.2.1. Два значения слова «глубина» .....	55
21.2.2. Сумма масштабированных величин .....	56
21.2.3. Разделение веса .....	59
21.2.4. Локальное воспринимаемое поле .....	61
21.2.5. Ядро .....	62
21.3. Свертка .....	63
21.3.1. Фильтры .....	66
21.3.2. С высоты птичьего полета .....	69
21.3.3. Иерархии фильтров .....	69
21.3.4. Дополнение пробелами (паддинг) .....	76
21.3.5. Величина шага .....	78
21.4. Многомерная свертка .....	80
21.4.1. Фильтры со многими каналами .....	83
21.4.2. Большой шаг в иерархиях .....	85
21.5. Свертка 1D .....	86
21.6. Свертка 1×1 .....	87
21.7. Слой свертки .....	89
21.7.1. Инициализация весов фильтров .....	90
21.8. Транспонированная свертка .....	91
21.9. Пример сети свертки .....	95
21.9.1. Сеть VGG16 .....	98
21.9.2. Посмотрим на фильтры. Часть 1 .....	102
21.9.3. Посмотрим на фильтры. Часть 2 .....	107
21.10. Противники .....	111
Справочные материалы .....	114
Заимствованные рисунки .....	116

<b>Глава 22. Рекуррентные нейронные сети .....</b>	<b>117</b>
22.1. Зачем здесь эта глава .....	117
22.2. Введение .....	118
22.3. Состояние .....	121
22.3.1. Использование состояния .....	122
22.4. Структура ячейки RNN.....	126
22.4.1. Ячейка со многими состояниями.....	129
22.4.2. Интерпретация величин состояния.....	132
22.5. Организация входов .....	132
22.6. Обучение RNN .....	135
22.7. LSTM и GRU .....	138
22.7.1. Вентиль.....	138
22.7.2. LSTM .....	141
22.8. Структура RNN.....	146
22.8.1. Один или много входов и выходов .....	146
22.8.2. Глубокие RNN .....	149
22.8.3. Двухнаправленные RNN.....	150
22.8.4. Глубокие двухнаправленные RNN .....	151
22.9. Пример.....	151
Справочные материалы .....	157
 <b>Глава 23. Keras. Часть 1.....</b>	 <b>161</b>
23.1. Зачем здесь эта глава .....	161
23.1.1. Структура этой главы .....	162
23.1.2. Ноутбуки .....	162
23.1.3. Предупреждения Python.....	162
23.2. Библиотеки и отладка .....	163
23.2.1. Версии и стиль программирования .....	164
23.2.2. Программирование на Python и отладка .....	165
23.3. Обзор.....	166
23.3.1. Что такое модель? .....	167
23.3.2. Тензоры и решетки .....	167
23.3.3. Установка Keras.....	167
23.3.4. Форма тензоров изображений .....	168
23.3.4. Графический процессор и ускорители.....	171
23.4. Начало работы .....	171
23.4.1. Hello, World .....	172
23.5. Подготовка данных .....	174
23.5.1. Переформатирование .....	175
23.5.2. Загрузка данных .....	182
23.5.3. Глядя на данные .....	184
23.5.4. Разбиение на обучающий и тестовый наборы.....	189
23.5.5. Исправление типа данных.....	190
23.5.6. Нормализация данных.....	191
23.5.7. Исправление маркировок.....	193
23.5.8. Вся предварительная обработка в одном месте .....	197

23.6. Создание модели.....	198
23.6.1. Преобразование решетки в список.....	199
23.6.2. Создание модели.....	201
23.6.3. Компиляция модели .....	206
23.6.4. Резюме создания модели.....	209
23.7. Обучение модели.....	210
23.8. Обучение и использование модели .....	213
23.8.1. Взгляд на выходные данные.....	214
23.8.2. Предсказание.....	217
23.8.3. Анализ истории обучения .....	221
23.9. Сохранение и загрузка.....	223
23.9.1. Сохранение всего в одном файле.....	223
23.9.2. Сохранение только весов .....	224
23.9.3. Сохранение только архитектуры .....	224
23.9.4. Использование предварительно обученных программ .....	225
23.9.5. Сохранение шагов предварительной обработки .....	226
23.10. Обратные вызовы.....	227
23.10.1. Контрольная точка .....	227
23.10.2. Скорость обучения .....	230
23.10.3. Ранняя остановка .....	231
Справочные материалы .....	233
Заемствованные рисунки .....	235
<b>Глава 24. Keras. Часть 2.....</b>	<b>236</b>
24.1. Зачем здесь эта глава .....	236
24.2. Улучшение модели .....	236
24.2.1. Подсчет гиперпараметров .....	237
24.2.2. Изменение одного гиперпараметра .....	238
24.2.3. Другие пути улучшения .....	240
24.2.4. Добавление плотного слоя .....	241
24.2.5. Меньше – больше .....	242
24.2.6. Добавление выпадения.....	244
24.2.7. Наблюдения .....	249
24.3. Использование библиотеки Scikit-Learn .....	249
24.3.1. Упаковщик библиотеки Keras .....	250
24.3.2. Кросс-валидация .....	253
24.3.3. Кросс-валидация с нормализацией .....	257
24.3.4. Поиск гиперпараметров .....	259
24.4. Нейронные сети свертки (CNN) .....	267
24.4.1. Сервисные слои .....	268
24.4.2. Подготовка данных для сетей свертки .....	270
24.4.3. Слои свертки.....	273
24.4.4. Использование свертки для MNIST.....	279
24.4.5. Комбинации слоев .....	288
24.4.6. Увеличение данных изображения.....	290
24.4.7. Синтетические данные .....	294
24.4.8. Поиск параметров для CNN.....	296

24.4. Рекуррентные нейронные сети (RNN) .....	296
24.5.1. Генерация последовательных данных .....	296
24.5.2. Подготовка данных для RNN .....	299
24.5.3. Построение и обучение RNN .....	305
24.5.4. Анализ работы RNN .....	308
24.5.5. Более сложные наборы данных .....	315
24.5.6. Глубокая RNN .....	317
24.5.7. Значение большого количества данных .....	320
24.5.8. Возвращаемые последовательности .....	323
24.5.9. RNN с фиксацией состояния .....	327
24.5.10. Распределенные во времени слои .....	329
24.5.11. Генерирование текста .....	333
24.6. Интерфейс прикладного программирования .....	339
24.6.1. Входные слои .....	341
24.6.2. Создание функциональной модели .....	342
Справочные материалы .....	347
Заемствованные рисунки .....	347
<b>Глава 25. Автокодировщики .....</b>	<b>348</b>
25.1. Зачем здесь эта глава .....	348
25.2. Введение .....	349
25.2.1. Кодирование с потерями и без потерь .....	349
25.2.2. Доменное кодирование .....	350
25.2.3. Смешивание представлений данных .....	352
25.3. Простейший автокодировщик .....	355
25.4. Более сложные автокодировщики .....	360
25.5. Исследование автокодировщиков .....	363
25.5.1. Скрытые переменные .....	363
25.5.2. Параметрическое пространство .....	366
25.5.2. Смешивание скрытых переменных .....	371
25.5.4. Прогнозирование нового входа .....	373
25.6. Обсуждение .....	374
25.7. Сверточный автокодировщик .....	375
25.7.1. Смешивание скрытых переменных .....	377
25.7.2. Прогнозирование нового входа .....	379
25.8. Понижение уровня шума .....	380
25.9. Вариационные автокодировщики .....	382
25.9.1. Распределение скрытых переменных .....	383
25.9.2. Структура вариационного автокодировщика .....	384
25.10. Изучение VAE .....	390
Справочные материалы .....	399
Заемствованные рисунки .....	400
<b>Глава 26. Обучение с подкреплением .....</b>	<b>401</b>
26.1. Зачем здесь эта глава .....	401
26.2. Цели .....	402

26.2.1. Обучение новой игре .....	403
26.3. Структура обучения с подкреплением .....	406
26.3.1. Шаг 1: агент выбирает действие.....	408
26.3.2. Шаг 2: отклик окружающей среды .....	409
26.3.3. Шаг 3: агент обновляется.....	410
26.3.4. Вариации простой версии .....	411
26.3.5. Обратно к общей картине.....	412
26.3.6. Сохранение опыта.....	413
26.3.7. Вознаграждения .....	414
26.4. Игра флиппер .....	419
26.5. L-обучение .....	421
26.5.1. Обработка непредсказуемости.....	431
26.6. Q-обучение .....	433
26.6.1. Q-величины и обновление .....	434
26.6.2. Политика Q-обучения .....	437
26.6.3. Собираем все вместе .....	439
26.6.4. Сходимость алгоритма Q-обучения.....	440
26.6.5. Q-обучение в действии.....	441
26.7. SARSA .....	448
26.7.1. SARSA в действии .....	451
26.7.2. Сравнение Q-обучения и SARSA.....	457
26.8. Общая картина .....	461
26.9. Воспроизведение опыта .....	462
26.10. Два применения .....	463
Справочные материалы .....	465
<b>Глава 27. Порождающие состязательные сети .....</b>	<b>467</b>
27.1. Зачем здесь эта глава .....	467
27.2. Метафора: фальшивые деньги .....	468
27.2.1. Обучение на основе опыта.....	471
27.2.2. Подделка с помощью нейронных сетей.....	473
27.2.3. Циклы обучения .....	475
27.3. Почему антагонистические сети? .....	477
27.4. Применение сетей GAN.....	478
27.4.1. Дискриминатор .....	478
27.4.2. Генератор .....	478
27.4.3. Обучение сети GAN .....	480
27.4.4. Играть в игру .....	482
27.5. Сеть GAN в действии .....	482
27.6. Сети DCGAN .....	488
27.6.1. Эмпирические правила.....	490
27.7. Проблемы .....	492
27.7.1. Использование больших образцов .....	493
27.7.2. Модальный коллапс.....	493
Справочные материалы .....	495

<b>Глава 28. Применение для творчества</b>	<b>497</b>
28.1. Зачем здесь эта глава	497
28.2. Визуализирующие фильтры	497
28.2.1. Выбор сети	497
28.2.2. Визуализация одного фильтра	498
28.2.3. Визуализация одного слоя	501
28.3. Глубокие сновидения	502
28.4. Нейронное преобразование стиля	507
28.4.1. Захват стиля в матрице	507
28.4.2. Общая картина	509
28.4.3. Потери содержания	510
28.4.4. Потери стиля	512
28.4.5. Перенос стиля	516
28.4.6. Обсуждение	522
28.5. Генерация другого текста этой книги	524
Справочные материалы	525
Заимствованные рисунки	526
<b>Глава 29. Наборы данных</b>	<b>527</b>
29.1. Общедоступные наборы данных	527
29.2. MNIST and Fashion-MNIST	528
29.3. Наборы данных, встроенные в библиотеку	528
29.3.1. scikit-learn	528
29.3.2. Keras	529
29.4. Коллекции кураторских наборов данных	529
29.5. Некоторые новые наборы данных	530
<b>Глава 30. Глоссарий</b>	<b>533</b>
<b>Предметный указатель</b>	<b>606</b>

# Предисловие

.....

## Добро пожаловать!

*Несколько слов введения в эту книгу,  
как получить файлы и рисунки,  
и благодарности тем, кто помогал мне*

### Что вы получите от этой книги

Привет!

Если вы интересуетесь глубоким обучением (Deep Learning, DL) и машинным обучением (Machine Learning, ML), то эта книга для вас.

Моя цель в данной книге – дать вам прочные навыки эффективного практического применения машинного обучения и глубокого обучения.

После прочтения этой книги вы будете уметь:

- разрабатывать и обучать собственные нейронные сети;
- использовать нейронные сети для понимания данных и создания новых данных;
- присваивать описательные категории текстам, изображениям и другим типам данных;
- предсказывать последующие значения последовательности данных;
- исследовать структуру ваших данных;
- обрабатывать ваши данные с максимальной эффективностью;
- использовать языки программирования и библиотеку DL по своему желанию;
- воспринимать новые знания и идеи и применять их на практике;
- получать удовольствие от обсуждения глубокого обучения с другими специалистами.

Мы используем серьезный, но дружелюбный подход, сопровождаемый большим количеством иллюстраций. Мы делаем это без каких-либо кодов и без всякой математики, за исключением умножения.

Если это звучит привлекательно для вас, добро пожаловать!

### Для кого эта книга

Эта книга создана для тех, кто хочет использовать машинное обучение и глубокое обучение в своей работе. Это программисты, инженеры, ученые, руководители, музыканты, врачи и все, кто хочет работать с большими объемами данных, извлекая из них полезную информацию или формируя новые данные.

Многие инструменты машинного обучения и особенно глубокого обучения имеются в многочисленных библиотеках со свободным доступом, которые любой при желании может немедленно загрузить.



Но хотя эти инструменты легко доступны и легко устанавливаемы, они все же требуют значительных технических знаний для правильного их применения. Со всем не трудно попросить компьютер выполнить что-то бессмысленное, и он радостно выполнит это, выдав на выходе бессмыслицу.

Такого рода вещи происходят все время. Хотя машинное обучение и глубокое обучение – мощный инструмент, он вовсе не слишком дружелюбен к пользователю.

Выбор правильных алгоритмов и затем применение их надлежащим образом требуют в дополнение последовательных, технически грамотных решений. Когда, как часто бывает, события развиваются не так, как планировалось, необходимы знания, чтобы понять, что происходит внутри системы, с тем чтобы исправить положение дел.

Существует много подходов к освоению этих существенных знаний в зависимости от того, как вы хотите ими овладеть.

Некоторые любят жесткий детальный алгоритмический анализ, сопровождаемый обширной математикой. Если это тот способ, с помощью которого вы хотите овладеть знаниями, то существуют солидные труды, которые предлагают этот стиль представления знаний [Bishop06], [Goodfellow17]. Он требует серьезных усилий, но дает глубокое понимание того, как и почему этот механизм работает. Если вы выберете данный способ, то вам придется проделать значительную работу, чтобы применить теоретические знания на практике.

Другая крайность, когда человек просто хочет знать, как решить некую конкретную задачу. Серьезных книг, которые содержат библиотеки с рецептами машинного обучения, достаточно много [Chollet17], [Müller-Guido16], [Raschka15], [VanderPlas6]. Эти методы проще, чем математический подход, но вы можете ощущать недостаток структурной информации, объясняющей, как это работает. Без этой информации и соответствующей терминологии трудно разобраться, почему что-то, что, вы полагали, должно работать, не работает или почему что-то не работает так хорошо, как вы полагали. Это может вызвать желание почитать литературу, описывающую новые идеи и результаты, потому что обсуждения обычно предполагают общий уровень знаний о предмете, используемой библиотеке и языке.

В этой книге принят промежуточный подход. Я намерен дать вам инструменты для уверенного практического применения глубокого обучения. Я хочу, чтобы вы могли в вашей работе сделать разумный выбор и были способны следовать в общем потоке новейших идей, появляющихся почти каждый день.

Моя цель здесь – осветить фундаментальные основы достаточно глубоко, чтобы у вас была надежная база поддержки в вашей работе. Я хочу, чтобы вы имели достаточную базу не только для понимания материалов этой книги, но и материалов, которые, возможно, понадобятся вам для консультаций и изучения в процессе работы с глубоким обучением.

Это не книга о программировании. Программирование – важный аспект, но оно неизбежно вовлекает во все детали, которые необязательны для нашего основного предмета изучения. Примеры программирования ограничат нас одной библиотекой или одним языком программирования. И хотя детали необходимы для создания законченных систем, они могут отвлечь нас, когда будет необходимо сосредоточить внимание на основополагающих идеях. Вместо того чтобы вдаваться в дискуссии о циклах, индексах и структуре данных, мы будем все это

обсуждать здесь независимо от какого-либо языка или библиотеки. Если вы усвоите главную идею, то прочтение документации для любой библиотеки будет несложным делом.

Мы спустимся на землю в главах 15, 23 и 24, когда будем обсуждать научную библиотеку машинного обучения и библиотеку глубокого обучения Keras<sup>1</sup>. Обе эти библиотеки базируются на языке Python. В этих главах мы погружаемся в детали библиотеки языка программирования Python и имеем в них много примеров с кодами.

Даже если вы не знаете данного языка, эти программы послужат вам примером технологий и программных структур, что поможет справляться с новыми проблемами. Коды в этих главах с программированием доступны в качестве файлов на языке Python. Они могут быть использованы с помощью программного окружения, базирующейся на браузере графической веб-оболочки Jupyter [Jupyter16] или с помощью классического, разработанного для Python окружения, такого как PyCharm [JetBrain17].

Большинство других глав также может быть поддержано выборочными файлами (notebooks) на Python. В них приводится код для каждой компьютерной графики в книге, часто используя методы, обсуждаемые в этой главе. Поскольку в книге отсутствует ориентация на Python и программирование (за исключением упомянутых выше глав), эти записи присутствуют в книге как бы «за сценой» и только слегка комментируются.

Машинное обучение, глубокое обучение и большие данные оказывают неожиданно быстрое и глубокое влияние на общество во всем мире. Что это означает для людей и культуры – сложный и важный предмет. Ряд интересных книг и статей, обсуждающих эту тему, часто приходит к тонкой смеси положительных и отрицательных выводов [Agüera y Arcas17] [Barrat15] [Domingos15] [Kaplan16].

## ПОЧТИ БЕЗ МАТЕМАТИКИ

Многие, отнюдь неглупые люди не являются поклонниками сложных математических уравнений, и если это вы, то здесь вы дома.

В этой книге почти нет математики. Если вы справляетесь с умножением, то этого достаточно, поскольку это вся математика, которую мы используем.

Большинство обсуждаемых нами алгоритмов базируется на солидных теоретических источниках и является результатом аккуратного анализа и проработки. Важно знать это обстоятельство, когда вы модифицируете алгоритм для какой-то новой задачи или другой реализации. Но на практике почти все используют хорошо оптимизированные реализации с открытым кодом, написанные экспертами и доступные в бесплатных библиотеках.

Наша цель – понять принципы этих технологий, усвоить, как применять их надлежащим образом и как интерпретировать результаты. Ничто из этого не требует от нас вдаваться в математические подробности.

Если вы любите математику или хотите ознакомиться с теорией, следуйте ссылкам к каждой главе. Многие из этих материалов превосходны, интеллекту-

<sup>1</sup> См.: Джулли А., Пал С. Библиотека Keras – инструмент глубокого обучения. М.: ДМК Пресс, 2017. – Прим. перев.

ально полновесны и описывают детали, которые я сознательно опустил в этой книге. Но если математика не ваш конек, то и нет необходимости в математических деталях.

## МНОГО РИСУНКОВ

Некоторые идеи более отчетливо проявляются в рисунках, чем в словах. И даже если слова выполняют свою работу, рисунки цементируют идеи. Поэтому эта книга обильно проиллюстрирована рисунками.

Все рисунки данной книги доступны для свободного считывания (см. далее).

## ЗАГРУЗКИ

Вы можете загрузить Jupiter/Python для этой книги, все рисунки и другие файлы, относящиеся к этой книге, совершенно свободно.

## ВСЕ ФАЙЛЫ (NOTEBOOKS)

Все файлы оболочки Jupiter/Python в данной книге доступны на GitHub.

Файлы для главы 15 (scikit-learn – Python-модуль для машинного обучения) и глав 23 и 24 (Keras) содержат все коды, которые представлены в этих главах.

Другие файлы доступны «за сценой» для просмотра того, как получены рисунки. Они слегка документированы и служат скорее справкой, чем учебным пособием.

Файлы опубликованы по лицензии Массачусетского технологического института (MIT), что, по существу, означает, что вы свободны использовать их для любых целей. Никаких гарантий, что в них отсутствуют дефекты, что они будут надежно работать, что они не приведут к сбоям и т. п., не дается. Чувствуйте себя свободными извлекать коды и адаптировать их так, как вам удобно, хотя в лицензии говорится об использовании только в личных целях (это для файлов с пометкой LICENSE).

<https://github.com/blueberry music/DeepLearningBookCode-Volume1>

<https://github.com/blueberry music/DeepLearningBookCode-Volume2>

## ВСЕ РИСУНКИ

Все рисунки в этой книге доступны на GitHub в формате PNG с высоким разрешением. Вы можете свободно использовать их в классах, обсуждениях, лекциях, докладах, статьях и даже других книгах.

Так же, как код, рисунки опубликованы по лицензии Массачусетского технологического института (MIT), и вы можете использовать их по своему усмотрению, сохраняя уведомление об авторских правах. Используя рисунки, вы не обязаны кредитовать меня как их создателя, но я буду признателен вам, если вы сделаете это.

Наименования файлов соответствуют номерам рисунков в книге, поэтому их нетрудно найти. Когда вы будете искать что-либо визуально, то будет полезно по-

смотреть страницы в уменьшенном формате. Каждая такая страница содержит 20 изображений:

<https://github.com/blueberrymusic/DeepLearningBookFigures-Thumbnails>

Сами рисунки сгруппированы в двух томах:

<https://github.com/blueberrymusic/DeepLearningBookFigures-Volume1>

<https://github.com/blueberrymusic/DeepLearningBookFigures-Volume2>

## Источники

Перечень литературы и ресурсов содержит другие файлы, такие как шаблоны для иконок глубокого обучения, которые используются нами в этой книге.

<https://github.com/blueberrymusic/DeepLearningBook-Resources>

## ОПЕЧАТКИ

Хотя мной приняты все возможные меры, в книге такого объема трудно избежать ошибок. Если вы обнаружите что-то, что покажется вам неверным, пожалуйста, дайте мне знать по адресу [andrew@dlbasics.com](mailto:andrew@dlbasics.com).

## ДВА ТОМА

Книга оказалась очень большой, поэтому я сделал ее двухтомником с примерно одинаковым размером.

Поскольку двухтомник является, по существу, одной книгой, второй том начинается там, где заканчивается первый. Если вы в данный момент читаете второй том, то, следовательно, вы или уже прочитали первый, или чувствуете себя достаточно уверенно для понимания содержания второго тома.

## БЛАГОДАРНОСТИ

Авторы любят говорить, что никто не пишет книги в одиночку. Мы говорим так, потому что так оно и есть.

Я чрезвычайно благодарен Эрику Брауну (Eric Braun), Эрику Хейнесу (Eric Haines), Стиву Друкеру (Steven Drucker) и Тому Рейке (Tom Reike) за последовательную и воодушевляющую помощь в этом проекте, помогавшую мне уверенно чувствовать себя на протяжении всего времени осуществления проекта. Спасибо за ваше дружелюбие и поддержку.

Большое спасибо моим рецензентам за щедрые и глубокие комментарии, значительно улучшившие книгу: Адаму Финкельштейну (Adam Finkelstein), Алексу Колберну (Alex Colburn), Александру Келлеру (Alexander Keller), Алин Рокфорд (Alyn Rockwood), Анджело Песке (Angelo Pesce), Барбаре Монес, (Barbara Mones), Брайану Уайвиллу (Brian Wyvill), Крейгу Каплану (Craig Kaplan), Дагу Робле, (Doug Roble), Эрику Брауну (Eric Braun), Эрику Хейнесу (Eric Haines), Грегу Тэрку (Greg Turk), Джеффу Халтвисту (Jeff Hultquist), Джессике Ходжинс (Jessica Hodgins),

Кристи Мортон (Kristi Morton), Лезли Истед (Lesley Istead), Луис Авардо (Luis Avarado), Мэтт Фарр (Matt Pharr), Майку Тика (Mike Tyka), Морган МакГви́ре (Morgan McGuire), Паулю Бэдсли (Paul Beardsley), Паулю Страуссу (Paul Strauss), Петеру Шэрли (Peter Shirley), Филиппу Служаллеку (Philipp Slusallek), Сербан Порумбеску (Serban Porumbescu), Стефанусу Ду Тойту (Stefanus Du Toit), Стивен Друкер (Steven Drucker), Венхао Ю (Wenhao Yu) и Закори Эриксон (Zackory Erickson).

Особая благодарность ответственным рецензентам Александру Келлеру (Alexander Keller), Эрику Хейнесу (Eric Haines), Джессике Ходжинс (Jessica Hodgins) и Луис Авардо (Luis Avarado), которые прочитали всю книгу или большую часть рукописи и дали существенные предложения как по представлению содержания, так и по структуре книги.

Благодарю Морган МакГви́ре (Morgan McGuire) за применение технологии Markdeep, что позволило мне в большей степени сосредоточиться на том, что я должен сказать, чем на том, каким должен быть формат.

Спасибо Тодду Жиманскому (Todd Szymanski) за вдумчивые советы по оформлению и компоновке содержания и титула книги и выявление ошибок в их размещении.

Спасибо первым читателям, которые выявили опечатки и другие проблемы:

Кристиану Форгангу (Christian Forfang), Дэвиду Полу (David Pol), Эрику Хейнесу (Eric Haines), Гопи Меенакхисундараму (Gopi Meenakshisundaram), Косте Смоленскому (Kostya Smolenskiy), Мурисио Вивес, (Mauricio Vives), Майку Вонгу (Mike Wong) и Мринал Мохит (Mrinal Mohit).

Все эти люди улучшили книгу, но окончательные решения были за мной. И проблемы, которые остались, являются моей ответственностью.

## СПРАВОЧНЫЕ МАТЕРИАЛЫ

Этот раздел появляется в каждой главе. Он содержит справки по всем документам, которые относятся к содержанию данной главы. Здесь также могут присутствовать другие полезные материалы: статьи, веб-страницы, документация, логи и иные источники.

Где только возможно, я предпочитал те источники, которые доступны онлайн, поэтому вы можете немедленно получить их, используя Сеть. Исключение составляют обычно книги, но иногда я привожу важные онлайн-справки, даже если за ними стоят требования оплаты.

[Agüera y Arcas17] *Blaise Agüera y Arcas, Margaret Mitchell, and Alexander Todorov. Physiognomy's New Clothes. Medium, 2017. <https://medium.com/@blaisea/physiognomys-new-clothes-f2d4b59fdd6a>.*

[Barrat15] *James Barrat. Our Final Invention: Artificial Intelligence and the End of the Human Era. St. Martin's Griffin, 2015.*

[Bishop06] *Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer-Verlag, 2006. С. 149–152.*

[Chollet17] *François Chollet. Deep Learning with Python. Manning Publications, 2017.*

[Domingos15] *Pedro Domingos. The Master Algorithm. Basic Books, 2015.*

[Goodfellow17] *Ian Goodfellow, Yoshua Bengio, Aaron Courville*. Deep Learning. MIT Press, 2017. <http://www.deeplearningbook.org/><sup>1</sup>.

[JetBrains17] *Jet Brains*. Pycharm Community Edition IDE. 2017. <https://www.jetbrains.com/pycharm/>.

[Jupyter16] The Jupyter team. 2016. <http://jupyter.org/>.

[Kaplan16] *Jerry Kaplan*. Artificial Intelligence: What Everyone Needs to Know. Oxford University Press, 2016.

[Müller-Guido16] *Andreas C. Müller and Sarah Guido*. Introduction to Machine Learning with Python. O'Reilly Press, 2016<sup>2</sup>.

[Raschka15] *Sebastian Raschka*. Python Machine Learning. Packt Publishing, 2015<sup>3</sup>.

[VanderPlas16] *Jake VanderPlas*. Python Data Science Handbook. O'Reilly Media, 2016.

---

<sup>1</sup> Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение. М.: ДМК Пресс, 2017. ISBN: 978-5-97060-618-6.

<sup>2</sup> Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. М.: Вильямс, 2017. ISBN: 978-5-9908910-8-1.

<sup>3</sup> Рашка С. Python и машинное обучение. М.: ДМК Пресс, 2017. ISBN: 978-5-97060-409-0.

# Глава 20

## Глубокое обучение

*Мы увидим базовые структуры  
сетей глубокого обучения  
и обзор многих типов слоев,  
из которых они состоят*

### 20.1. ЗАЧЕМ ЗДЕСЬ ЭТА ГЛАВА

В предыдущих главах мы заложили хорошие основы для разработки алгоритмов нейронных сетей. В этой главе используем их для построения сетей путем объединения искусственных нейронов в **слои**. Как мы видели в главе 18, это позволит нам использовать эффективный алгоритм обратного распространения для повышения производительности сети. Сеть состоит из последовательности слоев, и часто ее называют **глубокой сетью**, и когда такая сеть обучает по представленным ей данным, то мы называем это **глубоким обучением**.

Мы обсудим терминологию глубокого обучения и рассмотрим наиболее популярные слои, которые используются в сетях глубокого обучения, а также некоторые примеры сетей и как построить новые сети и интерпретировать их результаты.

В этой главе будет рассказано об остальной части книги, где рассматриваются специализированные формы глубокого обучения для различных задач.

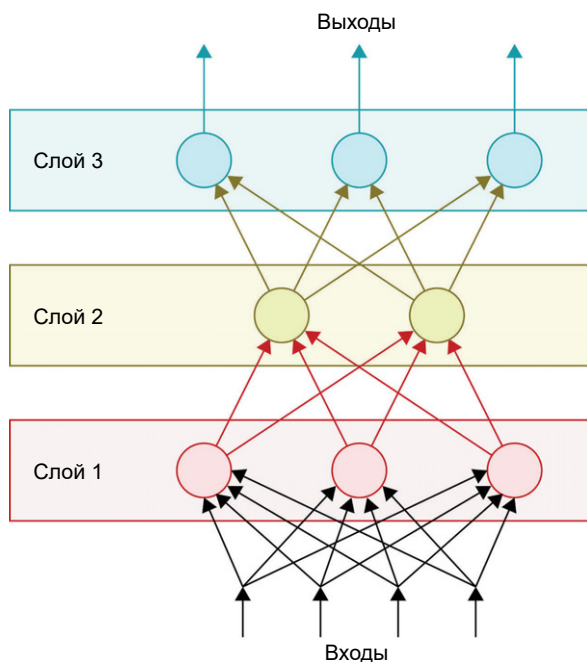
### 20.2. ОБЗОР ГЛУБОКОГО ОБУЧЕНИЯ

Нейронные сети строятся как большое количество слоев, и часто их называют **глубокими сетями** (их можно было бы назвать «высокими», «широкими» или «длинными» сетями, но общепринятым стало «глубокие»). Когда мы используем глубокие сети, мы обычно говорим, что осуществляем **глубокое обучение**.

Выражение **глубокое обучение** обычно относится к нейронным сетям, которые состоят из последовательности слоев. Более общее выражение **машинное обучение** обычно относится и к глубокому обучению, и к другим алгоритмам, которые мы рассматривали (например, классификатору в главе 13), которые не основаны на нейронных сетях. Но некоторые авторы трактуют «машинное обучение» и «глубокое обучение» как две разные области, так что «машинное обучение» касается алгоритмов, где не используются нейронные сети. Поэтому одни книги со словами «машинное обучение» в названии включают нейронные сети, а другие – нет. Всегда стоит убедиться, какое из них конкретный автор использует.

Результатом послойной организации нейронов является способность обучающей сети анализировать данные **иерархически**. Первые слои видят сырые данные, и каждый последующий слой способен использовать для обработки больший объем информации от нейронов предыдущего слоя. Для примера: при рассмотрении фотографии первый слой обычно смотрит на индивидуальные пиксели. Следующий слой смотрит на группы пикселей, слой после него – на группы этих групп и т. д. Начальные слои могут заметить, что одни пиксели темнее других, в то время как последующие слои замечают, что скопление пикселей похоже на глаз, а еще значительно позже слой может идентифицировать полосы, которые показывают, что изображение в целом – это тигр.

Рисунок 20.1 показывает пример архитектуры глубокого обучения, использующей три слоя.



**Рис. 20.1** ❖ Сеть глубокого обучения. Мы имеем 4 входа, проходящих через 3 слоя и создающих 3 выхода в конце. Мы говорим, что сеть полносвязная, потому что каждый нейрон в каждом слое получает сигнал от каждого предыдущего слоя

Когда мы рисуем слои вертикально, как на рис. 20.1, входы почти всегда мы рисуем внизу, а выходы, где появляются наши результаты, – почти всегда вверх.

Самый верхний слой (слой 3 на рис. 20.1) называется **выходным слоем**. Хотя с величинами, которые поступают из этого слоя, перед тем как воспользоваться ими, могут проводиться другие операции, например такие как софтмакс, с которым мы познакомились в главе 17, мы обычно рассматриваем этот слой как конечный в нейронной сети, потому что он последний, содержащий нейроны.

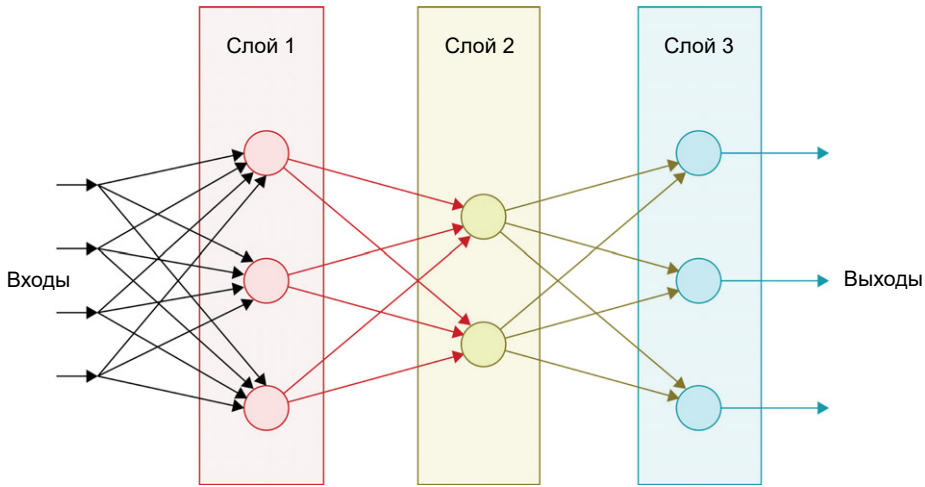
Возможно, мы ожидаем, что соответственно вначале будет входной слой, и это было бы естественно, если бы мы дали такое название слою 1 на рис. 20.1. Но



терминология не такова. «Входной слой» существует, но он редко показывается явно. Он скорее относится к памяти, которая хранит входные величины. Мы можем представлять себе входной слой как ряд стрелок вниз рис. 20.1.

Слои 1 и 2 на рис. 20.1 называются **скрытыми слоями**. Если мы представим себе кого-то, смотрящего на сеть извне, сверху или снизу, то он видит только выходной или входной слой. Мы можем представить слои, находящиеся между ними, как скрытые от взгляда, и потому их называют «скрытыми слоями» (их можно видеть со стороны, но мы не будем обращать внимания на это небольшое упущение терминологии).

Иногда мы встречаемся с рисунками сети слева направо, как на рис. 20.2.



**Рис. 20.2** ❖ Та же сеть, что и на рис. 20.1, нарисованная с данными, проходящими слева направо

Даже когда сеть нарисована таким образом, все равно будем использовать терминологию вертикальной ориентации. Авторы могут сказать, что слой 2 выше слоя 1 и ниже слоя 3. Мы можем всегда придерживаться такого подхода, независимо от того, как нарисована диаграмма. Если мы имеем в виду «выше» или «над», это означает, что слой ближе к выходу, а если «ниже» или «под» – значит, ближе к входу.

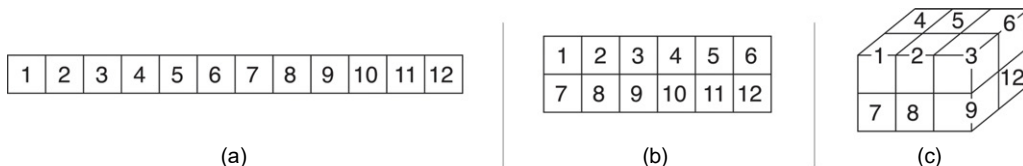
### 20.2.1. Тензоры

Хотя фундаментом сетей глубокого обучения является манипуляция числами, концептуально важна организация данных в виде списков чисел. Этот список может быть одномерным, как на рис. 20.3 (а), то есть просто числами, записанными одно за другим.

Мы можем называть это **сеткой**, или **матрицей**. Трехмерный список, как на рис. 20.3 (с), может хранить объемные данные, или выборки, каждая из которых состоит из множества характеристик, измеренных многократно. Мы называем их **объемами**, или **блоками**.

Чтобы упростить обсуждение, мы будем называть список любого размера и размерности тензором. Слово тензор имеет более широкое значение в некоторых об-

ластях математики и физики. Здесь мы используем его просто как группу чисел, организованных как многомерный список.



**Рис. 20.3** ❖ Три тензора, каждый из 12 элементов: (a) 1D-тензор является списком; (b) 2D-тензор является сеткой; (c) 3D-тензор является кубом. Во всех этих случаях, как и в случае большей размерности, вся структура полностью заполнена. То есть все ряды, столбцы и прочие поверхности имеют одинаковую длину

Поэтому мы часто будем употреблять выражение «входной тензор» (имея в виду входные величины), «выходной тензор», имея в виду выходные величины, а также «тензор» для других тензоров внутри сети, когда он вычисляет новое представление входящих данных.

Мы говорим, что каждый тензор имеет размерность и размер в каждой размерности. Все это вместе говорит о **форме** тензора.

## 20.3. Вход и выход слоев

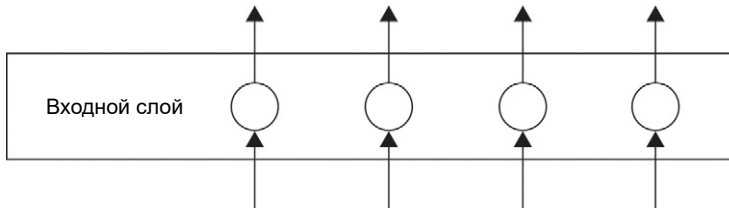
Многие сети имеют единственный входной слой и единственный выходной слой. Эти обозначения просто определяют положение слоя во множестве слоев: вход является началом (снизу или слева), а выход – концом (вверху или справа).

Как мы обсуждали раньше, входной слой не является слоем нейронов. Это лишь концептуальный хранитель для входных данных. Входной слой обычно создается и сохраняется для нас автоматически библиотекой глубокого обучения, и мы редко имеем с ним дело напрямую. Тем не менее мы должны помнить о нем хотя бы потому, что иногда мы хотим обработать наш вход, перед тем как остальная сеть получит его, поэтому мы помещаем некоторого рода шаг обработки между входным слоем и первым слоем нейронов в сети. В противоположность этому выходной слой содержит нейроны, и мы создаем его явным образом при построении сети. Его тип и структура полностью зависят от нас. Часто формальное определение для выхода отсутствует. Какой бы слой мы не поместили вверху нашего набора слоев, мы называем его «выходным слоем» для данной архитектуры.

### 20.3.1. Входной слой

**Входной слой** обычно не показывается на диаграмме архитектуры глубокого обучения. Он просто является памятью, хранящей входные данные. Заметим, что это не нейроны, поскольку входной слой не обрабатывается. Его можно представлять себе просто как некоторое собрание ячеек памяти, каждая из которых хранит одно число входа. Рисунок 20.4 иллюстрирует идею.

Некоторые авторы используют термин «входной слой», имея в виду первый слой обработки в сети, поэтому надо быть настороже, встречая термин «первый слой», и быть уверенным, в каком смысле используется этот термин.



**Рис. 20.4** ❖ Входной слой является просто хранилищем, в котором мы временно храним входные данные

### 20.3.2. Выходной слой

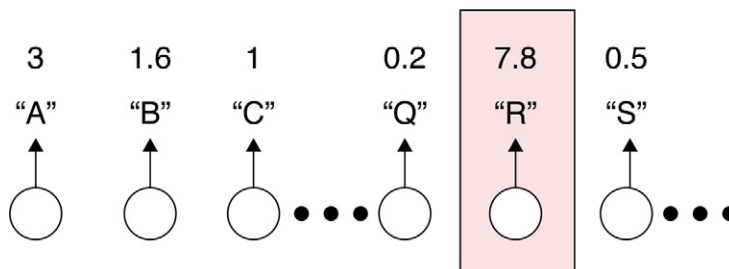
**Выходной слой** – это то, где результаты сети взаимодействуют с внешним миром.

Когда мы строим нашу архитектуру, то выбираем число нейронов в выходном слое, соответствующее решаемой задаче.

Если это задача регрессии с одним-единственным численным выходом, то в этом слое будет один нейрон, и величина этого нейрона будет нашим предсказанием.

Если мы строим бинарный классификатор, то у нас есть выбор. Величины, близкие к 0, означают вход одного класса, в то время как величины, близкие к единице, означают вход другого класса. В качестве альтернативы мы можем иметь два выходных нейрона, один для каждой категории. Обычно находят, какой нейрон имеет большую величину, и приписывают ему соответствующую категорию входа.

Классификатор с большим количеством классов чаще будет иметь столько выходов, сколько классов. Например, предположим, что мы пытаемся распознать прописные буквы латинского алфавита. Мы тогда должны иметь 26 выходных нейронов, один для каждой буквы, что обеспечит величину для каждой буквы. Мы можем выбрать выход с наибольшей величиной как лучший выбор для этой категории входа. На рис. 20.5 показана идея. Если мы интерпретируем эти выходы как вероятности, то можем пропустить их через софтмакс, как это обсуждалось в главе 17.



**Рис. 20.5** ❖ Если мы классифицируем отдельные буквы, то можем иметь 26 выходов, каждый из которых даст величину для данной буквы. Здесь показано, что буква «R» имеет наибольшую величину

## 20.4. Обзор слоев глубокого обучения

Большинство библиотек предлагает широкий выбор типов слоев. В этом разделе мы рассмотрим некоторые наиболее употребительные и полезные типы. Поскольку структура каждой библиотеки ориентируется на определенный тип слоев и то,

как они работают, будет наиболее простым в качестве примера сосредоточиться на какой-нибудь одной библиотеке. Мы выбрали в качестве таковой Keras [Keras16], потому что она предлагает хороший выбор и будет детально рассмотрена в главах 23 и 24. Даже для этой библиотеки наш обзор не будет исчерпывающим.

Мы сосредоточимся здесь только на базисной структуре и функции каждого слоя. Большинство слоев имеет опции параметров, которые могут быть использованы для настройки их поведения, если нас не устраивает их работа по умолчанию.

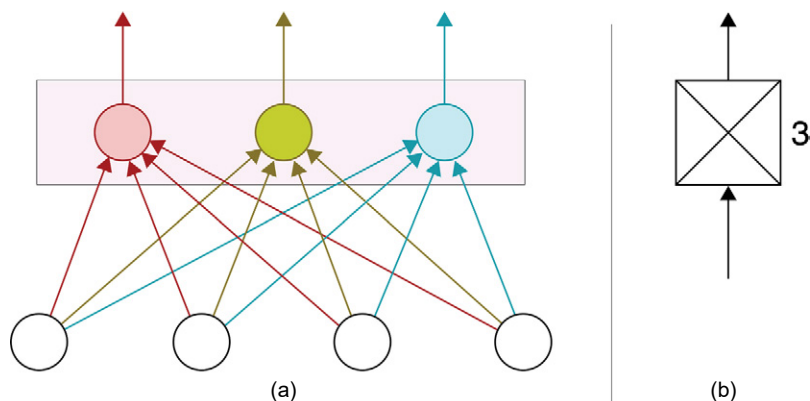
Одной из опций, доступных для большинства обрабатывающих слоев, является функция активации, применяемая на выходе нейронов. Вспомним из главы 17, что функция активации является небольшим нелинейным преобразованием, которое применяется на выходе каждого нейрона перед передачей его дальше. Хотя теоретически могут быть применены различные функции активации в каждом нейроне каждого слоя, это используется редко. На практике мы обычно применяем одну и ту же функцию активации в каждом нейроне данного слоя.

Последующий обзор сознательно сделан кратким. Мы вернемся к некоторым из этих слоев в отдельной главе, посвященной их принципам и использованию. Другие будут рассмотрены более детально при рассмотрении их применения.

### 20.4.1. Полносвязные слои

**Полносвязные слои** (также называемые **FC** (Fully-Connected), или **плотными** (dense)) – это набор нейронов, каждый из которых получает вход от каждого нейрона предыдущего слоя. Например, если имеется 4 нейрона в плотном слое и 4 нейрона в предыдущем слое, то каждый нейрон этого слоя будет иметь 4 входа, или всего  $4 \times 4 = 16$  соединений.

На рис. 20.6 показана диаграмма полносвязного слоя с тремя нейронами, следующего за слоем из 4 нейронов.



**Рис. 20.6** ❖ Полносвязный слой: (а) цветные нейроны образуют полносвязный слой. Каждый нейрон в верхнем слое получает вход от каждого нейрона в предыдущем слое; (б) наш схематический символ для полносвязного слоя

На рис. 20.6 (б) показан схематический символ, который мы будем использовать для плотных слоев. Идея заключается в том, что два нейрона вверх и вниз

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

[e-Univers.ru](http://e-Univers.ru)