

Отзывы о первом издании

«Самое понятное объяснение глубокого обучения, которое я когда-либо встречал... приятно и легко читается».

– *Ричард Тобиас, Cephasonics*

«Эта книга сокращает разрыв между идеями и работающей системой глубокого обучения».

– *Петр Рабинович, Akamai*

«Все основные темы и концепции глубокого обучения раскрыты и доходчиво объяснены с использованием примеров кода и графиков вместо математических формул».

– *Срджан Сантич, Springboard.com*

Оглавление

1	■	Что такое глубокое обучение?	24
2	■	Математические основы нейронных сетей	55
3	■	Введение в Keras и TensorFlow.....	104
4	■	Примеры работы с нейросетью: классификация и регрессия	143
5	■	Основы машинного обучения	173
6	■	Обобщенный рабочий процесс машинного обучения	215
7	■	Работа с Keras: углубленные навыки.....	239
8	■	Глубокое обучение в компьютерном зрении	277
9	■	Глубокое обучение для компьютерного зрения.....	321
10	■	Глубокое обучение и временные ряды	370
11	■	Глубокое обучение в обработке текстов	408
12	■	Генеративные модели глубокого обучения	484
13	■	Глубокое обучение в реальной жизни	547
14	■	Заключение.....	571

Содержание

Оглавление	6
Предисловие	15
Благодарности	17
Об этой книге	18
Об иллюстрации на обложке	22
Об авторах	23

1	Что такое глубокое обучение?	24
1.1	Искусственный интеллект, машинное и глубокое обучение	25
1.1.1	Искусственный интеллект	25
1.1.2	Машинное обучение	26
1.1.3	Извлечение правил и представлений из данных	28
1.1.4	«Глубина» глубокого обучения	31
1.1.5	Принцип действия глубокого обучения в трех рисунках	33
1.1.6	Каких успехов достигло глубокое обучение	35
1.1.7	Не верьте рекламной шумихе	36
1.1.8	Перспективы развития ИИ	37
1.2	Краткая история машинного обучения	38
1.2.1	Вероятностное моделирование	39
1.2.2	Первые нейронные сети	39
1.2.3	Ядерные методы	40
1.2.4	Деревья решений, случайные леса и градиентный бустинг	42
1.2.5	Назад к нейронным сетям	43
1.2.6	Отличительные черты глубокого обучения	44
1.2.7	Современный ландшафт машинного обучения	45
1.3	Почему глубокое обучение? Почему сейчас?	47
1.3.1	Оборудование	48
1.3.2	Данные	49
1.3.3	Алгоритмы	50
1.3.4	Новая волна инвестиций	51
1.3.5	Демократизация глубокого обучения	52
1.3.6	Ждать ли продолжения этой тенденции?	52

2	Математические основы нейронных сетей	55
2.1	Первое знакомство с нейронной сетью	56
2.2	Представление данных для нейронных сетей	60
2.2.1	Скаляры (тензоры нулевого ранга)	61
2.2.2	Векторы (тензоры первого ранга)	61
2.2.3	Матрицы (тензоры второго ранга)	62
2.2.4	Тензоры третьего и более высокого рангов	62
2.2.5	Ключевые атрибуты	62
2.2.6	Манипулирование тензорами в R	64
2.2.7	Пакеты данных	64
2.2.8	Практические примеры тензоров с данными	65
2.2.9	Векторные данные	65
2.2.10	Временные ряды, или последовательности данных	66
2.2.11	Изображения	67
2.2.12	Видеоданные	67
2.3	Шестеренки нейронных сетей: операции с тензорами	68
2.3.1	Поэлементные операции	69
2.3.2	Операции с тензорами разной размерности	70
2.3.3	Скалярное произведение тензоров	72
2.3.4	Изменение формы тензора	74
2.3.5	Геометрическая интерпретация операций с тензорами	75
2.3.6	Геометрическая интерпретация глубокого обучения	79
2.4	Механизм нейронных сетей: оптимизация на основе градиента	80
2.4.1	Что такое производная?	82
2.4.2	Производная операций с тензорами: градиент	83
2.4.3	Стохастический градиентный спуск	85
2.4.4	Объединение производных: алгоритм обратного распространения ошибки	88
2.5	Возвращаясь к нашему первому примеру	95
2.5.1	Повторная реализация нашего первого примера с нуля в TensorFlow	97
2.5.2	Выполнение одного шага обучения	99
2.5.3	Полный цикл обучения	101
2.5.4	Оценка модели	102
	Краткие итоги главы	103
3	Введение в Keras и TensorFlow	104
3.1	Что такое TensorFlow?	105
3.2	Что такое Keras?	106
3.3	Keras и TensorFlow: краткая история	107
3.4	Интерфейсы Python и R: краткая история	108
3.5	Настройка среды разработки для глубокого обучения	109
3.5.1	Установка Keras и TensorFlow	110
3.6	Первые шаги с TensorFlow	111
3.6.1	Тензоры TensorFlow	112
3.7	Атрибуты тензоров	113

3.7.1	Форма тензора и ее изменение	114
3.7.2	Срезы тензоров	116
3.7.3	Операции с тензорами разной размерности	117
3.7.4	Модуль <i>tf</i>	117
3.7.5	Неизменность тензоров и переменные	119
3.7.6	Математические операции в <i>TensorFlow</i>	120
3.7.7	Взгляд на <i>API GradientTape</i> с другой стороны	121
3.7.8	Полный пример: линейный классификатор в чистом <i>TensorFlow</i>	122
3.8	Анатомия нейронной сети и основы <i>API Keras</i>	127
3.8.1	Слои: строительные блоки глубокого обучения	128
3.8.2	От слоев к моделям	132
3.8.3	Этап «компиляции»: настройка процесса обучения	134
3.8.4	Выбор функции потерь	137
3.8.5	Использование метода <i>fit()</i>	138
3.8.6	Отслеживание потерь и показателей на контрольных данных	139
3.8.7	Использование модели после обучения	140
	Краткие итоги главы	141

4 Примеры работы с нейросетью: классификация и регрессия

4.1	Классификация отзывов к фильмам: пример бинарной классификации	145
4.1.1	Набор данных <i>IMDB</i>	145
4.1.2	Подготовка данных	147
4.1.3	Создание модели	148
4.1.4	Проверка вашего выбора	151
4.1.5	Использование обученной сети для прогнозирования на новых данных	154
4.1.6	Продолжаем эксперименты	155
4.1.7	Промежуточные итоги	155
4.2	Классификация новостных лент: пример многоклассовой классификации	156
4.2.1	Набор данных <i>Reuters</i>	156
4.2.2	Подготовка данных	158
4.2.3	Построение модели	158
4.2.4	Проверка модели	159
4.2.5	Предсказания на новых данных	161
4.2.6	Другой способ обработки меток и потерь	162
4.2.7	Важность использования достаточно больших промежуточных слоев	162
4.2.8	Дальнейшие эксперименты	163
4.2.9	Промежуточные итоги	163
4.3	Предсказание цен на дома: пример регрессии	164
4.3.1	Набор данных с ценами на жилье в Бостоне	164
4.3.2	Подготовка данных	165
4.3.3	Построение модели	165

4.3.4	Оценка качества модели методом K -кратной перекрестной проверки	166
4.3.5	Выдача прогнозов на новых данных	171
4.3.6	Промежуточные выводы	171
	Краткие итоги главы.....	171

5	Основы машинного обучения	173
5.1	Обобщение – цель машинного обучения	173
5.1.1	Недообучение и переобучение	174
5.1.2	Базовые принципы обобщения в глубоком обучении	180
5.2	Оценка моделей машинного обучения	187
5.2.1	Наборы данных для обучения, проверки и контроля.....	187
5.2.2	Использование критериев, основанных на здравом смысле	191
5.2.3	Что следует помнить об оценке модели	192
5.3	Улучшение качества обучения модели	193
5.3.1	Настройка ключевых параметров градиентного спуска.....	193
5.3.2	Использование лучшей априорно обоснованной архитектуры	196
5.3.3	Увеличение емкости модели	197
5.4	Как улучшить обобщение	199
5.4.1	Подготовка набора данных	199
5.4.2	Конструирование признаков	200
5.4.3	Использование ранней остановки	202
5.4.4	Регуляризация модели	202
	Краткие итоги главы.....	213

6	Обобщенный рабочий процесс машинного обучения	215
6.1	Постановка задачи.....	217
6.1.1	Уточнение задачи	217
6.1.2	Получение исходных данных.....	219
6.1.3	Добейтесь понимания данных.....	223
6.1.4	Выберите меру успеха.....	224
6.2	Разработка модели.....	225
6.2.1	Подготовка данных	225
6.2.2	Выбор протокола оценки	227
6.2.3	Как превзойти простой базовый уровень.....	228
6.2.4	Масштабирование: разработка модели, способной к переобучению	229
6.2.5	Регуляризация и настройка модели.....	230
6.3	Развертывание модели	231
6.3.1	Представление модели заказчику.....	231
6.3.2	Передача модели заказчику	232
6.3.3	Мониторинг модели в рабочей среде.....	236
6.3.4	Поддержка и обновление модели.....	236
	Краткие итоги главы.....	237

7	Работа с Keras: углубленные навыки	239
7.1	Широкий спектр рабочих процессов Keras	240
7.2	Различные способы построения моделей Keras	240
7.2.1	<i>Sequential API</i>	241
7.2.2	<i>Functional API</i>	244
7.2.3	Создание подкласса класса Model	251
7.2.4	Смешивание и сочетание разных компонентов	255
7.2.5	Используйте правильные инструменты	256
7.3	Использование встроенных циклов обучения и оценки	256
7.3.1	Разработка собственных метрик.....	257
7.3.2	Использование обратных вызовов	260
7.3.3	Разработка собственных обратных вызовов	262
7.3.4	Мониторинг и визуализация с помощью <i>TensorBoard</i>	264
7.4	Разработка собственных циклов обучения и оценки	266
7.4.1	Обучение или логический вывод.....	267
7.4.2	Использование метрик на низком уровне.....	268
7.4.3	Полный цикл обучения и оценки	269
7.4.4	Увеличьте быстрдействие с помощью tf.function()	272
7.4.5	Использование fit() с пользовательским циклом обучения.....	273
	Краткие итоги главы.....	276
8	Глубокое обучение в компьютерном зрении	277
8.1	Введение в сверточные нейронные сети	278
8.1.1	Операция свертки	281
8.1.2	Выбор максимального значения из соседних (<i>max-pooling</i>).....	286
8.2	Обучение сверточной нейронной сети с нуля на небольшом наборе данных	289
8.2.1	Целесообразность глубокого обучения для решения задач с небольшими наборами данных	290
8.2.2	Загрузка данных.....	290
8.2.3	Построение сети.....	293
8.2.4	Предварительная обработка данных	295
8.2.5	Расширение данных	301
8.3	Использование предварительно обученной сверточной нейронной сети	305
8.3.1	Выделение признаков	306
8.3.2	Дообучение ранее обученной модели	316
	Краткие итоги главы.....	320
9	Глубокое обучение для компьютерного зрения	321
9.1	Три основные задачи компьютерного зрения.....	322
9.2	Пример сегментации изображения.....	323
9.3	Современные стандартные архитектуры сверточных сетей	333
9.3.1	Модульность, иерархия и повторное использование	334
9.3.2	Остаточные связи	337

9.3.3	Пакетная нормализация	341
9.3.4	Разделяемые по глубине свертки	344
9.3.5	Применим знания на практике: мини-модель, подобная Xception	347
9.4	Интерпретация знаний сверточной нейросети	350
9.4.1	Визуализация промежуточных активаций	351
9.4.2	Визуализация сетевых фильтров	357
9.4.3	Визуализация тепловых карт активации класса	363
	Краткие итоги главы	369

10 Глубокое обучение и временные ряды.....370

10.1	Различные виды задач временных рядов.....	370
10.2	Пример прогнозирования температуры.....	372
10.2.1	Подготовка данных.....	376
10.2.2	Простое решение задачи без привлечения машинного обучения.....	380
10.2.3	Решение с использованием базовой модели машинного обучения.....	382
10.2.4	Эксперимент с одномерной сверточной сетью	384
10.2.5	Первый вариант простой рекуррентной модели	387
10.3	Рекуррентные нейронные сети	388
10.3.1	Рекуррентный слой в Keras.....	391
10.4	Продвинутое применение рекуррентных нейронных сетей ...	396
10.4.1	Использование рекуррентного прореживания для борьбы с переобучением.....	397
10.4.2	Наложение рекуррентных слоев	400
10.4.3	Использование двунаправленных рекуррентных сетей	402
10.4.4	Что дальше	405
	Краткие итоги главы.....	407

11 Глубокое обучение в обработке текстов.....408

11.1	Обработка естественного языка: обзор отрасли	408
11.2	Подготовка текстовых данных.....	411
11.2.1	Стандартизация текста	412
11.2.2	Разделение текста (токенизация).....	413
11.2.3	Индексация словаря.....	414
11.2.4	Использование слоя <code>layer_text_vectorization</code>	416
11.3	Два подхода к представлению групп слов: наборы и последовательности	420
11.3.1	Подготовка данных обзоров фильмов IMDB.....	421
11.3.2	Обработка слов без учета порядка	424
11.3.3	Обработка последовательности слов	432
11.4	Архитектура Transformer.....	446
11.4.1	Механизм самовнимания	446
11.4.2	Многоголовное внимание	452
11.4.3	Кодировщик в архитектуре Transformer	453
11.4.4	Когда следует использовать модели последовательности, а не модели мешка слов.....	463

11.5	Помимо классификации текста: обучение преобразованию последовательностей.....	464
11.5.1	Пример машинного перевода	466
11.5.3	Рекуррентная модель преобразования последовательностей	469
11.5.4	Преобразование последовательностей с Transformer	476
	Краткие итоги главы.....	482
12	Генеративные модели глубокого обучения	484
12.1	Генерирование текста с помощью Keras.....	486
12.1.1	Краткая история генеративных сетей.....	486
12.1.2	Как генерируют последовательности данных?.....	488
12.1.3	Важность стратегии выбора	488
12.1.4	Реализация генерации текста с помощью Keras.....	491
12.1.5	Обратный вызов генерации текста с выборкой при разной температуре	495
12.1.6	Подведение итогов	502
12.2	DeepDream	502
12.2.1	Реализация DeepDream в Keras.....	503
12.2.2	Подведение итогов	511
12.3	Нейронный перенос стиля.....	512
12.3.1	Функция потерь содержания.....	513
12.3.2	Функция потерь стиля	513
12.3.3	Реализация переноса стиля в Keras.....	514
12.3.4	Подведение итогов	522
12.4	Генерация изображений с помощью вариационных автокодировщиков	522
12.4.1	Выбор шаблонов из скрытых пространств изображений	523
12.4.2	Концептуальные векторы для редактирования изображений	524
12.4.3	Вариационные автокодировщики.....	525
12.4.4	Реализация VAE с помощью Keras	528
12.4.5	Подведение итогов	534
12.5	Введение в генеративно-состязательные сети	534
12.5.1	Реализация генеративно-состязательной сети	536
12.5.2	Полезные технические приемы	537
12.5.3	Получение набора данных CelebA.....	538
12.5.4	Дискриминатор.....	540
12.5.5	Генератор.....	541
12.5.6	Состязательная сеть	542
12.5.7	Подведение итогов	545
	Краткие итоги главы.....	546
13	Глубокое обучение в реальной жизни	547
13.1	Получение максимальной отдачи от ваших моделей	548
13.1.1	Оптимизация гиперпараметров	548
13.1.2	Ансамблирование моделей.....	557
13.2	Масштабируемое обучение моделей.....	559

13.2.1	Ускорение обучения на GPU со смешанной точностью	560
13.2.2	Обучение модели на нескольких GPU	563
13.2.3	Обучение модели на TPU	568
	Краткие итоги главы	570

14	Заключение	571
14.1	Краткий обзор ключевых понятий	572
14.1.1	Различные подходы к ИИ	572
14.1.2	Что выделяет глубокое обучение среди других подходов к машинному обучению	573
14.1.3	Как правильно воспринимать глубокое обучение	573
14.1.4	Ключевые технологии глубокого обучения	575
14.1.5	Обобщенный рабочий процесс машинного обучения	576
14.1.6	Основные архитектуры сетей	577
14.1.7	Пространство возможностей	582
14.2	Ограничения глубокого обучения	584
14.2.1	Риск очеловечивания моделей глубокого обучения	585
14.2.2	Принципиальное различие между автоматом и интеллектом	587
14.2.3	Различие между локальным и экстремальным обобщением	589
14.2.4	Предназначение интеллекта	592
14.2.5	Восхождение по уровням обобщения	593
14.3	Курс на большую универсальность в ИИ	594
14.3.1	О важности постановки правильной цели: правило короткого пути	594
14.3.2	Новая цель	597
14.4	Реализация интеллекта: недостающие ингредиенты	599
14.4.1	Построение и использование абстрактных аналогий	599
14.4.2	Два полюса абстракции	601
14.4.3	Сочетание двух полюсов абстракции	604
14.4.4	Недостающая половина картинки	604
14.5	Будущее глубокого обучения	606
14.5.1	Модели как программы	606
14.5.2	Машинное обучение и синтез программ	608
14.5.3	Сочетание глубокого обучения и синтеза программ	608
14.5.4	Непрерывное обучение и повторное использование модульных подпрограмм	611
14.5.5	Долгосрочная перспектива	612
14.6	Как не отстать от прогресса в быстро развивающейся отрасли	614
14.6.1	Решения реальных задач на сайте Kaggle	614
14.6.2	Знакомство с последними разработками на сайте arXiv	614
14.6.3	Исследование экосистемы Keras	615
14.7	Заключительное слово	616
	Приложение. Введение в Python для пользователей R	617
	Предметный указатель	641

Предисловие

Если вы решили приобрести эту книгу, то наверняка слышали о небывалом успехе методики глубокого обучения в области искусственного интеллекта. Мы прошли путь от почти бесполезного распознавания образов и речи до невероятно эффективного решения этих задач. Последствия такого внезапного прогресса отразились почти повсеместно. Сегодня мы применяем глубокое обучение для решения целого ряда важных задач в таких разных областях, как визуализация медицинских данных, сельское хозяйство, автономное вождение, образование, предотвращение стихийных бедствий и промышленное производство.

Тем не менее, я считаю, что глубокое обучение все еще находится в зачаточном состоянии. Пока оно реализовало лишь малую часть своего потенциала. Со временем глубокое обучение проникнет в каждую область, где может принести пользу, – трансформация, которая займет не одно десятилетие.

Однако для того чтобы начать внедрение технологии глубокого обучения во все задачи, которые можно решить с ее применением, мы должны сделать ее доступной как можно большему числу людей, включая неспециалистов, – то есть тех, кто не является инженером-исследователем или аспирантом. Чтобы раскрыть весь потенциал глубокого обучения, мы должны полностью демократизировать его. И сегодня я считаю, что мы находимся на пороге исторического перелома, когда глубокое обучение выходит из академических лабораторий и отделов исследований и разработок крупных технологических компаний, чтобы стать обыденной частью набора инструментов каждого разработчика – очень похоже на историю веб-технологий в конце 1990-х. Сейчас почти любой желающий может создать веб-сайт или веб-приложение для своего бизнеса или сообщества, хотя

в 1998 году для этого потребовалась бы команда специалистов. В недалеком будущем любой, у кого есть идея и базовые навыки программирования, сможет создавать интеллектуальные приложения, которые обучаются на основе данных.

Когда в марте 2015 года я выпустил первую версию фреймворка глубокого обучения Keras, я не стремился сделать общедоступным искусственный интеллект (ИИ). Я несколько лет занимался исследованиями в области машинного обучения и создал Keras для использования в собственных экспериментах. Но с 2015 года в область глубокого обучения пришли сотни тысяч новичков; многие из них выбрали Keras в качестве любимого инструмента. Наблюдая за тем, как множество новичков и опытных специалистов используют Keras самыми неожиданными и эффективными способами, я пришел к выводу, что нужно задуматься о доступности и демократизации ИИ. Я осознал, что чем шире мы будем распространять эти технологии, тем ценнее они будут становиться. Доступность быстро стала одной из главных целей Keras, и за несколько лет сообществу разработчиков удалось добиться фантастических достижений в этом направлении. Мы «вручили» технологию глубокого обучения сотням тысяч людей, и они, в свою очередь, воспользовались ею для решения важных проблем, которые до недавнего времени считались неразрешимыми.

Книга, которую вы держите, – еще один шаг на пути к тому, чтобы сделать глубокое обучение доступным как можно большему количеству людей. Фреймворк Keras всегда нуждался в сопроводительном курсе, который одновременно освещал бы основы глубокого обучения, показывал примеры его использования и демонстрировал лучшие практики в применении Keras. В 2016 и 2017 годах я приложил изрядные усилия, что создать такой курс. Он лег в основу первого издания этой книги, выпущенной в декабре 2017 года. Книга быстро стала бестселлером по машинному обучению, разошлась тиражом более 50 000 экземпляров и была переведена на 12 языков.

Однако область глубокого обучения быстро развивается. С момента публикации первого издания произошло много важных событий – выпуск TensorFlow 2, растущая популярность архитектуры Transformer и многое другое. Поэтому в конце 2019 года я решил обновить свою книгу. Сначала я наивно думал, что смогу обойтись обновлением около 50 % контента, и объем второго издания почти не изменится. На самом деле после двух лет работы новая редакция оказалась более чем на треть длиннее, в ней 75 % нового материала. Это больше, чем обновление, это совершенно новая книга.

Я писал ее, стараясь максимально доступно объяснить идеи, лежащие в основе глубокого обучения и его реализации. Это не значит, что я преднамеренно упрощал изложение – я искренне убежден, что в теме глубокого обучения нет ничего сложного. Надеюсь, эта книга принесет вам пользу и поможет начать создавать интеллектуальные приложения и решать важные для вас задачи.

Об этой книге

Эта книга предназначена для всех, кто хочет освоить глубокое обучение с нуля или расширить свои знания о глубоком обучении. Независимо от того, являетесь ли вы инженером по разработке систем машинного обучения, специалистом по данным или студентом университета, вы найдете для себя много полезного на страницах этой книги.

Вы будете изучать глубокое обучение наиболее эффективным способом – начиная с простых понятий, а затем переходя к самым современным методам. Вы убедитесь, что эта книга обеспечивает баланс между интуитивным знанием, теорией и практикой. Мы старались как можно меньше использовать математические формулы, предпочитая вместо этого объяснять основные идеи машинного и глубокого обучения с помощью подробных фрагментов кода и интуитивно понятных образных моделей. Из многочисленных примеров кода, снабженных подробными комментариями, практических рекомендаций и простых объяснений вы получите знания, которых достаточно, чтобы использовать глубокое обучение для решения прикладных задач.

В примерах кода мы используем платформу глубокого обучения Keras с TensorFlow 2 в качестве вычислительного движка. Примеры демонстрируют лучшие известные нам приемы использования Keras и TensorFlow 2 по состоянию на 2022 год.

Прочитав эту книгу, вы получите четкое представление о том, что такое глубокое обучение, когда его следует применять, и каковы его ограничения. Вы познакомитесь со стандартным рабочим процессом поиска решения задачи машинного обучения, а также узнаете, как устранять часто возникающие проблемы. Вы научитесь использовать Keras для решения самых разнообразных прикладных задач, начиная с компьютерного зрения и заканчивая обработкой есте-

ственного языка – среди них классификация изображений, сегментация изображений, прогнозирование временных рядов, классификация текста, машинный перевод, генерация текста и многое другое.

Кому адресована эта книга

Эта книга написана для людей с опытом программирования на R, желающих начать знакомство с темой машинного обучения с технологии глубокого обучения. Но она также может быть полезной и другим категориям читателей:

- если вы специалист по обработке и анализу данных, знакомый с машинным обучением, эта книга позволит вам получить достаточно полное практическое представление о глубоком обучении, наиболее быстро развивающемся направлении в области машинного обучения;
- если вы исследователь или прикладной специалист в области глубокого обучения, желающий освоить фреймворк Keras, вы найдете в этой книге лучший углубленный курс по Keras;
- если вы аспирант, изучающий технологии глубокого обучения в ходе обязательного курса, в этой книге вы найдете практическое дополнение к своим учебникам, которое поможет вам лучше понять принцип действия нейросетей и познакомит с наиболее эффективными приемами.

Даже люди с техническим складом ума, которые не занимаются программированием регулярно, найдут эту книгу полезной для знакомства с базовыми и продвинутыми понятиями глубокого обучения.

Для понимания примеров кода вам понадобится знание языка R на среднем уровне. Не обязательно иметь опыт работы с машинным или глубоким обучением: эта книга охватывает все необходимые основы с нуля. Не требуется также иметь какой-то особенной математической подготовки — вполне достаточно знания математики на уровне средней школы.

О примерах кода

Эта книга содержит большое количество примеров исходного кода как в пронумерованных листингах, так и в виде обычного текста. В обоих случаях исходный код представлен шрифтом фиксированной ширины, чтобы он отличался от обычного текста. Вывод работающего кода аналогичным образом отформатирован шрифтом фиксированной ширины, но также снабжен вертикальной серой полосой слева. На протяжении всей книги вы найдете код и выходные данные кода, чередующиеся следующим образом:

```
print("R is awesome!")
```

| [1] "R is awesome!"

Во многих случаях исходный код был переформатирован; нам пришлось добавить разрывы строк и изменить отступы, чтобы код умещался на доступном пространстве страницы. В редких случаях этого было недостаточно, и некоторые листинги содержат маркеры продолжения строки (↪). Кроме того, многие комментарии к исходному коду были удалены из листингов, если код подробно описан в тексте. Многие листинги содержат дополнительные примечания, указывающие на важные нюансы кода.

Вы можете загрузить все примеры кода с сайта книги по адресу <https://livebook.manning.com/book/deep-learning-with-r-second-edition/> или с репозитория GitHub <https://github.com/t-kalinowski/deep-learning-with-R-2nd-edition-code>, а также на сайте издательства «ДМК Пресс» по адресу dmpress@gmail.com.

Об авторах

Франсуа Шолле (François Chollet) – создатель Keras, одного из наиболее популярных фреймворков глубокого обучения. В настоящее время он работает инженером-программистом в Google, где возглавляет команду Keras. Кроме того, он занимается исследованиями в области абстракции, рассуждений и способов достижения большей универсальности в искусственном интеллекте.

Томаш Калиновски (Tomasz Kalinowski) работает инженером-программистом в RStudio, где занимается сопровождением пакетов TensorFlow и Keras R. На предыдущих должностях он работал ученым и инженером, применяя машинное обучение к широкому спектру наборов данных и предметных областей.

Дж. Дж. Аллер (J. J. Allaire) – основатель RStudio и создатель интегрированной среды разработки RStudio IDE. Является автором интерфейса R к библиотекам TensorFlow и Keras.

1

Что такое глубокое обучение?

Эта глава охватывает следующие темы:

- обобщенные определения основных понятий;
- история развития машинного обучения;
- ключевые факторы роста популярности глубокого обучения и потенциал развития.

За последние несколько лет тема искусственного интеллекта (ИИ) вызвала большую шумиху в средствах массовой информации. Машинное обучение, глубокое обучение и ИИ упоминались в бесчисленном количестве статей, многие из которых никак не связаны с описанием технологий. Нам обещали появление виртуальных собеседников, автомобилей с автопилотом и виртуальных помощников. Иногда будущее рисовали в мрачных тонах, а иногда изображали утопическим: освобождение людей от рутинного труда и выполнение основной работы роботами, наделенными искусственным интеллектом. Будущему или сегодняшнему специалисту в области машинного обучения важно уметь выделять полезный сигнал из шума, видеть в раздутых пресс-релизах изменения, действительно способные повлиять на мир. Наше будущее поставлено на карту, и вам предстоит сыграть в нем активную роль: закончив чтение этой книги, вы войдете в ряды тех, кто разрабатывает системы ИИ. Потому давайте рассмотрим следующие вопросы. Чего уже достигло глубокое обуче-

ние? Насколько это важно? В каком направлении пойдет дальнейшее развитие? Можно ли верить поднятой шумихе?

Эта глава закладывает фундамент для дальнейшего обсуждения ИИ, машинного и глубокого обучения.

1.1 Искусственный интеллект, машинное и глубокое обучение

Прежде всего, определим, что подразумевается под искусственным интеллектом. Что такое ИИ, машинное и глубокое обучение (рис. 1.1)? Как они связаны друг с другом?



Рис. 1.1 Искусственный интеллект, машинное и глубокое обучение

1.1.1 Искусственный интеллект

Идея искусственного интеллекта появилась в 1950-х, когда группа энтузиастов из только зарождающейся области информатики задалась вопросом, можно ли заставить компьютеры «думать», – вопросом, последствия которого мы изучаем до сих пор.

Хотя многие лежащие в основе ИИ идеи зародились в предшествующие годы и даже десятилетия, «искусственный интеллект» окончательно оформился как область исследований в 1956 году, когда Джон Маккарти, в то время молодой доцент кафедры математики в Дартмутском колледже, организовал летний семинар с весьма амбициозными задачами:

Исследование должно опираться на гипотезу о том, что каждый аспект обучения или любое другое свойство интеллекта в принципе поддаются настолько точному описанию, что их можно будет смоделировать с помощью машины. Будут предприняты попытки найти способ, как заставить машины использовать язык, формировать абстракции и концепции, решать задачи, традиционно остающиеся уде-

лом людей, и улучшать себя. Мы думаем, что можно добиться значительного прогресса в решении одной или нескольких из этих проблем, если тщательно подобранная группа ученых будет работать над ней вместе в течение лета.

В конце лета семинар завершился, а исчерпывающее решение задачи так и не удалось найти. Тем не менее, на нем присутствовало много исследователей, которые впоследствии стали пионерами в области ИИ, и фактически он запустил интеллектуальную революцию, которая продолжается и по сей день.

Коротко эту область можно определить так: *автоматизация интеллектуальных задач, обычно выполняемых людьми*. Соответственно, ИИ – это область, охватывающая машинное обучение и глубокое обучение, а также включающая многие подходы, не связанные с обучением. Только представьте, что вплоть до 1980-х в большинстве книг про ИИ вообще не упоминалось обучение! Например, первые программы для игры в шахматы действовали по жестко определенным правилам, заданным программистами, и не могли квалифицироваться как осуществляющие машинное обучение. Долгое время многие эксперты полагали, что искусственный интеллект уровня человека можно создать, если дать программисту достаточный набор явных правил для манипулирования знаниями. Этот подход известен как символический ИИ и являлся доминирующей парадигмой ИИ с 1950-х до конца 1980-х. Пик его популярности пришелся на бум *экспертных систем* в 1980-х.

Символический ИИ прекрасно справлялся с решением четко определенных логических задач, таких как игра в шахматы, но, как оказалось, невозможно задать строгие правила для решения более сложных, нечетких задач, таких как классификация изображений, распознавание речи и перевод на другие языки. На смену символическому ИИ пришел новый подход: *машинное обучение*.

1.1.2 Машинное обучение

Много лет назад в викторианской Англии жила леди Ада Лавлейс – друг и соратник Чарльза Бэббиджа, изобретателя аналитической вычислительной машины, первого известного механического компьютера. Несомненно, аналитическая машина опередила свое время, но она не задумывалась как универсальный компьютер, когда разрабатывалась в 1830-х и 1840-х, потому что идея универсальных вычислений еще не родилась. Эта машина просто давала возможность использовать механические операции для автоматизации некоторых вычислений из области математического анализа, что и обусловило такое ее название. Надо сказать, что аналитическая машина была интеллектуальным потомком более ранних попыток кодирования математических операций в форме шестерни, таких

как *паскалина* или ступенчатый счетный механизм Лейбница – усовершенствованная версия паскалины. Разработанная Блезом Паскалем в 1642 году (в возрасте 19 лет!), паскалина была первым в мире механическим калькулятором, который мог складывать, вычитать, умножать и даже делить числа.

В 1843 году Ада Лавлейс заметила:

Аналитическая машина не может создавать что-то новое. Она может делать все, что мы и сами знаем, как выполнять... ее цель состоит лишь в том, чтобы помогать нам выполнять то, с чем мы уже хорошо знакомы.

Даже с учетом 179-летней исторической перспективы наблюдения леди Лавлейс не перестает наводить на размышления. Может ли компьютер общего назначения «создать» что-нибудь или он всегда будет вынужден тупо выполнять процессы, которые мы, люди, полностью понимаем? Родятся ли у него когда-нибудь оригинальные мысли? Может ли он учиться на собственном опыте? Может ли он проявить креативность?

Позднее пионер ИИ Алан Тьюринг в своей знаменитой статье «Computing Machinery and Intelligence»¹ назвал это замечание «аргументом Ады Лавлейс». В этой статье был представлен тест Тьюринга, а также перечислены основные идеи, которые могут привести к созданию ИИ². Тьюринг пришел к выводу – очень провокационному на то время – что в принципе компьютеры способны имитировать все аспекты человеческого интеллекта.

Обычный способ заставить компьютер выполнять полезную работу – это попросить человека-программиста написать *правила* – компьютерную программу, которой нужно следовать, чтобы преобразовать входные данные в соответствующие ответы, точно так же, как леди Лавлейс записывала пошаговые инструкции для аналитической машины. В машинном обучении люди вводят данные и ответы, соответствующие этим данным, а на выходе получают правила. Эти правила затем можно применить к новым данным для получения оригинальных ответов. В машинном обучении система *обучается*, а не программируется явно. Ей передаются многочисленные примеры, имеющие отношение к решаемой задаче, а она находит в этих примерах статистическую структуру, которая позволяет системе выработать правила для автоматического решения задачи (рис. 1.2). Например, чтобы автоматизировать задачу определения фотографий, сделанных в отпуске, можно передать системе машинного обучения множество примеров фотографий, уже классифици-

¹ A. M. Turing. *Computing Machinery and Intelligence*. Mind 59, no. 236 (1950): 433–460.

² Хотя тест Тьюринга часто воспринимают, как буквальную цель, которую должен достичь ИИ, сам Тьюринг просто использовал его как концептуальный прием в философской дискуссии о природе познания.

рованных людьми, и система изучит статистические правила классификации конкретных фотографий.

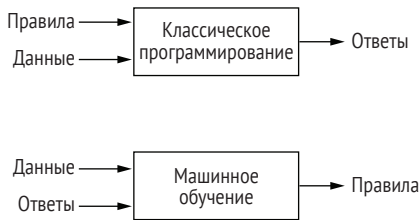


Рис. 1.2 Машинное обучение: новая парадигма программирования

Расцвет машинного обучения начался только в 1990-х, но эта область быстро превратилась в наиболее популярный и успешный раздел ИИ, и эта тенденция была подкреплена появлением более быстродействующей аппаратуры и огромных наборов данных. Машинное обучение тесно связано с математической статистикой, но имеет несколько важных отличий. В отличие от статистики, машинное обучение обычно имеет дело с большими и сложными наборами данных (например, состоящими из миллионов фотографий, каждая из которых состоит из десятков тысяч пикселей), к которым практически невозможно применить классические методы статистического анализа, такие как байесовские методы. Как результат, машинное и в особенности глубокое обучение имеют ограниченную математическую базу – возможно, слишком ограниченную – и основываются почти исключительно на инженерных решениях. В отличие от теоретической физики или математики, машинное обучение – это прикладная область, основанная на эмпирических данных и сильно зависящая от достижений в области программного и аппаратного обеспечения.

1.1.3 Извлечение правил и представлений из данных

Чтобы дать определение *глубокому обучению* и понять разницу между глубоким обучением и другими методами машинного обучения, сначала нужно получить некоторое представление о том, что делают алгоритмы машинного обучения. Чуть выше отмечалось, что машинное обучение выявляет правила решения задач обработки данных по примерам ожидаемых результатов. То есть для машинного обучения нам нужны три составляющие:

- *входные данные* – например, если решается задача распознавания речи, такими входными данными могут быть файлы с записью речи разных людей. Если решается задача классификации изображений, такими данными могут быть изображения.
- *примеры ожидаемых результатов* – в задаче распознавания речи это могут быть транскрипции звуковых файлов, составленные

людьми. В задаче классификации изображений ожидаемым результатом могут быть теги, такие как «собака», «кошка» и др.

- *способ оценки качества работы алгоритма* – это необходимо для определения, насколько далеко отклоняются результаты, возвращаемые алгоритмом, от ожидаемых. Оценка используется как сигнал обратной связи для корректировки работы алгоритма. Этот этап корректировки мы и называем обучением.

Модель машинного обучения трансформирует исходные данные в значимые результаты, «обучаясь» на известных примерах входных данных и результатов. То есть главной задачей машинного и глубокого обучения является *значимое преобразование данных*, или, иными словами, обучение *представлению* входных данных, приближающему нас к ожидаемому результату.

Прежде чем двинуться дальше, давайте определим, что есть представление данных? По сути, это другой способ взглянуть на данные – иначе их представить или закодировать. Например, цветное изображение можно закодировать в формате RGB (red-green-blue – красный-зеленый-синий) или HSV (hue-saturation-value – тон-насыщенность-значение): это два разных представления одних и тех же данных. Некоторые задачи трудно решаются с данными в одном представлении, но легко – в другом. Например, задача «выбрать все красные пиксели в изображении» легче решается с данными в формате RGB, тогда как задача «сделать изображение менее насыщенным» проще решается с данными в формате HSV. Главная задача моделей машинного обучения как раз и заключается в поиске соответствующего представления входных данных – преобразований, которые сделают данные более пригодными для решения задачи, такой как классификация.

Обратимся к конкретному примеру. Рассмотрим систему координат с осями X и Y и несколько точек в этой системе координат (x, y), как показано на рис. 1.3.

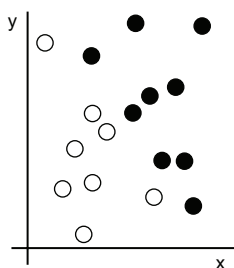


Рис. 1.3 Пример некоторых данных

Как видите, у нас имеется несколько белых и черных точек. Допустим нам нужно разработать алгоритм, принимающий координаты (x, y) точки и возвращающий наиболее вероятный цвет, черный или белый. В данном случае мы располагаем следующими данными:

- входными данными являются координаты точек;
- ожидаемым результатом является цвет точек;
- мерой качества алгоритма может быть, например, процент правильно классифицированных точек.

В данном случае нам нужно получить новый способ представления исходных данных, позволяющий четко отделять белые точки от черных. Таким преобразованием, кроме прочих других, могло бы быть изменение системы координат, как показано на рис. 1.4.



Рис. 1.4 Изменение системы координат

Координаты наших точек в этой новой системе координат можно назвать новым представлением данных. Причем более удачным! Задачу классификации данных «черный/белый» в этом представлении можно свести к простому правилу: «черные точки имеют координату $x > 0$ » или «белые точки имеют координату $x < 0$ ». Это новое представление, основанное на простом правиле, фактически решает поставленную задачу.

В этом случае мы определили изменение координат вручную – мы использовали наш человеческий интеллект, чтобы придумать собственное подходящее представление данных. Это нормально для такой чрезвычайно простой задачи, но сможете ли вы сделать то же самое, если перед вами стоит задача классификации изображений рукописных цифр? Сможете ли вы записать явные, исполняемые компьютером преобразования изображений, которые четко выделяют различия между 6 и 8 или между 1 и 7 для любого почерка?

В какой-то степени это возможно. Правила, основанные на представлениях цифр, таких как «количество замкнутых циклов» или вертикальные и горизонтальные пиксельные гистограммы, могут неплохо различать рукописные цифры. Но поиск таких полезных представлений вручную – тяжелая работа, и, как вы легко догадаетесь, полученная система, основанная на подобных правилах, очень неустойчива. Поддерживать работоспособность такой системы – сущий кошмар. Каждый раз, когда вы сталкиваетесь с новым примером почерка, нарушающим ваши тщательно продуманные правила, вам придется добавлять новые преобразования данных и новые правила, при этом учитывая их взаимодействие с каждым ранее добавленным правилом.

Наверняка вы уже подумали, что это сложный и болезненный процесс нужно автоматизировать. Что, если мы попытаемся целенаправленно искать различные наборы автоматически сгенерированных представлений данных и основанных на них правил, и выявлять лучшие из них, используя в качестве обратной связи процент правильно классифицированных цифр в некотором наборе эталонных данных? Так вот, это и есть машинное обучение! Под машинным обучением мы понимаем процесс автоматического поиска преобразований данных, которые создают полезные представления, руководствуясь некоторым сигналом обратной связи – такие представления, которые поддаются более простым правилам, решающим поставленную задачу.

В качестве таких преобразований могут выступать изменения координат (как в нашем примере с классификацией точек) или получение гистограммы пикселей и циклов обхода (как в нашем примере с классификацией цифр), но также это могут быть линейные проекции, переносы, нелинейные операции (например, правило «выбрать все точки такие, что $x > 0$ ») и т. д. Алгоритмы машинного обучения обычно не очень изобретательны в поиске этих преобразований; они просто просматривают predetermined набор операций, называемый *пространством гипотез*. Например, пространство всех возможных изменений координат будет нашим пространством гипотез в примере классификации точек.

Иными словами, машинное обучение – это поиск значимого представления некоторых входных данных в predetermined пространстве возможностей с использованием сигнала обратной связи. Эта простая идея позволяет решать чрезвычайно широкий круг интеллектуальных задач: от распознавания речи до автоматического вождения автомобиля. Теперь, когда вы получили представление о том, что понимается под *обучением*, давайте посмотрим, что особенного в *глубоком обучении*.

1.1.4 «Глубина» глубокого обучения

Глубокое обучение – это особый раздел машинного обучения: новый подход к поиску представления данных, делающий упор на изучение последовательных слоев (или уровней) все более значимых представлений. Под «глубиной» в глубоком обучении не подразумевают более глубокое понимание, достигаемое этим подходом; идея заключается в многослойном представлении. Количество слоев, на которые делится модель данных, называют *глубиной* модели. Другими подходящими названиями для этой области машинного обучения могли бы служить: *многослойное обучение* и *иерархическое обучение*. Современные модели глубокого обучения часто состоят из десятков и даже сотен последовательных слоев представления, и все они автоматически определяются под воздействием обучающих данных. Кроме того, существуют другие подходы к машинному обучению,

ориентированные на изучение одного-двух слоев представления данных; по этой причине их иногда называют *поверхностным обучением* (shallow learning).

В глубоком обучении такие многослойные представления изучаются с использованием моделей, которые называют *нейронными сетями*, структурированных в виде слоев, наложенных друг на друга. Термин «нейронная сеть» заимствован из нейробиологии, тем не менее хотя некоторые основополагающие идеи глубокого обучения отчасти заимствованы из науки о мозге (в частности из зрительной оболочки), модели глубокого обучения не являются моделями мозга. Нет никаких доказательств, что мозг реализует механизмы, подобные механизмам, используемым в современных моделях глубокого обучения. Вам могут встретиться научно-популярные статьи, где утверждается, что глубокое обучение работает подобно мозгу или моделирует работу мозга, но в действительности это не так. Было бы неправильно и контрпродуктивно заставлять начинающих освоение этой области думать, что глубокое обучение каким-то образом связано с нейробиологией; вам не нужно представление «как наш мозг», и вы также можете забыть все, что читали о гипотетической связи между глубоким обучением и биологией. Намного продуктивнее считать глубокое обучение математической основой для изучения представлений данных.

Как выглядят представления, получаемые алгоритмом глубокого обучения? Давайте исследуем, как сеть, имеющая несколько слоев в глубину (рис. 1.5), преобразует изображение цифры для ее распознавания.

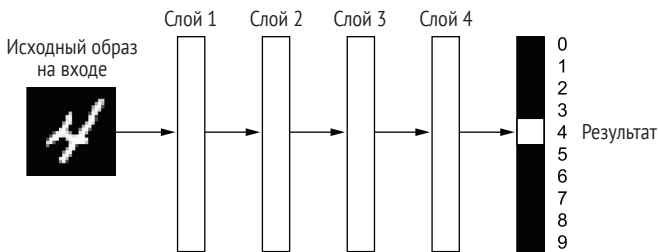


Рис. 1.5 Глубокая нейронная сеть для классификации цифр

Как показано на рис. 1.6, сеть поэтапно преобразует образ цифры в представление, все больше отличающееся от исходного образа и несущее все больше полезной информации о результате. Глубокую сеть можно рассматривать как многоэтапную операцию очистки информации (*дистилляции*), где информация проходит через последовательность фильтров и выходит из нее в *очищенном* виде (то есть в виде, пригодном для решения некоторых задач).

С технической точки зрения глубокое обучение – это многоступенчатый способ получения представления данных. Идея проста,

но, как оказывается, очень простые механизмы в определенном масштабе могут выглядеть непонятными и таинственными.

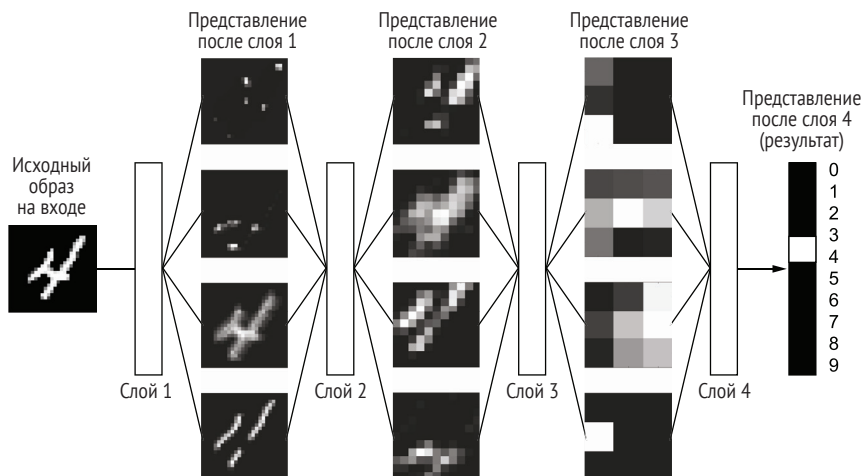


Рис. 1.6 Глубокие представления, получаемые моделью классификации цифр

1.1.5 Принцип действия глубокого обучения в трех рисунках

Теперь вы знаете, что суть машинного обучения заключается в преобразовании ввода (например, изображений) в результат (такой, как подпись «кошка»), которое получается путем исследования множества примеров входных данных и результатов. Вы также знаете, что глубокие нейронные сети превращают исходные данные в результат, выполняя длинную последовательность простых преобразований (слоев) и обучаются этим преобразованиям на примерах. Теперь посмотрим, как именно происходит обучение.

Что именно уровень делает со своими входными данными, определяется его весами, которые фактически являются набором чисел. Выражаясь техническим языком, можно сказать, что преобразование, реализуемое слоем, *параметризуется* его весами (рис. 1.7). (Весы также иногда называют *параметрами* слоя.) В данном контексте под *обучением* подразумевается поиск набора значений весов всех слоев в сети, при котором сеть будет правильно отображать образцовые входные данные в соответствующие им результаты. Но есть одна проблема: глубокая нейронная сеть может содержать десятки миллионов параметров. Поиск правильного значения для каждого из них может оказаться сложнейшей задачей, особенно если изменение значения одного параметра влияет на поведение всех остальных.

Чтобы чем-то управлять, сначала нужно получить возможность наблюдать за этим. Чтобы управлять результатом работы нейронной сети, нужно иметь возможность измерить, насколько этот результат далек от ожидаемого. Эту задачу решает *функция потерь* сети, ко-

торуую также называют *целевой функцией* или *функцией стоимости*. Функция потерь принимает предсказание, выданное сетью, и истинное значение (которое сеть должна была вернуть), и вычисляет оценку расстояния между ними, отражающую, насколько хорошо сеть справилась с данным конкретным примером (рис. 1.8).

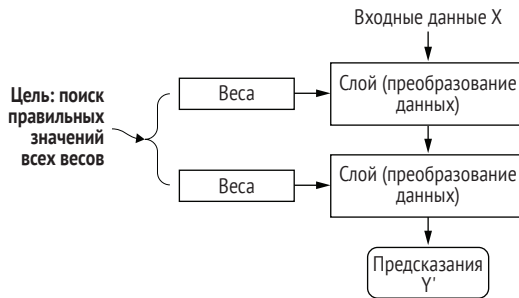


Рис. 1.7 Нейронная сеть параметризуется ее весами

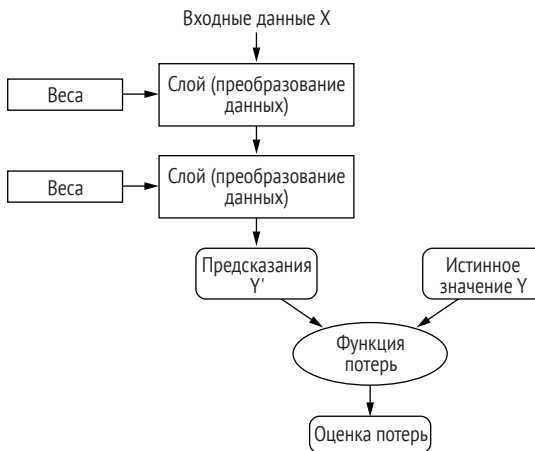


Рис. 1.8 Функция потерь оценивает качество результатов, производимых нейронной сетью

Основная хитрость глубокого обучения заключается в использовании этой оценки для корректировки значений весов с целью уменьшения потерь в текущем примере (рис. 1.9). Данная корректировка является задачей оптимизатора, который реализует так называемый алгоритм *обратного распространения ошибки* – центральный алгоритм глубокого обучения. Подробнее об алгоритме обратного распространения ошибки рассказывается в следующей главе.

Первоначально весам сети присваиваются случайные значения, то есть фактически сеть реализует последовательность случайных преобразований. Естественно, получаемый ею результат далек от идеала, и оценка потерь, соответственно, очень высока. Но с каждым

примером, обрабатываемым сетью, веса корректируются в нужном направлении, и оценка потерь уменьшается. Это цикл обучения, который повторяется достаточное количество раз (обычно десятки итераций с тысячами примеров) и порождает весовые значения, минимизирующие функцию потерь. Сеть с минимальными потерями и возвращающая результаты, близкие к истинным, называется обученной сетью. Повторюсь еще раз: это простой механизм, который при определенном масштабе начинает выглядеть непонятным и таинственным.

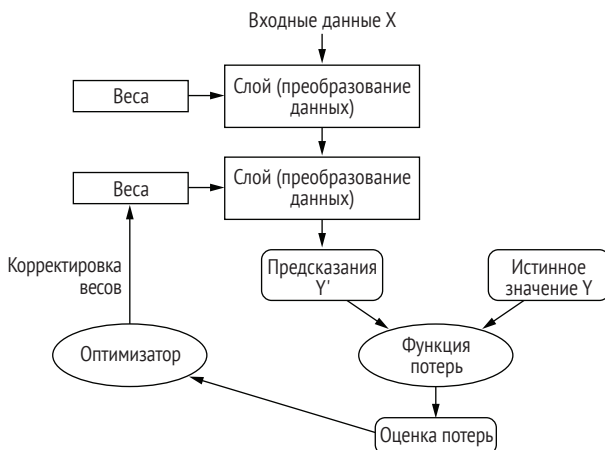


Рис. 1.9 Оценка потерь используется как обратная связь для корректировки весов

1.1.6 Каких успехов достигло глубокое обучение

Несмотря на то что глубокое обучение является давним разделом машинного обучения, фактическое его развитие началось только в начале 2010-х. За прошедшие несколько лет в этой области произошла настоящая революция, с особенно заметными успехами в моделировании восприятия, и даже обработки естественного языка – задач, кажущихся естественными и понятными для человека, но долгое время не дававшихся компьютерам. В частности, глубокое обучение достигло прорывов в следующих областях, традиционно сложных для машинного обучения:

- классификация изображений на уровне человека;
- распознавание речи на уровне человека;
- распознавание рукописного текста на уровне человека;
- улучшение качества машинного перевода с одного языка на другой;
- улучшение качества машинного чтения текста вслух;
- цифровые помощники, такие как Google Now и Amazon Alexa;
- управление автомобилем почти на уровне человека;

- повышение точности целевой рекламы, используемой компаниями Google, Baidu и Bing;
- повышение релевантности поиска в интернете;
- появление возможности отвечать на вопросы, заданные на разговорном языке;
- игра в Го лучше человека.

Мы все еще продолжаем исследовать возможности, которые таит в себе глубокое обучение. Мы начали применять его к широкому кругу проблем за пределами машинного восприятия и понимания естественного языка, таких как формальные рассуждения. Успех в этом направлении может означать начало новой эры, когда глубокое обучение будет помогать людям в науке, разработке программного обеспечения и многих других областях.

Мы все еще не оценили возможности глубокого обучения в полной мере. Мы начали с большим успехом применять его для решения широкого круга задач, которые еще несколько лет назад казались неразрешимыми, – автоматически расшифровывать десятки тысяч древних манускриптов, хранящихся в Апостольском архиве Ватикана, обнаруживать и классифицировать болезни растений в полевых условиях с помощью простого смартфона, помогать онкологам и рентгенологам интерпретировать данные медицинских изображений, прогнозировать стихийные бедствия, такие как наводнения, ураганы или даже землетрясения, и т. д. С каждой новой вехой мы приближаемся к эпохе, когда глубокое обучение будет применяться во всех областях человеческой деятельности – в науке, медицине, производстве, энергетике, транспорте, разработке программного обеспечения, сельском хозяйстве и даже художественном творчестве.

1.1.7 *Не верьте рекламной шумихе*

В сфере глубокого обучения за последние годы удалось добиться заметных успехов, однако ожидания на будущее десятилетие обычно намного превышают вероятные достижения. Даже при том, что многие значительные применения, такие как автопилоты для автомобилей, находятся практически на заключительной стадии реализации, многие другие, такие как полноценные диалоговые системы, перевод между произвольными языками на уровне человека и понимание естественного языка на уровне человека, скорее всего, еще долгое время будут оставаться недостижимыми. В частности, не стоит всерьез воспринимать разговоры об интеллекте на уровне человека. Завышенные ожидания от ближайшего будущего таят опасность: из-за невозможности реализации новых технологий инвестиции в исследования будут падать и прогресс замедлится на долгое время.

Такое уже происходило раньше. В прошлом ИИ пережил две волны подъема оптимизма, за которыми следовал спад, сопровождаемый разочарованиями и скептицизмом и, как результат, снижением

финансирования. Все началось с символического ИИ в 1960-х. В те ранние годы давались весьма многообещающие прогнозы развития ИИ. Один из самых известных пионеров и сторонников подхода символического ИИ Марвин Мински в 1967 году заявил: «В течение поколения... проблема создания “искусственного интеллекта” будет практически решена». Три года спустя, в 1970 году, он сделал более точное предсказание: «Через три – восемь лет у нас появится машина с интеллектом среднего человека». В 2023 году это достижение все еще кажется далеким будущим – пока мы не можем предсказать, сколько времени уйдет на это, – но в 1960-х и в начале 1970-х некоторые эксперты (как и многие люди ныне) полагали, что ИИ находится прямо за углом. Несколько лет спустя, из-за неоправдавшихся высоких ожиданий, исследователи и правительственные фонды отвернулись от этой области – так началась первая зима ИИ (вполне уместная метафора, потому что все это происходило вскоре после начала холодной войны).

Этот спад был не последним. В 1980-х начался подъем интереса к символическому ИИ благодаря буму *экспертных систем* в крупных компаниях. Первые успехи вызвали волну инвестиций, и корпорации по всему миру стали создавать свои отделы ИИ, занимающиеся разработкой экспертных систем. К 1985 году компании тратили более миллиарда долларов США в год на развитие технологии, но к началу 1990-х, из-за дороговизны в обслуживании, сложностей в масштабировании и ограниченности применения, интерес снова начал падать. Так началась вторая зима ИИ.

В настоящее время мы подходим к третьему циклу разочарования в ИИ, но пока мы еще находимся в фазе завышенного оптимизма. Сейчас лучше всего умерить наши ожидания на ближайшую перспективу и постараться донести до людей, мало знакомых с технической стороной этой области, что именно может дать глубокое обучение и на что оно не способно.

1.1.8 *Перспективы развития ИИ*

Даже при том, что наши ожидания на ближайшую перспективу могут быть нереалистичными, долгосрочная картина выглядит весьма ярко. Мы только начинаем применять глубокое обучение к решению многих важных задач: от медицинских диагнозов до цифровых помощников. В последние пять лет исследования в области ИИ продвигались удивительно быстро, во многом благодаря высокому уровню финансирования, никогда прежде не наблюдавшемуся в короткой истории искусственного интеллекта, но пока слишком малому, чтобы этот прогресс воплотить в продукты и процессы, формирующие наш мир. Большинство результатов исследований в глубоком обучении пока не нашли практического применения, по крайней мере применения к решению полного спектра задач, где эта технология могла бы найти применение. Ваш доктор и ваш бухгалтер пока не

используют ИИ. Вы сами в своей повседневной жизни тоже, вероятно, не используете технологии ИИ. Конечно, вы можете задавать простые вопросы своему смартфону и получать разумные ответы, вы можете получить весьма полезные рекомендации при выборе товаров на Amazon.com и по фразе «день рождения» быстро найти в Google Photos фотографии со дня рождения вашей дочери, который был в прошлом месяце. Это, несомненно, большой шаг вперед. Но такие инструменты лишь дополняют нашу жизнь. ИИ еще не занял центральное место в нашей жизни, работе и мыслях.

Сейчас трудно поверить, что ИИ может оказать значительное влияние на наш мир, потому что еще не развернулся в полной мере, – так в 1995 году трудно было верить в будущее влияние интернета. В то время большинство не понимало, какое отношение к ним может иметь интернет и как он изменит их жизнь. То же можно сегодня сказать о глубоком обучении и об искусственном интеллекте. Будьте уверены: эра искусственного интеллекта наступит. В недалеком будущем ИИ станет вашим помощником и даже другом; он ответит на ваши вопросы, поможет воспитывать детей и проследит за здоровьем. Он доставит продукты к вашей двери и отвезет вас из пункта А в пункт Б. Это будет ваш интерфейс к миру, который все более усложняется и наполняется информацией. И, что особенно важно, ИИ будет способствовать человечеству в движении вперед, помогая ученым делать новые прорывные открытия во всех областях науки, от геномики до математики.

По пути мы можем столкнуться с неудачами и, возможно, пережить новую зиму ИИ – так же как после всплеска развития интернет-индустрии в 1998–1999 годах произошел спад, вызванный уменьшением инвестиций в начале 2000-х. Но мы придем к цели – рано или поздно. В конечном итоге ИИ будет применяться во всех процессах в нашем обществе и в нашей жизни, так же как интернет сегодня.

Не верьте рекламной шумихе, но доверяйте долгосрочным прогнозам. Может потребоваться какое-то время, пока искусственный интеллект раскроет весь свой потенциал, глубину которого пока еще никто не может даже представить, но ИИ придет и изменит наш мир самым невероятным образом.

1.2 *Краткая история машинного обучения*

Глубокое обучение достигло уровня общественного внимания и инвестиций, невиданных прежде в истории искусственного интеллекта, но это не первая успешная форма машинного обучения. Можно с уверенностью сказать, что большинство алгоритмов машинного обучения, используемых сейчас в промышленности, не являются алгоритмами глубокого обучения. Глубокое обучение не всегда является правильным инструментом: иногда просто недостаточно данных

для глубокого обучения, иногда проблема лучше решается с применением других алгоритмов. Если глубокое обучение – ваш первый контакт с машинным обучением, вы можете оказаться в ситуации, когда, получив в руки молоток глубокого обучения, вы начнете воспринимать все задачи машинного обучения как гвозди. Единственный способ не попасть в эту ловушку – познакомиться с другими подходами и практиковать их, когда это необходимо.

В этой книге мы не будем подробно обсуждать классические подходы к машинному обучению, но коротко представим их и опишем исторический контекст, в котором они разрабатывались. Это поможет вам увидеть, какое место занимает глубокое обучение в более широком контексте машинного обучения, и лучше понять, откуда пришло глубокое обучение и почему оно имеет большое значение.

1.2.1 *Вероятностное моделирование*

Вероятностное моделирование – это применение принципов статистики к анализу данных. Это одна из самых ранних форм машинного обучения, которая до сих пор находит широкое использование. Одним из наиболее известных алгоритмов в этой категории является наивный байесовский алгоритм.

Наивный байесовский алгоритм – это вид классификатора машинного обучения, основанный на применении теоремы Байеса со строгими (или «наивными», откуда и происходит название алгоритма) предположениями о независимости входных данных. Эта форма анализа данных предшествовала появлению компьютеров и десятилетиями применялась вручную, пока не появилась ее первая реализация на компьютере (в 1950-х годах). Теорема Байеса и основы статистики были заложены в восемнадцатом столетии, и это все, что нужно для использования наивных байесовских классификаторов.

С этим алгоритмом тесно связана модель *логистической регрессии* (сокращенно *logreg*), которую иногда рассматривают как аналог примера «hello world» в современном машинном обучении. Пусть вас не вводит в заблуждение название. Модель логистической регрессии – это алгоритм классификации, а не регрессии. Так же как наивный байесовский алгоритм, модель логистической регрессии была разработана задолго до появления компьютеров, но до сих пор остается востребованной благодаря своей простоте и универсальной природе. Часто это первое, что пытается сделать исследователь со своим набором данных, чтобы получить представление о классификации.

1.2.2 *Первые нейронные сети*

Ранние версии нейронных сетей были полностью вытеснены современными вариантами, о которых рассказывается на страницах этой

книги, но вам будет полезно знать, как возникло глубокое обучение. Основные идеи нейронных сетей в упрощенном виде были исследованы еще в 1950-х. Долгое время развитие этого подхода тормозилось из-за отсутствия эффективного способа обучения больших нейронных сетей. Но ситуация изменилась в середине 1980-х, когда несколько исследователей, независимо друг от друга, вновь открыли алгоритм обратного распространения ошибки – способ обучения цепочек параметрических операций с использованием метода градиентного спуска (далее в книге мы дадим точные определения этим понятиям) – и начали применять его к нейронным сетям.

Первое успешное практическое применение нейронных сетей датируется 1989 годом, когда Ян Лекун в Bell Labs объединил ранние идеи сверточных нейронных сетей и обратного распространения ошибки и применил их для решения задачи распознавания рукописных цифр. Получившаяся в результате нейронная сеть была названа LeNet и использовалась почтовой службой США в 1990-х для автоматического распознавания почтовых индексов на конвертах.

1.2.3 Ядерные методы

По мере привлечения внимания исследователей к нейронным сетям в 1990-х и благодаря первому успеху приобрел известность новый подход к машинному обучению, быстро отправивший нейронные сети обратно в небытие: ядерные методы (kernel methods). *Ядерные методы* – это группа алгоритмов классификации, из которых наибольшую известность получил *метод опорных векторов* (Support Vector Machine, SVM). Современная формулировка SVM была предложена Владимиром Вапником и Коринной Кортес в начале 1990-х в Bell Labs и опубликована в 1995 году¹. Прежняя линейная формулировка была опубликована Вапником и Алексеем Червоненкисом (Alexey Chervonenkis) в начале 1963 года².

Метод опорных векторов предназначен для решения задач классификации путем поиска хороших решающих границ (рис. 1.10), разделяющих два набора точек, принадлежащих разным категориям. Поиск таких границ метод опорных векторов осуществляет в два этапа.

- 1 Данные отображаются в новое пространство более высокой размерности, где граница может быть представлена как гиперплоскость (если данные были двумерными, как на рис. 1.10, гиперплоскость вырождается в линию).
- 2 Хорошая решающая граница (разделяющая гиперплоскость)

¹ Vladimir Vapnik and Corinna Cortes, *Support-Vector Networks*, Machine Learning 20, no. 3 (1995): 273–297.

² Vladimir Vapnik and Alexey Chervonenkis, *A Note on One Class of Perceptrons*, Automation and Remote Control 25 (1964).

вычисляется путем максимизации расстояния от гиперплоскости до ближайших точек каждого класса, этот этап называют максимизацией зазора. Это позволяет обобщить классификацию новых образцов, не принадлежащих обучающему набору данных.

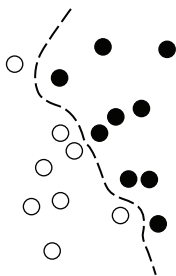


Рис. 1.10 Решающая граница

Методика отображения данных в пространство более высокой размерности, где задача классификации становится проще, может хорошо выглядеть на бумаге, но на практике часто оказывается трудноразрешимой. Вот тут и приходит на помощь изящная процедура *kernel trick* (ключевая идея, по которой ядерные методы получили свое название). Суть ее заключается в следующем: чтобы найти хорошие решающие гиперплоскости в новом пространстве, явно вычислять координаты точек в этом пространстве не требуется; достаточно вычислить расстояния между парами точек, что можно эффективно сделать с помощью функции ядра. Функция ядра – это незатратная вычислительная операция, отображающая любые две точки из исходного пространства и вычисляющая расстояние между ними в целевом пространстве представления, полностью минуя явное вычисление нового представления. Функции ядра обычно определяются вручную, а не извлекаются из данных – в случае с методом опорных векторов по данным определяется только разделяющая гиперплоскость.

На момент разработки метод опорных векторов демонстрировал лучшую производительность на простых задачах классификации и был одним из немногих методов машинного обучения, обладающих обширной теоретической базой и поддающихся серьезному математическому анализу, что сделало его хорошо понятным и легко интерпретируемым. Благодаря этим свойствам метод опорных векторов приобрел чрезвычайную популярность на долгие годы.

Однако метод опорных векторов оказался трудно применимым к большим наборам данных и не давал хороших результатов для таких задач, как классификация изображений. Так как SVM является поверхностным методом, для его применения к задачам распознавания требуется сначала вручную выделить представительную выборку (этот шаг называется *конструированием признаков*), что сопряжено со сложностями и чревато ошибками.

1.2.4 Деревья решений, случайные леса и градиентный бустинг

Деревья решений – это иерархические структуры, которые позволяют классифицировать входные данные или предсказывать выходные значения по заданным исходным значениям (рис. 1.11). Они легко визуализируются и интерпретируются. Деревья решений, формируемые на основе данных, заинтересовали исследователей в 2000-х, и к 2010 году им часто отдавали предпочтение перед ядерными методами.

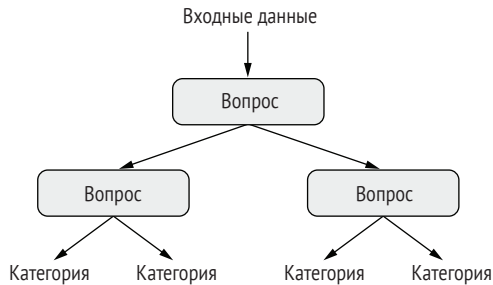


Рис. 1.11 Дерево решений: обучаемыми параметрами являются вопросы о данных. Таким вопросом мог бы быть, например, вопрос «Коэффициент 2 в данных больше 3,5?»

В частности, алгоритм *случайного леса* (random forest) предложил надежный и практичный подход к обучению на основе деревьев решений, включающий создание большого количества специализированных деревьев решений с последующим объединением выдаваемых ими результатов. Случайные леса применимы к широкому кругу задач – можно сказать, что они почти всегда являются оптимальным алгоритмом для любых задач поверхностного машинного обучения. Когда в 2010 году был запущен популярный конкурсный веб-сайт Kaggle (<http://kaggle.com>), посвященный машинному обучению, случайные леса быстро превратились в наиболее популярную платформу, пока в 2014 году не появился метод градиентного бустинга. Метод градиентного бустинга, во многом напоминающий случайный лес, – это прием машинного обучения, основанный на объединении слабых моделей прогнозирования, обычно – деревьев решений. Он использует градиентный бустинг, способ улучшения любой модели машинного обучения путем итеративного обучения новых моделей, специализированных для устранения слабых мест в предыдущих моделях. Применительно к деревьям решений прием градиентного бустинга позволяет получить модели, которые в большинстве случаев превосходят случайные леса – при сохранении аналогичных свойств. На сегодняшний день это один из лучших алгоритмов, хотя и не самый лучший, для решения задач, не связанных с распознава-

нием. Наряду с глубоким обучением это один из наиболее широко используемых приемов в конкурсах на сайте Kaggle.

1.2.5 Назад к нейронным сетям

Примерно в 2010 году, несмотря на почти полную потерю интереса к нейронным сетям со стороны научного сообщества, ряд исследователей, продолжавших работать с нейронными сетями, стали добиваться важных успехов: группы Джеффри Хинтона из университета в Торонто, Йошуа Бенжю из университета в Монреале, Яна Лекуна из университета в Нью-Йорке и исследователи в научно-исследовательском институте искусственного интеллекта IDSIA в Швейцарии.

В 2011 году Ден Киресан из IDSIA выиграл академический конкурс по классификации изображений с применением глубоких нейронных сетей, обучаемых на GPU, – это был первый практический успех современного глубокого обучения. Но перелом произошел в 2012 году, когда группа Хинтона приняла участие в ежегодном соревновании ImageNet по крупномасштабному распознаванию образов. ImageNet предложила очень сложное на то время задание, заключающееся в классификации цветных изображений с высоким разрешением на 1000 разных категорий после обучения по выборке, включающей 1,4 млн изображений. В 2011 году модель-победитель, основанная на классических подходах к распознаванию образов, показала точность лишь 74,3 %¹. В 2012 году команда Алекса Крижевски (Alex Krizhevsky), в которой советником был Джеффри Хинтон (Geoffrey Hinton), достигла точности в 83,6 % – значительный прорыв. С тех пор каждый год первые позиции в этом соревновании занимают глубокие сверточные нейронные сети. В 2015 году победитель достиг точности в 96,4 %, и задача классификации на ImageNet была сочтена решенной полностью.

Начиная с 2012 года глубокие сверточные нейронные сети (convolutional neural networks, сокращенно *convnet*) перешли в разряд передовых алгоритмов для всех задач распознавания образов; в более общем плане они с успехом могут использоваться в любых задачах распознавания. На крупных конференциях по распознаванию образов, проводившихся в 2015 и 2016 годах, было трудно найти презентацию, не включающую сверточные нейросети в том или ином виде. В то же время глубокое обучение нашло применение во многих других видах задач, таких как обработка естественного языка. Оно полностью заменило метод опорных векторов и деревья решений в широком круге задач. Например, в течение нескольких лет Европейская организация по ядерным исследованиям (European Organization for Nuclear Research, CERN) использовала методы на основе

¹ Показатель «точность первой пятерки» измеряет, как часто модель выбирает правильный ответ как одну из своих пяти догадок (из 1000 возможных ответов, в случае ImageNet).

деревьев решений для данных, получаемых с детектора частиц ATLAS в большом адронном коллайдере, но затем в CERN было принято решение перейти на использование глубоких нейронных сетей на основе Keras из-за их более высокой производительности и простоты обучения на больших наборах данных.

1.2.6 Отличительные черты глубокого обучения

Основная причина быстрого взлета глубокого обучения заключается в лучшей производительности во многих задачах. Однако это не единственная причина. Глубокое обучение также существенно упрощает решение проблем, полностью автоматизируя важнейший шаг в машинном обучении, выполнявшийся раньше вручную, – конструирование признаков.

Предшествовавшие методы машинного обучения – методы поверхностного обучения – включали преобразование входных данных только в одно или два последовательных пространства, обычно посредством простых преобразований, таких как нелинейная проекция в пространство более высокой размерности (метод опорных векторов) или деревья решений. Однако точные представления, необходимые для решения сложных задач, обычно нельзя получить такими способами. Поэтому людям приходилось прилагать большие усилия, чтобы привести исходные данные к виду, более пригодному для обработки этими методами: им приходилось вручную улучшать слой представления своих данных, то есть заниматься конструированием признаков. Глубокое обучение, напротив, полностью автоматизирует этот шаг: с применением методов глубокого обучения все признаки извлекаются за один проход, без необходимости конструировать их вручную. Это очень упростило процесс машинного обучения, потому что часто сложный и многоступенчатый конвейер оказалось возможным заменить единственной простой сквозной моделью глубокого обучения.

Вы можете спросить: если суть вопроса заключается в получении нескольких последовательных слоев представлений, можно ли многократно применить методы поверхностного обучения для имитации эффекта глубокого обучения? На практике наблюдается быстрое уменьшение последовательных применений методов поверхностного обучения, поскольку оптимальный слой первого представления в трехслойной модели не является оптимальным первым слоем в однослойной или двухслойной модели. Особенность преобразования в глубоком обучении состоит в том, что модель может обучать все слои представления *вместе* и одновременно, а не последовательно (этот тип обучения также называют «жадным»). При совместном изучении признаков, когда модель корректирует один

из своих внутренних признаков, все прочие признаки, зависящие от него, автоматически корректируются в соответствии с изменениями, без вмешательства человека. Все контролируется единственным сигналом обратной связи: каждое изменение в модели служит конечной цели. Это намного более мощный подход, чем жадно накладывать поверхностные модели друг на друга, потому что позволяет изучать более сложные абстрактные представления, разбивая их на длинные ряды промежуточных пространств (слоев), в которых каждое последующее пространство получается в результате простого преобразования предыдущего.

Методика глубокого обучения обладает двумя важными характеристиками: она *поэтапно, послойно конструирует все более сложные представления и совместно исследует промежуточные представления*, благодаря чему каждый слой обновляется в соответствии с потребностями представления слоя выше и потребностями слоя ниже. Вместе эти два свойства делают глубокое обучение намного успешнее предыдущих подходов к машинному обучению.

1.2.7 Современный ландшафт машинного обучения

Отличный способ получить представление о текущей ситуации с алгоритмами и инструментами машинного обучения – посмотреть соревнования по машинному обучению на Kaggle. Благодаря острой конкуренции (в некоторых конкурсах участвуют тысячи участников, а призовой фонд исчисляется миллионами долларов) и большому количеству охватываемых задач машинного обучения на сайте Kaggle можно реально оценить, какие подходы используются и насколько успешно. Так какой же алгоритм уверенно выигрывает состязания? Какими инструментами пользуются победители?

В начале 2019 года на сайте Kaggle провели опрос, в ходе которого команды, вошедшие в пятерку лучших в соревнованиях с 2017 года, спросили, какой основной программный инструмент они использовали для выполнения конкурсных заданий (рис. 1.12). Оказывается, лучшие команды склонны использовать либо методы глубокого обучения (чаще всего с помощью библиотеки Keras), либо градиентный бустинг над решающими деревьями (библиотеки LightGBM или XGBoost).

Речь не только о победителях соревнований. Kaggle также ежегодно проводит опрос среди специалистов в области машинного обучения и науки о данных по всему миру. Этот опрос, в котором приняли участие десятки тысяч респондентов, является одним из самых надежных источников информации о состоянии отрасли. На рис. 1.13 показан процент использования различных программных платформ машинного обучения.

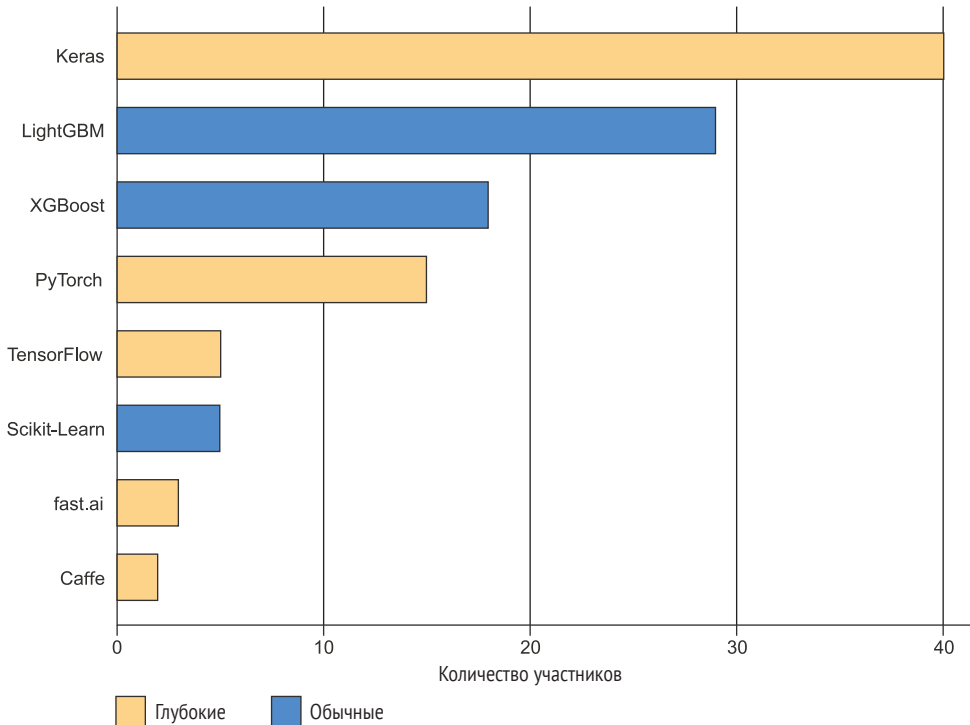


Рис. 1.12 Наиболее популярные инструменты машинного обучения, применяемые в конкурсах Kaggle

С 2016 по 2020 год во всей отрасли машинного обучения и науки о данных доминировали два подхода: глубокое обучение и градиентный бустинг над решающими деревьями. В частности, второй подход используется для задач, где доступны структурированные данные, тогда как глубокое обучение используется для задач восприятия, таких как классификация изображений.

Пользователи градиентного бустинга над решающими деревьями обычно используют Scikit-Learn, XGBoost или LightGBM. В свою очередь, большинство практиков глубокого обучения используют Keras, часто в сочетании с родительским фреймворком TensorFlow. Общим для этих инструментов является то, что все они доступны в виде библиотек для языков R или Python, которые на сегодняшний день наиболее широко применяются в машинном обучении и обработке данных.

Этим двум методам вы должны уделять особое внимание, чтобы добиться успеха в применении машинного обучения в наше время: метод градиентного бустинга для задач поверхностного обучения и глубокое обучение для задач распознавания. В техническом плане это означает, что вы должны владеть двумя библиотеками – XGBoost и Keras, занимающими доминирующее положение в конкурсах на сайте Kaggle. Как только вы взяли в руки эту книгу, вы уже сделали большой шаг к этой цели.

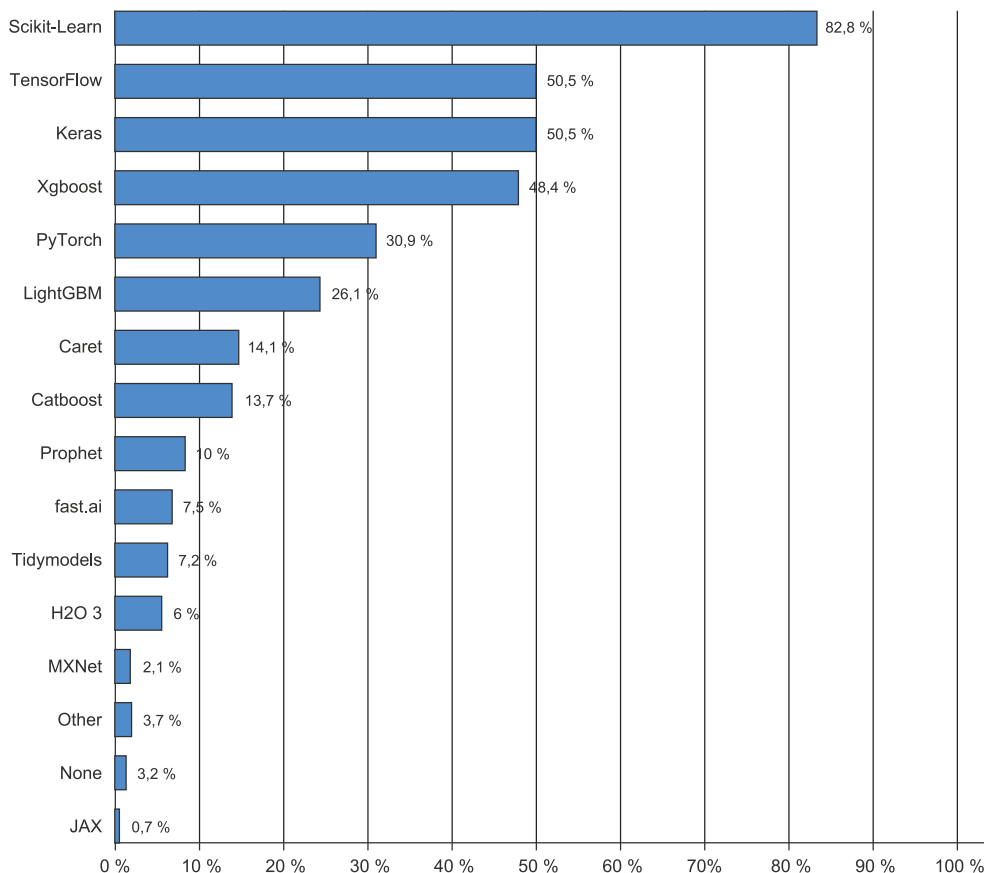


Рис. 1.13 Использование инструментов в машинном обучении и науке о данных (источник: <https://www.kaggle.com/kaggle-survey-2020>)

1.3 Почему глубокое обучение? Почему сейчас?

Две ключевые идеи глубокого обучения для решения задач распознавания образов – сверточные нейронные сети и алгоритм обратного распространения ошибки – были хорошо известны уже в 1990 году. Алгоритм *долгой краткосрочной памяти* (Long Short-Term Memory, LSTM), составляющий основу глубокого обучения для прогнозирования временных рядов, был предложен в 1997 году и с тех пор почти не изменился. Так почему же глубокое обучение начало применяться только с 2012 года? Что изменилось за эти два десятилетия? В целом машинное обучение стоит на «трех китах»:

- оборудование;
- наборы данных и тесты;
- современные алгоритмы.

Поскольку в области машинного обучения главную роль играют экспериментальные выводы, а не теория, алгоритмические достижения возможны только при наличии данных и оборудования, пригодных для проверки идей (или, как это часто бывает, для возрождения старых идей). Машинное обучение – это не математика и не физика, где прорывы могут быть сделаны с помощью ручки и бумаги. Это инженерная наука.

Настоящим узким местом на протяжении 1990-х и 2000-х годов были данные и оборудование. Но в течение этого времени происходило бурное развитие интернета, а для рынка компьютерных игр были созданы высокопроизводительные графические процессоры.

1.3.1 Оборудование

Между 1990 и 2010 годами быстродействие стандартных процессоров выросло примерно в 5000 раз. В результате сейчас на ноутбуке можно запускать небольшие модели глубокого обучения, тогда как 25 лет назад это в принципе было невозможно.

Но типичные модели глубокого обучения, используемые для распознавания образов или речи, требуют вычислительной мощности на порядки больше, чем мощность ноутбука. В течение 2000-х такие компании, как NVIDIA и AMD, вложили миллионы долларов в разработку быстрых процессоров с массовым параллелизмом (графических процессоров – Graphical Processing Unit, GPU) для поддержки графики все более реалистичных видеоигр – недорогих, специализированных суперкомпьютеров, предназначенных для отображения на экране сложных трехмерных сцен в режиме реального времени. Эти инвестиции принесли пользу научному сообществу, когда в 2007 году компания NVIDIA выпустила CUDA (<https://developer.nvidia.com/about-cuda>), программный интерфейс для линейки своих GPU. Несколько GPU теперь могут заменить мощные кластеры на обычных процессорах в различных задачах, допускающих возможность массового распараллеливания вычислений, начиная с физического моделирования. Глубокие нейронные сети, выполняющие в основном умножение множества маленьких матриц, также допускают высокую степень распараллеливания; и ближе к 2011 году некоторые исследователи начали писать CUDA-реализации нейронных сетей. Одними из первых стали Дэн Кайесан¹ и Алекс Крижевски².

Получилось так, что игровая индустрия субсидировала создание суперкомпьютеров для следующего поколения приложений искусственного интеллекта. Иногда крупные достижения начинают

¹ *Flexible, High Performance Convolutional Neural Networks for Image Classification*, Proceedings of the 22nd International Joint Conference on Artificial Intelligence (2011), <http://mng.bz/nNOK>.

² *ImageNet Classification with Deep Convolutional Neural Networks*, Advances in Neural Information Processing Systems 25 (2012), <http://mng.bz/2286>.

ся с игр. Современный графический процессор NVIDIA TITAN RTX, стоивший 2500 долларов США в конце 2019 года, способен выдать пиковую производительность 16 терафлопс с одинарной точностью (16 трлн операций в секунду с числами типа float32). Это почти в 500 раз больше производительности лучшего суперкомпьютера 1990-х Intel Touchstone Delta. Графическому процессору TITAN RTX требуется всего несколько часов для обучения модели ImageNet типа той, что выиграла конкурс ILSVRC в 2012 или 2013 году. Крупные компании обучают модели глубокого обучения на кластерах, состоящих из сотен GPU.

Более того, индустрия глубокого обучения начинает выходить за рамки GPU и инвестировать средства в развитие еще более специализированных процессоров, наиболее эффективно показывающих себя в области глубокого обучения. В 2016 году на своей ежегодной конференции Google I/O компания Google продемонстрировала свой проект тензорного процессора (Tensor Processing Unit, TPU): разработанная с нуля архитектура позволяет обучать глубокие нейронные сети значительно быстрее и гораздо эффективнее с точки зрения энергопотребления, чем лучшие графические процессоры. В 2020 году третья версия карты TPU имела вычислительную мощность 420 терафлопс. Это в 10 000 раз больше, чем у Intel Touchstone Delta в 1990 году.

Карты TPU предназначены для сборки в крупномасштабные модули, называемые «стручками» (pod). Один модуль (1024 карты TPU) достигает пиковой производительности 100 петафлопс. Для сравнения, это около 10 % пиковой вычислительной мощности самого крупного на сегодняшний день суперкомпьютера IBM Summit в национальной лаборатории Ок-Ридж, который состоит из 27 000 графических процессоров NVIDIA и достигает пиковой производительности около 1,1 эксафлопса.

1.3.2 Данные

Иногда искусственный интеллект называют новой промышленной революцией. Если глубокое обучение – паровой двигатель этой революции, то данные – это уголь: сырье, питающее наши интеллектуальные машины, без которого невозможно движение вперед. К вопросу о данных: вдобавок к экспоненциальному росту емкости устройств хранения информации, наблюдавшемуся в последние 20 лет (согласно закону Мура), перемены в игровом мире вызвали бурный рост интернета, благодаря чему появилась возможность накапливать и распространять очень большие объемы данных для машинного обучения. В настоящее время крупные компании работают с коллекциями изображений, видео и текстовых материалов, которые невозможно было бы собрать без интернета. Например, изображения на сайте Flickr, классифицированные пользователями, стали золотой жилой для разработчиков моделей распознавания образов.

То же можно сказать о видеороликах на YouTube. А Википедия стала ключевым источником наборов данных для задач обработки естественного языка.

Если и есть набор данных, ставший катализатором для развития глубокого обучения, то это коллекция ImageNet, включающая 1,4 млн изображений, классифицированных вручную на 1000 категорий (каждое изображение отнесено только к одной категории). Но особенной коллекцию ImageNet делает не столько ее огромный размер, сколько ежегодные соревнования¹, в которых она задействована.

Как показывает пример Kaggle, публичные конкурсы начиная с 2010 года – отличный способ мотивации исследователей и инженеров преодолевать все новые и новые рубежи. Наличие общих критериев оценки достижений исследователей значительно помогло недавнему росту глубокого обучения.

1.3.3 Алгоритмы

Кроме оборудования и данных, до конца 2000-х нам не хватало надежного способа обучения очень глубоких нейронных сетей. Как результат, нейронные сети оставались очень неглубокими, имеющими один или два слоя представления; в связи с этим они не могли противостоять более совершенным поверхностным методам, таким как метод опорных векторов и случайные леса. Основная проблема заключалась в *распространении градиента* через глубокие пакеты слоев. Сигнал обратной связи, используемый для обучения нейронных сетей, затухает по мере увеличения количества слоев.

Ситуация изменилась в 2009–2010 годах с появлением некоторых простых, но важных алгоритмических усовершенствований, позволивших улучшить распространение градиента:

- улучшенные *функции активации*;
- улучшенные *схемы инициализации весов*, начиная с предварительного послойного обучения, от которого быстро отказались;
- улучшенные *схемы оптимизации*, такие как RMSProp и Adam.

Только когда эти улучшения позволили создавать модели с 10 слоями и более, началось развитие глубокого обучения. Наконец в 2014, 2015 и 2016 годах были открыты еще более совершенные способы распространения градиента, такие как пакетная нормализация, остаточные связи и отдельные свертки по глубине.

Сегодня мы можем обучать с нуля модели произвольной глубины. Это открыло возможность использования чрезвычайно больших моделей, обладающих значительной *репрезентативностью*, то есть кодирующих очень богатые пространства гипотез. Такая невероятная масштабируемость является одной из определяющих характеристик

¹ ImageNet Large Scale Visual Recognition Challenge (ILSVRC), <http://www.image-net.org/challenges/LSVRC>.

современного глубокого обучения. Архитектуры крупномасштабных моделей с десятками уровней и десятками миллионов параметров легли в основу важных достижений как в компьютерном зрении (например, такие архитектуры, как ResNet, Inception или Xception), так и в обработке естественного языка (например, архитектуры на основе трансформеров, такие как BERT, GPT-3 или XLNet).

1.3.4 Новая волна инвестиций

Как мы уже говорили, в 2012–2013 годах глубокое обучение вывело на новый уровень распознавание образов и, в конечном счете, все задачи распознавания. За этим последовала постепенно нарастающая волна инвестиций в индустрию, намного превосходящая все предыдущие, наблюдавшиеся в истории ИИ (рис. 1.14).

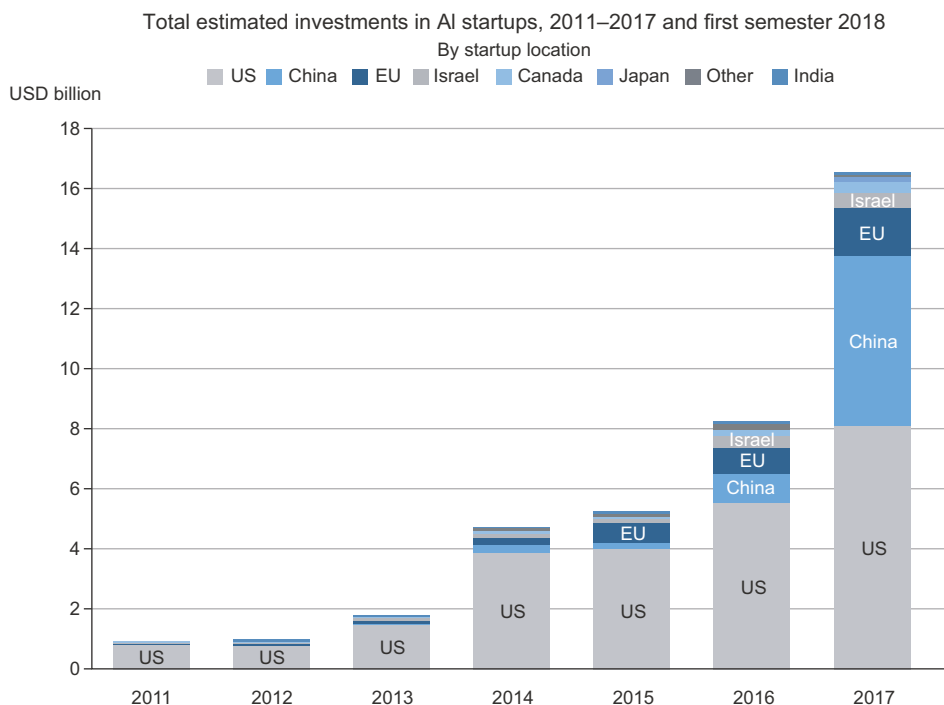


Рис. 1.14 Объем общих инвестиций в стартапы ИИ по данным ОЭСР (источник: <http://mng.bz/zGN6>)

В 2011 году, как раз перед тем, как в центре внимания оказалось глубокое обучение, общие венчурные инвестиции в ИИ по всему миру составили менее миллиарда долларов, которые почти полностью пошли на практическое применение поверхностных подходов машинного обучения. В 2015 году инвестиции выросли до более чем 5 млрд долларов, а в 2017 году – до ошеломляющих 16 млрд

долларов. За эти несколько лет появились сотни стартапов, пытающихся извлечь выгоду из поднявшейся шумихи. Между тем крупные компании, такие как Google, Facebook, Baidu и Microsoft, инвестировали деньги в исследования, проводившиеся внутренними подразделениями, и объемы этих инвестиций почти наверняка превысили инвестиции венчурного капитала.

Машинное обучение, и глубокое обучение в частности, заняло центральное место среди стратегических продуктов этих технологических гигантов. В конце 2015 году генеральный директор Google Сундар Пичаи отметил: «Машинное обучение – это повод для решительной смены системы координат во всех видах нашей деятельности. Мы вдумчиво применяем его во всех наших продуктах, будь то поиск, реклама, YouTube или Play. И мы с самого начала – и систематически – применяем машинное обучение во всех этих областях»¹.

В результате этой волны инвестиций за прошедшие 10 лет число людей, работающих над глубоким обучением, увеличилось с нескольких сотен до десятков тысяч, а прогресс в исследованиях достиг небывалого уровня.

1.3.5 *Демократизация глубокого обучения*

Одним из ключевых факторов, обусловивших приток новых лиц в глубокое обучение, стала демократизация инструментов, используемых в этой области. На начальном этапе глубокое обучение требовало значительных знаний и опыта программирования на C++ и владения CUDA, чем могли похвастать очень немногие.

В настоящее время базовых навыков написания сценариев R или Python достаточно для проведения масштабных исследований в области глубокого обучения. В первую очередь это стало возможным благодаря появлению библиотеки TensorFlow – фреймворка для работы с тензорами, который поддерживает автодифференциацию, что значительно упрощает реализацию новых моделей, – а также удобных для пользователя библиотек, таких как Keras, которые делают глубокое обучение таким же простым, как использование кубиков LEGO. После выпуска в начале 2015 года Keras быстро стал популярным решением для глубокого обучения среди большого количества новых стартапов, аспирантов и исследователей, работающих в этой области.

1.3.6 *Ждать ли продолжения этой тенденции?*

Есть ли что-то особенное в глубоком обучении, что делает его «правильным» выбором и для компаний, делающих инвестиции, и для исследователей? Или глубокое обучение – это просто увлечение, которое не продлится долго? Будем ли мы использовать глубокие нейронные сети через 20 лет?

¹ Sundar Pichai, Alphabet earnings call, Oct. 22, 2015.

Глубокое обучение имеет несколько свойств, которые оправдывают его статус как революции в ИИ, и оно задержится надолго. Возможно, мы перестанем использовать нейронные сети через два десятилетия, но все, что останется взамен, будет прямым наследником современного глубокого обучения и его основных идей. Важнейшие свойства глубокого обучения можно разделить на три категории:

- *простота* – глубокое обучение избавляет от необходимости конструировать признаки, заменяя сложные, противоречивые и тяжелые конвейеры простыми обучаемыми моделями, которые обычно строятся с использованием пяти-шести тензорных операций;
- *масштабируемость* – глубокое обучение легко поддается распараллеливанию на GPU или TPU, поэтому оно в полной мере может использовать закон Мура. Кроме того, обучение моделей можно производить итеративно, на небольших пакетах данных, что дает возможность проводить обучение на наборах данных произвольного размера. (Единственным узким местом является объем доступной вычислительной мощности для параллельных вычислений, которая, как следует из закона Мура, является быстро приближающимся барьером);
- *гибкость и пригодность к многократному использованию* – в отличие от многих предыдущих подходов, модели глубокого обучения могут обучаться на дополнительных данных без полного перезапуска, что делает их пригодными для непрерывного и продолжительного обучения – очень важное свойство для очень больших промышленных моделей. Кроме того, обучаемые модели глубокого обучения можно перенацеливать и, соответственно, использовать многократно: например, модель, обученную классификации изображений, можно включить в конвейер обработки видео. Это позволяет использовать предыдущие наработки для создания все более сложных и мощных моделей. Это также позволяет применить глубокое обучение к очень маленьким объемам данных.

Глубокое обучение находится в центре внимания всего несколько лет, и мы еще не определили границы его возможностей. Каждый год мы узнаем о новых и новых вариантах использования и инженерных усовершенствованиях, которые снимают предыдущие ограничения. После научной революции прогресс обычно развивается по сигмоиде: сначала наблюдается быстрый рост, который постепенно стабилизируется, когда исследователи сталкиваются с труднопреодолимыми ограничениями, и затем дальнейшие усовершенствования замедляются.

В первом издании этой книги в 2016 году я предсказывал, что глубокое обучение все еще находится в первой половине этой сигмоиды, а в следующие несколько лет нас ждет гораздо более революционный прогресс. Это подтвердилось на практике: в 2017 и 2018 годах

появились модели глубокого обучения на основе Transformer для обработки естественного языка, которые стали революцией в этой области, и в то же время глубокое обучение продолжало демонстрировать новые успехи в компьютерном зрении и распознавании речи. Сегодня, в 2022 году, глубокое обучение, похоже, вошло во вторую половину этой сигмоиды. Мы все еще можем ожидать значительного прогресса в ближайшие годы, но мы, вероятно, вышли из начальной фазы взрывного роста.

Сегодня меня очень вдохновляют примеры использования технологии глубокого обучения для решения бесчисленного перечня задач. Глубокое обучение все еще находится в стадии становления, и потребуется много лет, чтобы полностью раскрыть его потенциал.