Содержание

Предисловие ... 14

Глава 1. Обзор машинного обучения ... 20
Обучающиеся машины ... 20
Как машины могут обучаться? ... 21
Биологические корни ... 23
Что такое глубокое обучение? ... 24
Вниз по кроличьей норе ... 25
Формулировка вопросов ... 26
Математические основания машинного обучения: линейная алгебра ... 26
Скаляры ... 26
Векторы ... 26
Матрицы ... 27
Тензоры ... 27
Гиперплоскости ... 28
Математические операции ... 28
Преобразование данных в векторы ... 28
Решение систем уравнений ... 30
Математические основания машинного обучения: статистика ... 32
Вероятность ... 32
Условные вероятности ... 34
Апостериорная вероятность ... 34
Распределения вероятности ... 35
Выборка и генеральная совокупность ... 37
Методы с перевыборкой ... 37
Смещение выборки ... 37
Правдоподобие ... 38
Как работает машинное обучение? ... 38
Регрессия ... 38
Классификация ... 40
Кластеризация ... 40
Недообучение и переобучение ... 41
Оптимизация ... 41
Выпуклая оптимизация ... 42
Градиентный спуск ... 43
Стохастический градиентный спуск ... 45
Квазиньютоновские методы оптимизации ... 46
Порождающие и дискриминантные модели ... 46
Логистическая регрессия ... 47
Логистическая функция ... 48
Интерпретация результата логистической регрессии ... 48
Оценивание модели ... 49
Матрица неточностей ... 49
Итоги ... 52

Глава 2. Основы нейронных сетей и глубокого обучения ... 53
Нейронные сети ... 53
Биологический нейрон ... 55
Перцептрон ... 57
Многослойные сети прямого распространения ... 60
Обучение нейронных сетей ... 64
Обучение методом обратного распространения ... 65
Функции активации ... 71
Линейная функция ... 71
Сигмоида ... 72
Функция tanh ... 73
Функция hardtanh ... 73
Функция softmax ... 73
Линейная ректификация ... 74
Функции потерь ... 76
Применяемые обозначения ... 76
Функции потерь для регрессии ... 77
Функции потерь для классификации ... 78
Функции потерь для реконструкции ... 80
Гиперпараметры ... 81
Скорость обучения ... 81
Регуляризация ... 82
Импульс ... 82
Разреженность ... 82

Глава 3. Основания глубоких сетей ... 83
Определение глубокого обучения ... 83
Что такое глубокое обучение? ... 83
Как организована эта глава ... 92
Общие архитектурные принципы глубоких сетей ... 92
Параметры ... 93
Слои ... 93
Функции активации ... 94
Функции потерь ... 95
Алгоритмы оптимизации ... 96
Гиперпараметры ... 98
Итоги ... 102
Строительные блоки глубоких сетей ... 102
Ограниченные машины Больцмана ... 103
Автокодировщики ... 108
Вариационные автокодировщики ... 109

Глава 4. Основные архитектуры глубоких сетей ... 111
Сети, предобученные без учителя ... 111
Глубокие сети доверия ... 111
Порождающие состязательные сети ... 114
Сверточные нейронные сети (СНС) ... 117
Биологические корни ... 118
Интуитивное описание ... 119
Общий взгляд на архитектуру СНС ... 120
Входной слой ... 121

Пулинговые слои ... 128
Полносвязные слои ... 129
Другие применения СНС ... 129
Самые известные СНС ... 130
Итоги ... 130
Рекуррентные нейронные сети ... 131
Моделирование времени ... 131
Трехмерный вход ... 133
Почему не марковские модели? ... 135
Общая архитектура рекуррентной нейронной сети ... 136
LSTM-сети ... 136
Предметно-ориентированные приложения и гибридные сети ... 143
Рекурсивные нейронные сети ... 144
Архитектура сети ... 144
Разновидности рекурсивных нейронных сетей ... 145
Применение рекурсивных нейронных сетей ... 145
Итоги и обсуждение ... 145
Приведет ли глубокое обучение к отмиранию всех прочих алгоритмов? ... 146
Оптимальный метод зависит от задачи ... 146
Когда мне может понадобиться глубокое обучение? ... 147

Глава 5. Построение глубоких сетей ... 148
Выбор глубокой сети для решения задачи ... 148
Табличные данные и многослойные перцептроны ... 148
Изображения и сверточные нейронные сети ... 149
Последовательности, временные ряды и рекуррентные нейронные сети ... 150
Применение гибридных сетей ... 151
Инструментарий DL4J ... 151
Векторизация и DataVec ... 152
Среды выполнения и ND4J ... 152
Основные концепции DL4J API ... 153
Загрузка и сохранение моделей ... 153
Получение входных данных для модели ... 154
Задание архитектуры модели ... 154
Обучение и оценивание ... 155
Моделирование CSV-данных с помощью многослойных перцептронов ... 156
Подготовка входных данных ... 158
Задание архитектуры сети ... 159
Обучение модели ... 161
Оценивание модели ... 161
Моделирование рукописных цифр с помощью СНС ... 162
Реализация СНС LeNet на Java ... 163
Загрузка и векторизация входных изображений ... 165
Архитектура сети LeNet в DL4J ... 165
Обучение СНС ... 168
Моделирование последовательных данных с помощью рекуррентной нейронной сети ... 169
Порождение текста в стиле Шекспира с помощью LSTM-сети ... 169
Классификация временных рядов, содержащих показания датчика, с помощью LSTM-сети ... 177
Применение автокодировщиков для обнаружения аномалий ... 183

Java-программа автокодировщика ... 183
Подготовка входных данных ... 187
Архитектура и обучение сети автокодировщика ... 187
Оценивание модели ... 188
Использование вариационных автокодировщиков для реконструкции цифр из набора MNIST ... 189
Программа реконструкции цифр для набора MNIST ... 190
Изучение модели VAE ... 192
Применение глубокого обучения в обработке естественного языка ... 195
Обучение погружениям слов с помощью Word2Vec ... 196
Распределенные представления предложений с помощью векторов абзацев ... 201
Применение векторов абзацев для классификации документов ... 204

Глава 6. Настройка глубоких сетей ... 209
Основные концепции настройки глубоких сетей ... 209
Интуитивное описание построения глубоких сетей ... 209
Преобразование интуитивных представлений в пошаговую процедуру ... 210
Подбор сетевой архитектуры, соответствующей входным данным ... 211
Итоги ... 212
Соотнесение назначения модели с выходным слоем ... 213
Выходной слой регрессионной модели ... 213
Выходной слой модели классификации ... 213
Количество слоев, количество параметров и объем памяти ... 216
Многослойные нейронные сети прямого распространения ... 216
Управление количеством слоев и параметров ... 217
Оценка требований к объему памяти ... 219
Стратегии инициализации весов ... 220
Ортогональная инициализация весов в РНС ... 221
Выбор функции активации ... 221
Сводная таблица функций активации ... 223
Применение функций потерь ... 223
Скорость обучения ... 225
Использование отношения обновлений к параметрам ... 226
Конкретные рекомендации по заданию скорости обучения ... 227
Как разреженность влияет на обучение ... 228
Применение методов оптимизации ... 229
Рекомендации по применению СГС ... 230
Применение распараллеливания и GPU для ускорения обучения ... 231
Онлайновое обучение и параллельные итеративные алгоритмы ... 232
Распараллеливание СГС в DL4J ... 234
Графические процессоры ... 236
Управление периодами и размером мини-пакета ... 237
Компромиссы при определении размера мини-пакета ... 238
О применении регуляризации ... 239
Априорная функция как регуляризатор ... 239
Регуляризация по максимальной норме ... 240
Прореживание ... 240
Другие вопросы, связанные с регуляризацией ... 242
Дисбаланс классов ... 242
Методы выборки из классов ... 244
Взвешенные функции потерь ... 244

Борьба с переобучением ... 245
Использование статистики сети из интерфейса настройки ... 246
Обнаружение неудачной инициализации весов ... 248
Обнаружение неперемешанных данных ... 249
Обнаружение проблем, относящихся к регуляризации ... 251

Глава 7. Настройка глубоких сетей с конкретной архитектурой ... 253
Сверточные нейронные сети (СНС) ... 253
Общие архитектурные паттерны сверточных сетей ... 254
Конфигурирование сверточных слоев ... 257
Конфигурирование пулинговых слоев ... 261
Перенос обучения ... 262
Рекуррентные нейронные сети ... 263
Входные данные и входной слой сети ... 264
Выходные слои и RnnOutputLayer ... 264
Обучение сети ... 265
Отладка типичных проблем в LSTM ... 267
Дополнение и маскирование ... 267
Применение маскирования для оценивания и скоринга ... 268
Варианты архитектуры рекуррентных сетей ... 269
Ограниченные машины Больцмана ... 269
Скрытые блоки и моделирование доступной информации ... 270
Типы блоков ... 271
Регуляризация в ОМБ ... 271
Глубокие сети доверия ... 272
Применение импульса ... 272
Применение регуляризации ... 273
Задание числа скрытых блоков ... 273

Глава 8. Векторизация ... 274
Введение в векторизацию в машинном обучении ... 274
Зачем нужно векторизовать данные? ... 275
Стратегии обработки табличных исходных данных ... 277
Конструирование признаков и методы нормировки ... 279
Применение библиотеки DataVec для ETL и векторизации ... 285
Векторизация изображений ... 286
Представление изображений в DL4J ... 286
Нормировка данных изображения с помощью DataVec ... 288
Векторизация последовательных данных ... 289
Основные виды источников последовательных данных ... 289
Векторизация последовательных данных с помощью DataVec ... 290
Векторизация текста ... 294
Мешок слов ... 295
TF-IDF ... 296
Сравнение Word2Vec и векторной модели ... 299
Работа с графами ... 300

Глава 9. Глубокое обучение и DL4J на платформе Spark ... 301
Введение в использование DL4J совместно с Spark и Hadoop ... 301
Запуск Spark из командной строки ... 303
Конфигурирование и настройка Spark ... 305

Выполнение Spark в кластере Mesos ... 306
Выполнение Spark поверх YARN ... 307
Общее руководство по настройке Spark ... 309
Настройка задач DL4J для Spark ... 311
Подготовка проекта Maven для Spark и DL4J ... 312
Шаблон секции зависимостей в файле pom.xml ... 314
Настройка POM-файла для CDH 5.X ... 317
Настройка POM-файла для HDP 2.4 ... 317
Отладка Spark и Hadoop ... 318
Типичные проблемы при работе с ND4J ... 318
Параллельное выполнение DL4J на платформе Spark ... 319
Минимальный пример обучения на платформе Spark ... 320
Рекомендации по использованию DL4J API для Spark ... 322
Пример многослойного перцептрона на платформе Spark ... 323
Конфигурирование архитектуры МСП для Spark ... 326
Распределенное обучение и оценивание модели ... 327
Создание и выполнение задачи Spark ... 328
Порождение текстов в стиле Шекспира с помощью Spark и LSTM-сети ... 328
Конфигурирование архитектуры LSTM-сети ... 330
Обучение, наблюдение за ходом работы и интерпретация результатов ... 331
Моделирование набора MNIST с помощью сверточной нейронной сети в кластере Spark ... 332
Конфигурирование задачи Spark и загрузка набора данных MNIST ... 334
Конфигурирование и обучение СНС LeNet ... 335

Приложение A. Что такое искусственный интеллект? ... 337
Положение на данный момент ... 338
Определение глубокого обучения ... 338
Определение искусственного интеллекта ... 338
Зима не за горами ... 345

Приложение B. RL4J и обучение с подкреплением ... 347
Введение ... 347
Марковский процесс принятия решений ... 347
Терминология ... 348
Различные варианты ... 349
Безмодельное обучение ... 349
Наблюдаемое состояние ... 349
Однопользовательские и состязательные игры ... 349
Q-обучение ... 350
От политики к нейронным сетям ... 350
Перебор политик ... 352
Исследование и использование ... 355
Уравнение Беллмана ... 356
Выборка начальных состояний ... 357
Реализация Q-обучения ... 358
Моделирование Q(s,a) ... 359
Воспроизведение опыта ... 359
Сверточные слои и предварительная обработка изображений ... 360
Обработка истории ... 361
Двойное Q-обучение ... 361

Ограничение ... 362
Масштабирование вознаграждений ... 362
Приоритетное воспроизведение ... 362
График, визуализация и среднее значение Q ... 362
RL4J ... 365
Заключение ... 366

Приложение C. Числа, которые должен знать каждый ... 367

Приложение D. Нейронные сети и обратное распространение: математическое описание ... 368
Введение ... 368
Обратное распространение в многослойном перцептроне ... 369

Приложение E. ND4J API ... 372
Дизайн и основы использования ... 372
Что такое NDArray? ... 373
Общий синтаксис ND4J ... 374
Основы работы с массивами NDArray ... 375
Класс Dataset ... 377
Создание входных векторов ... 378
Основы создания векторов ... 378
Класс MLLibUtil ... 379
Преобразование INDArray в вектор MLLib ... 379
Преобразование вектора MLLib в INDArray ... 379
Получение предсказаний от модели в DL4J ... 380
Совместное использование DL4J и ND4J ... 380

Приложение F. Библиотека DataVec ... 382
Загрузка данных для машинного обучения ... 382
Загрузка CSV-данных для многослойного перцептрона ... 384
Загрузка изображений для сверточной нейронной сети ... 385
Загрузка последовательных данных для рекуррентных нейронных сетей ... 386
Подготовка данных средствами DataVec ... 387
Преобразования DataVec: основные понятия ... 388
Преобразования DataVec: пример ... 389

Приложение G. Работа с DL4J на уровне исходного кода ... 391
Проверка, установлен ли Git ... 391
Клонирование основных проектов, связанных с DL4J ... 391
Скачивание исходного кода в виде zip-файла ... 392
Сборка библиотеки из исходного кода с помощью Maven ... 392

Приложение H. Подготовка проектов на базе DL4J ... 393
Создание нового проекта на базе DL4J ... 393
Java ... 393
Работа с Maven ... 394
Интегрированные среды разработки (IDE) ... 395
Настройка других POM-файлов Maven ... 396
ND4J и Maven ... 396

Приложение I. Подготовка проектов на базе DL4J к работе с GPU 397
Переключение библиотек в режим работы с GPU 397
Выбор GPU 397
Обучение на системе с несколькими GPU 398
CUDA на разных платформах 398
Мониторинг производительности GPU 399

Приложение J. Отладка проблем с установкой DL4J 400
Предыдущая установка 400
Ошибки нехватки памяти при сборке из исходного кода 400
Старые версии Maven 400
Maven и переменная среды PATH 400
Недопустимые версии JDK 401
C++ и другие средства разработки 401
Windows и путь к включаемым файлам 401
Мониторинг GPU 401
Использование JVisualVM 401
Работа с Clojure 402
Поддержка чисел с плавающей точкой в OS X 402
Ошибка разветвления-соединения в Java 7 402
Предостережения 402
Клонирование других репозиториев 402
Проверьте зависимости Maven 403
Переустановка зависимостей 403
Если ничего не помогает 403
Различные платформы 403
OS X 403
Windows 403
Linux 404

Предметный указатель 405

Об авторах 416

Об иллюстрации на обложке 417

Посвящается моим сыновьям Этану, Гриффину и Дэйну:
идите вперед, будьте упорными и храбрыми.

– Дж. Гибсон

Предисловие

О структуре книги

В первых четырех главах излагаются теория и фундаментальные факты в объеме, достаточном для того, чтобы специалист-практик мог двигаться дальше. А в последних пяти главах на этой основе исследуются различные области глубокого обучения с помощью библиотеки DL4J:

- построение глубоких сетей;
- методы настройки сетей;
- векторизация для различных типов данных;
- реализация процессов глубокого обучения на платформе Spark.

DL4J – сокращение для Deeplearning4j

В этой книге названия DL4J и Deeplearning4j считаются синонимами. Оба они относятся к набору инструментов в библиотеке Deeplearning4j.

Мы организовали материал именно таким образом, потому что ощущали потребность в книге, которая содержала бы «достаточно теории» и вместе с тем была бы достаточно практичной для построения процессов глубокого обучения производственного уровня. Мы полагаем, что выбранный гибридный подход хорошо отвечает поставленной цели.

Глава 1 содержит обзор концепций машинного обучения вообще и глубокого обучения в частности с целью ввести читателя в курс дела и сообщить достаточно сведений для понимания остальной части книги. Мы включили эту главу, чтобы начинающие могли освежить подзабытое или познакомиться с новыми для себя идеями, и тем самым хотели сделать книгу доступной максимально широкой аудитории.

В главе 2, основанной на материале главы 1, излагаются основы нейронных сетей. По сути своей она посвящена теории нейронных сетей, но мы старались представить информацию в доступной форме. В главе 3 описывается, как из нейронных сетей вырастают глубокие сети. В главе 4 представлены четыре основные архитектуры глубоких сетей, это фундамент для понимания последующих глав.

Глава 5 содержит ряд примеров Java-кода, в которых используются методы из первой части книги. В главах 6 и 7 рассматриваются фундаментальные основы настройки нейронных сетей общего вида, а затем эти знания применяются к настройке специальных архитектур глубоких сетей. Эти главы платформенно-независимы и в равной мере относятся к любой библиотеке глубокого обучения. В главе 8 приводятся обзор методов векторизации и основы работы с библиотекой DataVec (средством ETL и векторизации, входящим в состав DL4J). Глава 9 завершает основной текст книги обзором применения DL4J в системах Spark и Hadoop – здесь вы найдете реальные примеры, которые сможете выполнить в собственном кластере Spark.

В книге много приложений, в которых рассматриваются важные темы, не нашедшие отражения в основных главах, а именно:

- искусственный интеллект;
- использование Maven в проектах на основе DL4J;
- работа с GPU;
- использование ND4J API
- и прочее.

Кто такой «практик»?

В настоящее время термин «наука о данных» (data science) не имеет четкого определения и зачастую означает разные вещи. Мир науки о данных и искусственного интеллекта (ИИ) широк и расплывчат, как и многие термины в современной информатике. Связано это с тем, что машинное обучение проникло почти во все дисциплины.

У этого широкого проникновения есть исторические параллели с тем, как в 1990-х годах Всемирная паутина проникла во все дисциплины и привела много новых людей в область технологии. Так и теперь самые разные специалисты – инженеры, статистики, аналитики, люди творческих профессий – каждодневно пополняют ряды поборников машинного обучения. Эта книга призвана сделать глубокое (и машинное) обучение доступным самой широкой аудитории.

Если эта тематика вам интересна и вы читаете это предисловие – значит, *вы практик, и эта книга для вас*.

Кому стоит прочитать эту книгу?

Мы решили начать книгу не с игрушечных примеров, которые затем постепенно обрастают деталями, а с фундаментальных основ, которые позволят в полной мере погрузиться в глубокое обучение.

Нам кажется, что во многих книгах опущены базовые вопросы, с которыми каждый практик должен хотя бы бегло познакомиться. Опираясь на свой опыт работы в области машинного обучения, мы решили включить материал, который необходим начинающему практику для успешной работы над своими проектами в сфере глубокого обучения.

Возможно, вы захотите пропустить первые две главы и сразу перейти к основам глубокого обучения. Но мы полагаем, что многие оценят наличие вводных сведений, поскольку это делает гладким переход к основанным на них более трудным темам. Ниже мы предлагаем различные подходы к чтению книги в зависимости от подготовки читателя.

Практический специалист по машинному обучению в коммерческой компании

В этой категории мы выделяем две подгруппы:

- специалист по анализу данных (data scientist);
- Java-программист.

Специалист по анализу данных

Эти люди уже строили модели раньше и свободно ориентируются в науке о данных. Если вы из их числа, то главу 1 можете пропустить, а главу 2 бегло просмот-

реть. Мы рекомендуем сразу перейти к главе 3, поскольку вы, скорее всего, достаточно подготовлены к знакомству с основаниями глубоких сетей.

Java-программист

Java-программистам обычно поручают интегрировать код машинного обучения в производственные системы. Если это как раз к вам и относится, то вас, наверное, заинтересует глава 1, поскольку она позволит познакомиться с терминологией, применяемой в науке о данных. Вам также будет интересно приложение E, поскольку код интеграции модельных оценок обычно не обходится без ND4J API.

Руководитель предприятия

Среди рецензентов нашей книги были и руководители компаний из списка Fortune 500, им материал понравился, т. к. позволил лучше понять, что вообще происходит в области глубокого обучения. Один из них отметил, что после окончания колледжа уже «прошло какое-то время», так что глава 1 пришлась весьма кстати для освежения памяти. Если вы руководитель, то рекомендуем начать с беглого прочтения главы 1, чтобы вспомнить кое-какую терминологию. Но, вероятно, вам стоит пропустить главы, посвященные API и конкретным примерам.

Ученый

Если вы работаете в академическом учреждении, то главы 1 и 2 вряд ли будут вам интересны, т. к. в высшей школе вы это уже проходили. Зато вас наверняка заинтересуют главы о настройке нейронных сетей вообще и для конкретной архитектуры, поскольку эта информация основана на результатах исследований и применима ко всем реализациям глубокого обучения. Тем, кто предпочитает высокопроизводительную линейную алгебру на виртуальной машине Java (JVM), будет также интересно обсуждение библиотеки ND4J.

Графические выделения

В книге применяются следующие графические выделения.

Курсив

Новые термины, URL-адреса, адреса электронной почты, имена и расширения имен файлов.

`Моноширинный`

Листинги программ, а также элементы кода в основном тексте: имена переменных и функций, базы данных, типы данных, переменные окружения, предложения и ключевые слова языка. Также применяется для набора имен модулей и пакетов, команд или иного текста, которые следует вводить буквально, и результатов работы команд.

`Моноширинный курсив`

Текст, вместо которого следует подставить значения, заданные пользователем или определяемые контекстом.

Так обозначается примечание общего характера.

Так обозначается предупреждение или предостережение.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com или www.дмк.рф на странице с описанием соответствующей книги.

Мы высоко ценим, хотя и не требуем, ссылки на наши издания. В ссылке обычно указываются название книги, имя автора, издательство и ISBN, например: «Программирование на языке Rust» Джима Блэнди, Джейсона Орендорф (O'Reilly, ДМК Пресс). Copyright © 2018 Jim Blandy and Jason Orendorff, 978-1-491-92728-1 (англ.), 978-5-97060-236-2 (рус.).

Если вы полагаете, что планируемое использование кода выходит за рамки изложенной выше лицензии, пожалуйста, обратитесь к нам по адресу dmkpress@gmail.com.

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте www.dmkpress.com, зайдя на страницу книги, и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в тексте или в коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Благодарности

От Джоша

В отборе материала для этой книги и ее рецензировании я опирался на помощь людей куда умнее меня. Проект такого размера, как DL4J, не может существовать в вакууме, поэтому в формулировании многих идей и рекомендаций я полагался на мнение экспертов из сообщества и инженеров компании Skymind.

Я никогда не думал, что работа над тем, что впоследствии превратилось в DL4J, вместе с Адамом (после случайной встречи на конференции MLConf) когда-ни-

будь воплотится в книгу. Честно говоря, хотя я стоял у истоков DL4J, Адам сделал куда больше записей в репозиторий, чем я. Поэтому я чрезвычайно признателен Адаму за его преданность проекту, самой идее глубокого обучения на JVM и за то, что он не отступился на начальном этапе, когда все было зыбко и неясно. И да, ты был прав: ND4J оказалась отличной идеей.

Написание книги может стать долгим путешествием в одиночестве, поэтому я особенно благодарен Алексу Блэку, который не только потратил массу времени на рецензирование, но и предложил материал для приложений. Энциклопедические познания Алекса касательно опубликованной литературы по нейронным сетям были чрезвычайно важны при шлифовке многих мелких деталей и стали гарантией правильности изложения в целом и в частности. Без Алекса главы 6 и 7 и наполовину не были бы тем, чем являются теперь.

Сьюзан Эрали (Susan Eraly) оказала ценнейшую помощь в написании раздела о функциях потерь, предоставила материал для приложений (многие уравнения, приведенные в этой книге, были выправлены Сьюзан) и сделала много замечаний в процессе рецензирования. Мелани Уоррик (Melanie Warrick) сыграла основную роль в рецензировании ранних вариантов книги, делилась своими замечаниями и рассказала немало интересного о внутренних механизмах работы сверточных нейронных сетей (CNN).

Дэвид Кэйл (David Kale) частенько оставлял замечания, не пропуская небрежностей в отношении деталей работы сетей и библиографических ссылок. Дэйв всегда представлял академический взгляд на необходимый уровень строгости с учетом того, для какой аудитории мы пишем.

Джеймс Лонг (James Long) критически внимал моим тирадам на тему того, что включать в книгу, а что опустить, и представлял точку зрения статистика-практика. Нередко, когда было не ясно, как лучше изложить какой-то сложный вопрос, Джеймс выступал в роли слушателя, на котором я опробовал обсуждение с разных сторон. Если Дэвид Кэйл и Алекс Блэк зачастую напоминали мне о необходимости математической строгости, то Джеймс играл роль рационального адвоката дьявола, не допускающего, чтобы читатель «утонул в математических деталях».

Вячеслав Кокорин (Raver) привнес свои знания в разработку примеров из области обработки естественного языка и технологии Word2Vec.

Хотел бы отметить поддержку со стороны Криса Николсона (Chris Nicholson), генерального директора компании Skymind. Крис оказывал поддержку этой книге на каждом этапе, и в значительной степени благодаря ему у нас были время и ресурсы для завершения работы над ней.

Я выражаю благодарность всем, кто предлагал материалы для приложений: Алексу Блэку (обратное распространение, DataVec), Вячеславу Кокорину (GPU), Сьюзан Эрали (GPU) и Рубену Фижелю (Ruben Fiszel) (обучение с подкреплением). На различных этапах в рецензировании книги принимали также участие Гран Ингерсол (Grant Ingersol), Дин Уэмплер (Dean Wampler), Роберт Чонг (Robert Chong), Тэд Маласка (Ted Malaska), Райан Джено (Ryan Geno), Ларс Джордж (Lars George), Сунеель Мартхи (Suneel Marthi), Франсу Гарилло (Francois Garillot) и Дон Браун (Don Brown). Все ошибки, которые могли остаться в книге, – целиком и полностью моя вина.

Я благодарен нашему уважаемому редактору Тиму Макговерну (Tim McGovern) за отзывы, замечания и просто терпение – ведь проект растянулся на годы и стал

больше на три главы. Мы высоко ценим, что он позволил нам сорвать сроки, но сделать все как надо.

Ниже перечислены другие люди, которые оказали влияние на мою карьеру, приведшую к появлению этой книги: мои родители (Льюис и Конни), д-р Энди Новобилиски (Andy Novobiliski) (аспирантура), д-р Мина Сартипи (Dr. Mina Sartipi) (научный руководитель), д-р Билли Харрис (Billy Harris) (курс алгоритмов для аспирантов), д-р Джо Дьюмас (Joe Dumas) (аспирантура), Ричи Кэрролл (Ritchie Carroll) (создатель openPDC), Пол Трачан (Paul Trachian), Кристоф Бисиглиа (Christophe Bisciglia) и Майк Олсон (Mike Olson) (затащившие меня в компанию Cloudera), Малком Рэйми (Malcom Ramey) (за мою первую работу в качестве программиста), университет Теннесси в Чаттануге и пиццерию Лупи (где я питался во время учебы в аспирантуре).

И последними по порядку, но отнюдь не по значимости я хочу поблагодарить свою жену Лесли и сыновей Этана, Гриффина и Дэйна за терпение, с которым они переносили мою работу по ночам, а иногда и во время отпуска.

От Адама

Я благодарен своей команде в компании Skymind за ту огромную работу, которую они проделали, рецензируя бесчисленные варианты книги. Особое спасибо Крису, который мирился с моей безумной идеей писать книгу, одновременно пытаясь основать стартап.

Библиотека родилась в 2013 году со случайной встречи с Джошем на конференции MLConf и выросла в проект, который ныне используется во всем мире. Благодаря DL4J я поездил по миру и открыл для себя целый мир новых впечатлений.

Прежде всего я хочу поблагодарить своего соавтора Джоша Паттерсона, который взял на себя львиную долю работы над книгой и заслуживает всяческой признательности. Он тратил вечера и выходные, торопясь выпустить книгу в свет, пока я продолжал работать над кодом и включал в него все новые возможности.

Вслед за Джошем отдаю должное многим нашим коллегам и соавторам – тем, кто был с нами с самого начала, как Алекс, Вячеслав Кокорин (Raver), и тем, кто, как Дэйв, присоединился позже и зорко следил за правильностью математики.

Тим Макговерн оказался прекрасным слушателем, на котором я опробовал некоторые свои завиральные идеи относительно книг для издательства O'Reilly, а также любезно позволил мне самому выбрать название книги.

Глава 1

Обзор машинного обучения

Сконденсировать факт из туманности нюансов.

– Нил Стивенсон «Лавина»

Обучающиеся машины

Интерес к машинному обучению резко возрос в последние десять лет. Машинное обучение включают в программы факультетов информатики, по нему проводят конференции, а в заголовках «Wall Street Journal» оно встречается чуть ли не ежедневно. Говоря о машинном обучении, многие путают два вопроса: что оно действительно может и чего от него хотели бы. По большому счету, *машинное обучение* – это применение алгоритмов, которые извлекают информацию из исходных данных и представляют ее в виде той или иной *модели*. Эта модель используется, чтобы вывести другие данные, которые в модели отсутствуют.

Нейронные сети – один из типов моделей машинного обучения, они существуют уже по меньшей мере 50 лет. Фундаментальной единицей нейронной сети является *блок*, или *узел*, – приблизительный аналог биологического нейрона в мозге млекопитающих. Модель связей между нейронами также основана на работе биологического мозга, равно как и эволюция связей с течением времени (в результате «обучения»). В следующих двух главах мы будем подробно рассматривать функционирование таких моделей.

В середине 1980-х и в начале 1990-х годов произошли важные подвижки в области архитектуры нейронных сетей. Но для получения хороших результатов требовалось очень много данных и времени, что замедляло их практическое внедрение, так что постепенно интерес угас. В начале 2000-х годов наблюдался экспоненциальный рост вычислительной мощности, и промышленность стала свидетелем «кембрийского взрыва» методов вычислений – ничего подобного раньше не было и в помине. Глубокое обучение, появившееся в результате взрывного роста вычислительной мощности в этом десятилетии, стало серьезным игроком и победителем многих важных состязаний по машинному обучению. В 2017 году интерес не спадает, сегодня глубокое обучение можно встретить в любом закоулке машинного обучения.

Ниже мы еще вернемся к нашему определению глубокого обучения. Эта книга построена так, чтобы практик, например вы, мог снять ее с полки и сделать следующее:

- получить общие сведения об основах линейной алгебры и машинного обучения;

- получить общие сведения об основах нейронных сетей;
- изучить четыре основные архитектуры глубоких сетей;
- воспользоваться примерами кода для исследования вариантов глубоких сетей на практике.

Мы надеемся, что материал покажется вам практичным и не слишком сложным. И давайте начнем с вопроса о том, что же такое машинное обучение, а также с базовых понятий, которые помогут лучше понять, о чем говорится в книге.

Как машины могут обучаться?

Чтобы определить, как машины могут обучаться, нужно определить, что мы понимаем под «обучением». В быту мы имеем в виду «приобретение знаний посредством самостоятельного изучения, на опыте или в процессе общения с преподавателем». Немного сместив фокус, можно считать, что машинное обучение – это применение алгоритмов для получения структурных описаний из примеров данных. Компьютер обучается чему-то, относящемуся к информационным структурам, содержащимся в исходных данных. Структурные описания – другое название моделей, которые служат для представления информации, извлеченной из данных. Эти структуры, или модели, можно использовать для предсказания неизвестных данных. Структурные описания (или модели) могут принимать разную форму, в том числе:

- решающие деревья;
- линейная регрессия;
- веса связей в нейронных сетях.

Модель каждого типа определяет свой способ применения правил к известным данным для предсказания неизвестных. В случае решающих деревьев создается древовидный набор правил, а в случае линейных моделей – набор параметров, представляющих входные данные.

В нейронных сетях имеется так называемый *вектор параметров*, представляющий веса связей между узлами сети. Ниже в этой главе мы опишем этот тип моделей подробнее.

Машинное обучение и добыча данных

Термин *добыча данных*[1] существует уже много десятилетий и, как многие термины машинного обучения, понимается или используется неправильно. В этой книге под «добычей данных» мы будем понимать «извлечение информации из данных». Отличие машинного обучения состоит в том, что оно относится к алгоритмам, которые применяются в процессе добычи данных для получения структурных описаний из исходных данных. Вот простая интерпретация добычи данных:

- для обучения концепциям нам нужны примеры исходных данных;
- примеры состоят из строк или экземпляров данных, из которых видны определенные закономерности в данных;

[1] В отечественной литературе устоялся термин «добыча данных», хотя он совершенно не отражает смысла понятия. Английское слово «mining» означает «добыча полезных ископаемых» с ударением на слове «полезных», т. е. «извлечение руды из породы». Правильнее было бы говорить о «добыче информации» или прибегнуть к описательному переводу типа «глубокий анализ данных», но мы будем придерживаться сложившейся терминологии. – *Прим. перев.*

- машина обучается концепциям на этих закономерностях посредством алгоритмов машинного обучения.

В целом этот процесс можно считать «добычей данных».

Артур Сэмюэл (Arthur Samuel), пионер в области искусственного интеллекта (ИИ), работавший в компании IBM и в Стэнфордском университете, определял машинное обучение следующим образом:

> Область исследований, посвященная наделению компьютеров способностью обучаться без явного программирования.

Сэмюэл разработал программу, которая могла играть в шашки и адаптировала свою стратегию по мере того, как обучалась ассоциировать вероятность выигрыша или проигрыша с определенными позициями на доске. Эта фундаментальная схема поиска паттернов, ведущих к победе или к поражению, с последующим распознаванием и подкреплением успешных паттернов и по сей день лежит в основе машинного обучения и ИИ.

Концепция машин, способных самостоятельно обучаться для достижения целей, поражала наше воображение в течение многих десятилетий. Наверное, наилучшим образом она была выражена дедушками современного ИИ, Стюартом Расселом (Stuart Russell[2]) и Питером Норвигом (Peter Norvig[3]), в книге «Artificial Intelligence: A Modern Approach»[4]:

> Как может медленный, крохотный мозг, не важно, биологический или электронный, воспринимать, понимать, предсказывать и манипулировать миром, который гораздо больше и сложнее его самого?

Эта цитата отсылает нас к идее о том, что концепции обучения были заимствованы у процессов и алгоритмов, подсмотренных в природе. На рис. 1.1 наглядно изображено наше представление о связи между ИИ, машинным обучением и глубоким обучением.

Рис. 1.1 ❖ Связь между ИИ и глубоким обучением

[2] https://www2.eecs.berkeley.edu/Faculty/Homepages/russell.html.

[3] http://norvig.com/.

[4] *Стюарт Р., Норвиг П.* Искусственный интеллект. Современный подход. М.: Вильямс, 2017.

Область ИИ широка и давно уже является предметом исследований. Глубокое обучение – это часть машинного обучения, которая, в свою очередь, является частью ИИ. Теперь кратко рассмотрим другой источник глубокого обучения: связь искусственных нейронных сетей с биологией.

Биологические корни

Биологическая нейронная сеть (мозг) содержит приблизительно 86 миллиардов нейронов, каждый из которых соединен с большим числом других.

Общее число связей в мозге человека
По осторожной оценке исследователей, между нейронами в мозге человека существуют более 500 триллионов связей. Самые крупные современные искусственные нейронные сети даже отдаленно не приближаются к этой величине.

С точки зрения обработки информации, биологический нейрон представляет собой возбуждаемый блок, который может обрабатывать и передавать информацию с помощью электрических и химических сигналов. Нейрон считается основным компонентом головного мозга, спинного мозга и ганглий периферийной нервной системы. Как мы увидим ниже, искусственные нейронные сети структурно гораздо проще.

Сравнение биологической и искусственной нейронных сетей
Биологические нейронные сети гораздо (на несколько порядков) сложнее искусственных!

У искусственных нейронных сетей есть два свойства, отражающих общие представления о работе мозга. Во-первых, самой мелкой единицей нейронной сети является *искусственный нейрон* (или, для краткости, *блок*). Образцом для искусственного нейрона служит биологический нейрон мозга: как и биологический нейрон, он возбуждается в результате поступления входных сигналов. Искусственные нейроны передают часть полученной информации (но не всю) другим искусственным нейронам, часто в преобразованном виде. Далее в этой главе мы детально рассмотрим эти преобразования в контексте нейронных сетей.

Во-вторых, как биологические нейроны можно научить передавать дальше только сигналы, полезные для достижения более общих целей мозга, так и искусственные нейроны можно обучить передавать только полезные сигналы. Далее мы увидим, как эти идеи позволяют искусственной нейронной сети моделировать биологическую посредством битов и функций.

Биологические идеи в информатике

Применение биологических идей в информатике не ограничивается искусственными нейронными сетями. За последние 50 лет предметом исследований, нашедших отражение в информатике, стали и другие природные объекты, например:

- муравьи:
- термиты[5];
- пчелы;
- генетические алгоритмы.

[5] Patterson, 2008. TinyTermite: A Secure Routing Algorithm; Sartipi and Patterson, 2009. TinyTermite: A Secure Routing Algorithm on Intel Mote 2 Sensor Network Platform.

Например, муравейник рассматривался учеными как мощный децентрализованный компьютер, в котором ни один отдельный муравей не является точкой общего отказа[6]. Муравьи постоянно переключаются с одной задачи на другую в поисках близких к оптимальному решений задачи балансировки нагрузки с помощью таких метаэвристик, как количественная стигмергия. Колонии муравьев способны убирать территорию, обороняться, строить гнездо и добывать пищу, выделяя для каждой задачи почти оптимальное число работников, основанное на относительной потребности, при этом не существует какого-то одного муравья, координирующего работу.

Что такое глубокое обучение?

Дать определение глубокому обучению не так-то просто, потому что его формы медленно изменялись на протяжении последнего десятилетия. Одно из полезных определений гласит, что глубокое обучение имеет дело с «нейронными сетями, имеющими более двух слоев». Проблема в том, что такие сети считались глубокими где-то в 1980-х годах. Но нам кажется, что для демонстрации впечатляющих результатов, наблюдаемых в последние годы, нейронные сети архитектурно должны были перерасти ранние формы (что стало возможно благодаря значительному увеличению вычислительной мощности). Перечислим некоторые аспекты эволюции нейронных сетей:

- больше нейронов, чем в предшествующих сетях;
- более сложные способы соединения нейронов и слоев;
- резкий рост вычислительной мощности, доступной для обучения;
- автоматическое выделение признаков.

В этой книге мы будем определять глубокую сеть как нейронную сеть с большим числом параметров и слоев, имеющую одну из четырех фундаментальных архитектур:

- сети, предобученные без учителя;
- сверточные нейронные сети;
- рекуррентные нейронные сети;
- рекурсивные нейронные сети.

Существует ряд вариаций этих архитектур, например гибридная сверточно-рекуррентная нейронная сеть. Но в этой книге мы сосредоточимся только на четырех вышеперечисленных архитектурах.

Автоматическое выделение признаков – еще одно крупное преимущество глубокого обучения над традиционными алгоритмами машинного обучения. Под этим мы понимаем процесс определения сетью тех характеристик набора данных, которые можно надежно использовать для пометки данных. Исторически специалисты по машинному обучению тратили месяцы, годы, а иногда и десятки лет жизни на то, чтобы вручную создать исчерпывающий набор признаков для классификации данных. Со времени Большого взрыва глубокого обучения, начавшегося в 2006 году, необходимость тратить годы ручного труда исчезла, поскольку новейшие алгоритмы машинного обучения способны сами находить признаки, по которым можно классифицировать входные данные. По точности глубокое обучение превзошло традиционные алгоритмы почти для всех типов данных, требуя

[6] https://mitpress.mit.edu/books/ant-colony-optimization.

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине
«Электронный универс»
e-Univers.ru