

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 dmkpress@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.