

Содержание

Предисловие	11
Введение	13
Благодарности	14
Об авторах	15
Об иллюстрации на обложке	16
Об этой книге	17
Часть I. ОСНОВЫ	22
На пути к глубокому обучению: введение в машинное обучение	23
1.1. Что такое машинное обучение?	24
1.1.1. Связь машинного обучения и искусственного интеллекта	26
1.1.2. Что можно и чего нельзя сделать с помощью машинного обучения	27
1.2. Пример машинного обучения	27
1.2.1. Использование машинного обучения в приложениях	30
1.2.2. Обучение с учителем	31
1.2.3. Обучение без учителя	33
1.2.4. Обучение с подкреплением	33
1.3. Глубокое обучение	35
1.4. Что вы узнаете из этой книги?	36
1.5. Резюме	37
Глава 2. Игра го как проблема машинного обучения	38
2.1. Почему игры?	38
2.2. Краткое введение в игру го	39
2.2.1. Описание доски	39
2.2.2. Размещение и захват камней	40
2.2.3. Завершение игры и подсчет очков	41
2.2.4. Правило ко	43
2.3. Форa	44
2.4. Дополнительные ресурсы	44
2.5. Чему можно научить машину?	44
2.5.1. Выбор ходов в дебюте	45
2.5.2. Поиск игровых состояний	45
2.5.3. Сокращение количества рассматриваемых ходов	45
2.5.4. Оценка игровых состояний	46
2.6. Определение силы ИИ для игры в го	47
2.6.1. Традиционные ранги го	47
2.6.2. Сравнительный анализ вашего ИИ для игры в го	48
2.7. Резюме	48

Глава 3. Реализация первого бота для игры в го	49
3.1. Представление игры го средствами языка Python	49
3.1.1. Реализация доски для игры в го	52
3.1.2. Отслеживание связанных групп камней в игре го	52
3.1.3. Размещение и захват камней на доске для игры в го	54
3.2. Фиксация игрового состояния и проверка допустимости ходов	56
3.2.1. Самозахват	57
3.2.2. Правило ко	58
3.3. Завершение игры	60
3.4. Создание первого слабого бота для игры в го	63
3.5. Ускорение игрового процесса с помощью Zobrist-хеширования	66
3.6. Игра против собственного бота	71
3.7. Резюме	73
Часть II. МАШИННОЕ ОБУЧЕНИЕ И ИГРОВОЙ ИИ	74
Глава 4. Игры и поиск по дереву	75
4.1. Классификация игр	76
4.2. Прогнозирование действий противника с помощью алгоритма минимаксного поиска	77
4.3. Крестики-нолики: пример использования минимаксного алгоритма	80
4.4. Сокращение пространства поиска путем редукции	83
4.4.1. Сокращение глубины поиска с помощью оценки позиции	85
4.4.2. Сокращение ширины поиска путем альфа-бета-отсечения	88
4.5. Оценка игрового состояния методом Монте-Карло	92
4.5.1. Реализация алгоритма Монте-Карло средствами языка Python	96
4.5.2. Выбор ветви для исследования	99
4.5.3. Применение алгоритма Монте-Карло к игре го	101
4.6. Резюме	103
Глава 5. Знакомство с нейронными сетями	105
5.1. Простой пример использования: классификация рукописных цифр	106
5.1.1. Набор данных MNIST	106
5.1.2. Предварительная обработка данных MNIST	107
5.2. Основы нейронных сетей	114
5.2.1. Логистическая регрессия как простая искусственная нейронная сеть	114
5.2.2. Сети с несколькими размерностями выходного сигнала	114
5.3. Сети прямого распространения	116
5.4. Оценка предсказаний. Функции потерь и оптимизация	119
5.4.1. Что такое функция потерь?	119
5.4.2. Среднеквадратическая ошибка	119
5.4.3. Поиск минимумов функции потерь	120
5.4.4. Градиентный спуск для нахождения минимумов	121
5.4.5. Стохастический градиентный спуск для функций потерь	123
5.4.6. Метод обратного распространения ошибки	124
5.5. Обучение нейронной сети средствами языка Python	127
5.5.1. Слои нейронной сети в Python	127

5.5.2. Слои активации в нейронных сетях	129
5.5.3. Плотные слои в Python как компоненты сетей прямого распространения	130
5.5.4. Создание последовательных нейронных сетей средствами языка Python	131
5.5.5. Применение сети к задаче классификации рукописных цифр	134
5.6. Резюме	135

Глава 6. Создание нейронной сети для данных игры го

6.1. Кодирование игрового состояния для подачи на вход нейронной сети.....	139
6.2. Генерирование обучающих игровых данных методом поиска по дереву	142
6.3. Использование библиотеки глубокого обучения Keras	144
6.3.1. Принципы проектирования с помощью библиотеки Keras	145
6.3.2. Установка библиотеки глубокого обучения Keras	145
6.3.3. Применение библиотеки Keras к рассмотренному ранее примеру.....	146
6.3.4. Предсказание ходов в игре го с помощью нейронной сети прямого распространения и библиотеки Keras	148
6.4. Анализ пространства с помощью сверточных сетей.....	152
6.4.1. Назначение сверточных слоев	152
6.4.2. Создание сверточных сетей с помощью библиотеки Keras	156
6.4.3. Сокращение пространственной размерности с помощью слоев пулинга	157
6.5. Предсказание вероятностей для ходов в игре го	158
6.5.1. Использование функции активации softmax в последнем слое	159
6.5.2. Перекрестная энтропия как функция потерь для задач классификации	159
6.6. Создание более глубоких сетей с помощью прореживания и блоков линейной ректификации	162
6.6.1. Исключение нейронов методом регуляризации	162
6.6.2. Функция активации ReLU	163
6.7. Объединяем все вместе и создаем более мощную сеть для предсказания ходов в игре го	164
6.8. Резюме	167

Глава 7. Глубокое обучение бота на основе данных.....

7.1. Импорт записей партий в го.....	170
7.1.1. Формат файла SGF	171
7.1.2. Загрузка и воспроизведение партий в го с сервера KGS.....	172
7.2. Подготовка игровых данных для глубокого обучения.....	173
7.2.1. Воспроизведение партии в го на основе ее записи в формате SGF	173
7.2.2. Создание обработчика данных игры го	175
7.2.3. Создание генератора данных игры го для обеспечения их эффективной загрузки	182
7.2.4. Параллельная обработка игровых данных и генераторы.....	183
7.3. Обучение глубокой сети на основе партий, сыгранных человеком	184
7.4. Создание более реалистичных кодировщиков данных игры го	189

7.5. Эффективное обучение с помощью адаптивных градиентов	191
7.5.1. Затухание и импульс в СГС	191
7.5.2. Оптимизация нейронных сетей с помощью метода Adagrad	192
7.5.3. Уточнение адаптивных градиентов с помощью Adadelta	194
7.6. Проведение экспериментов и оценка эффективности	194
7.6.1. Руководство по тестированию архитектур и гиперпараметров	195
7.6.2. Оценка показателей производительности для обучающих и тестовых данных.....	197
7.7. Резюме	198
Глава 8. Развертывание ботов	199
8.1. Создание агента для предсказания ходов на основе глубокой нейронной сети.....	200
8.2. Создание веб-интерфейса для бота	202
8.2.1. Пример бота для игры в го	205
8.3. Обучение и развертывание бота для игры в го в облаке	206
8.4. Взаимодействие с другими ботами по протоколу Go Text Protocol.....	207
8.5. Локальные игры против других ботов.....	209
8.5.1. Пропуск хода и выход из игры	210
8.5.2. Запуск игры бота против других ботов.....	211
8.6. Развертывание бота для игры в го на онлайн-сервере	216
8.6.1. Регистрация бота на онлайн-сервере для игры в го	219
8.7. Резюме.....	219
Глава 9. Обучение на практике: обучение с подкреплением.....	221
9.1. Цикл обучения с подкреплением.....	222
9.2. Данные опыта.....	223
9.3. Создание обучающегося агента	226
9.3.1. Сэмплирование из распределения вероятностей	227
9.3.2. Обрезка распределения вероятностей	229
9.3.3. Инициализация агента	229
9.3.4. Загрузка агента с диска и его сохранение на диск.....	230
9.3.5. Реализация функции выбора хода.....	231
9.4. Игра бота с самим собой: практика компьютерной программы.....	233
9.4.1. Представление данных опыта.....	233
9.4.2. Симуляция игр	235
9.5. Резюме	237
Глава 10. Обучение с подкреплением и градиенты политики.....	239
10.1. Выявление хороших решений с помощью случайных игр	240
10.2. Изменение политик нейронной сети методом градиентного спуска.....	244
10.3. Советы по обучению бота на основе его игры с самим собой	248
10.3.1. Оценка прогресса	248
10.3.2. Измерение небольших различий в силе.....	249
10.3.3. Настройка алгоритма стохастического градиентного спуска (СГС).....	250
10.4. Резюме	254

Глава 11. Обучение с подкреплением и методы на основе ценности действий	255
11.1. Игры и алгоритм Q-обучения	256
11.2. Реализация алгоритма Q-обучения в Keras	260
11.2.1. Создание сетей с двумя входами с помощью Keras	260
11.2.2. Реализация ϵ -жадной политики с помощью Keras	264
11.2.3. Обучение сети, реализующей функцию ценности действия	267
11.3. Резюме	268
Глава 12. Обучение с подкреплением и методы типа «актор – критик»	269
12.1. Преимущество позволяет выявить важные решения	270
12.1.1. Что такое преимущество?	270
12.1.2. Вычисление преимущества в процессе игры бота с самим собой	272
12.2. Создание нейронной сети для обучения методом «актор – критик»	274
12.3. Игра с агентом типа «актор – критик»	277
12.4. Обучение агента типа «актор – критик» на данных опыта	278
12.5. Резюме	283
Часть III. БОЛЬШЕ, ЧЕМ СУММА ВСЕХ ЧАСТЕЙ	284
Глава 13. AlphaGo: собираем все вместе	285
13.1. Обучение глубоких нейронных сетей для создания бота AlphaGo	288
13.1.1. Сетевые архитектуры, используемые в программе AlphaGo	288
13.1.2. Кодировщик доски AlphaGo	290
13.1.3. Обучение сетей политики в стиле AlphaGo	293
13.2. Бутстрэппинг игр бота с самим собой из сетей политики	295
13.3. Создание сети ценности на основе данных, полученных в ходе игры бота с самим собой	296
13.4. Повышение эффективности поиска с помощью сетей политики и ценности	297
13.4.1. Нейронные сети и развертывания ММК	298
13.4.2. Поиск по дереву с помощью комбинированной функции ценности	299
13.4.3. Реализация алгоритма поиска AlphaGo	302
13.5. Практические советы, касающиеся обучения бота AlphaGo	307
13.6. Резюме	308
Глава 14. AlphaGo Zero: интеграция поиска по дереву и обучения с подкреплением	310
14.1. Создание нейронной сети для поиска по дереву	311
14.2. Управление процессом поиска по дереву с помощью нейронной сети	313
14.2.1. Спуск по дереву	316
14.2.2. Расширение дерева	319
14.2.3. Выбор хода	321
14.3. Обучение	322
14.4. Повышение эффективности разведки с помощью шума Дирихле	326

14.5. Современные методы создания более глубоких нейронных сетей.....	327
14.5.1. Пакетная нормализация.....	328
14.5.2. Остаточные сети.....	328
14.6. Дополнительные ресурсы.....	329
14.7. Заключение.....	330
14.8. Резюме.....	331
Приложение А. Математические основы.....	332
Векторы, матрицы, и не только: основные конструкции линейной алгебры.....	332
Векторы: одномерные данные.....	333
Матрицы: двумерные данные.....	334
Тензоры 3-го ранга.....	335
Тензоры 4-го ранга.....	337
Математический анализ за пять минут: производные и нахождение максимума.....	337
Приложение Б. Алгоритм обратного распространения ошибки.....	340
Пара слов о нотации.....	340
Алгоритм обратного распространения ошибки для сетей прямого распространения.....	341
Обратное распространение ошибки для последовательных нейронных сетей.....	342
Обратное распространение ошибки для всех нейронных сетей в целом.....	343
Вычислительные сложности, связанные с обратным распространением ошибки.....	343
Приложение В. Программы и серверы для игры в го.....	345
Программы для игры в го.....	345
GNU Go.....	345
Pachi.....	346
Серверы для игры в го.....	346
OGS.....	346
IGS.....	347
Tygem.....	347
Приложение Г. Обучение и развертывание ботов с помощью Amazon Web Services.....	348
Обучение моделей на сервисе AWS.....	355
Размещение бота на сервисе AWS с помощью протокола HTTP.....	356
Приложение Д. Отправка бота на онлайн-сервер для игры в го.....	358
Регистрация и активация бота на сервере OGS.....	358
Локальное тестирование OGS-бота.....	360
Развертывание OGS-бота на сервисе AWS.....	362
Предметный указатель.....	365

Предисловие

Для нас, членов команды AlphaGo, история этого алгоритма стала главным приключением всей жизни. Все началось, как это часто бывает, с небольшого шага – обучения простой сверточной нейронной сети на записях партий в го, сыгранных сильными игроками-людьми. Это привело к кардинальным прорывам в области машинного обучения, а также подарило нам несколько незабываемых событий, включая матчи против таких грозных профессиональных игроков, как Фань Хуэй, Ли Седоль и Ки Джи. Мы гордимся тем, что эти матчи не только повлияли на ма-неру игры в го по всему миру, но и привлекли внимание множества людей к теме искусственного интеллекта.

Но почему, спросите вы, нас должны интересовать игры? Подобно тому как дети используют игры для изучения тех или иных аспектов реального мира, исследователи в области машинного обучения используют их для подготовки программ-агентов. В этом смысле проект AlphaGo является частью стратегии компании DeepMind по использованию игр в качестве симулированных микрокосмов реальности. Это помогает нам развивать область искусственного интеллекта и тренировать обучающихся агентов, чтобы в будущем создавать интеллектуальные системы, способные решать самые сложные мировые проблемы.

Работа алгоритма AlphaGo напоминает два режима мышления, которые нобелевский лауреат Даниэль Канеман описал в своей книге «Думай медленно, решай быстро», посвященной вопросам человеческого познания. В случае с AlphaGo ана-логом медленного режима мышления является алгоритм планирования, назы-ваемый *поиском по дереву методом Монте-Карло*, который просчитывает последо-вательность игровых состояний, начиная с заданной позиции, путем расширения дерева игры, включающего возможные будущие ходы и действия противника. Однако при наличии около 10^{170} (1 со 170 нулями) возможных игровых позиций просчет всех последовательностей оказывается невозможным. Чтобы обойти эту проблему и сократить пространство поиска, мы объединили алгоритм поиска по дереву методом Монте-Карло с компонентом *глубокого обучения*, состоящего из двух нейронных сетей, способных оценивать вероятность победы каждого из игроков и выбирать наиболее перспективные ходы.

В более поздней версии алгоритма, AlphaZero, используются принципы *обуче-ния с подкреплением*, что позволяет программе играть против самой себя, не по-лагаясь на записи партий, сыгранных человеком. Этот алгоритм с нуля обучился игре в го (а также в шахматы и сёги), часто обнаруживая (и в дальнейшем отбрасы-вая) многие из стратегий, разработанных за сотни лет игроками-людьми, и создав множество собственных уникальных стратегий.

Авторы этой книги Макс Памперла и Кевин Фергюсон станут вашими провод-никами в увлекательном путешествии от AlphaGo до более поздних версий этого алгоритма. В ходе чтения данного руководства вы не только реализуете движок для игры в го в стиле AlphaGo, но и получите отличное практическое понимание некоторых из основополагающих строительных блоков современных алгорит-мов ИИ: поиска по дереву методом Монте-Карло, глубокого обучения и обуче-

ния с подкреплением. Авторы искусно объединили эти темы, используя игру го в качестве захватывающего и доступного для понимания примера. В дополнение к этому вы изучите основы одной из самых красивых и сложных игр, когда-либо изобретенных человечеством.

Кроме того, эта книга призывает вас с самого начала приступить к созданию работающего го-бота, который будет постепенно эволюционировать от алгоритма, выбирающего ходы совершенно случайным образом, до сложного самообучающегося ИИ для игры в го. Помимо исчерпывающих объяснений основополагающих концепций, авторы предоставили исполняемый код на языке Python. Кроме того, они не пренебрегли такими темами, как форматы данных, развертывание бота и облачные вычисления, необходимыми для обеспечения работоспособности программы для игры в го.

Таким образом, книга «Глубокое обучение и игра в го» представляет собой легко читаемое и увлекательное введение в тему искусственного интеллекта и машинного обучения. Объединяя в себе некоторые из самых захватывающих этапов развития области искусственного интеллекта, она превращается в интереснейший вводный курс по данному предмету. Любой читатель, прошедший этот путь от начала до конца, приобретет необходимые знания для понимания и создания современных систем ИИ, применяемых в ситуациях, требующих сочетания «быстрого» сопоставления образов с «медленным» планированием, которые являются аналогом двух режимов мышления, необходимых для осуществления базового процесса познания.

– *Тор Грпель*, научный сотрудник компании DeepMind,
от имени команды AlphaGo компании DeepMind

Введение

Когда в начале 2016 года о программе AlphaGo заговорили в новостях, мы были чрезвычайно взволнованы этим новаторским достижением в сфере компьютерных игр. В то время считалось, что до создания искусственного интеллекта для игры в го, способного играть на человеческом уровне, оставалось не менее 10 лет. Мы тщательно следили за играми и были готовы пожертвовать сном ради того, чтобы посмотреть трансляции матчей в прямом эфире. Но мы были в хорошей компании – миллионы людей по всему миру были зачарованы играми AlphaGo против Фань Хуэя, Ли Седоля, Ки Цжи и других профессиональных игроков.

Вскоре после появления этого алгоритма мы приступили к работе над небольшой библиотекой с открытым исходным кодом, которую назвали BetaGo (github.com/maxpumperla/betago), чтобы посмотреть, сможем ли мы самостоятельно реализовать некоторые базовые механизмы, лежащие в основе алгоритма AlphaGo. Идея BetaGo состояла в том, чтобы продемонстрировать интересующимся разработчикам некоторые из методов, использующихся в этой программе. Несмотря на то что наших ресурсов (времени, вычислительной мощности или интеллекта) было недостаточно, чтобы конкурировать с невероятным достижением компании DeepMind, мы получили огромное удовольствие в процессе создания собственного го-бота.

С тех пор нам много раз предоставлялась возможность рассказать об ИИ для игры в го. Поскольку мы являемся не только поклонниками этой игры, но и практиками машинного обучения, мы иногда забывали о том, как мало широкая публика могла вынести из событий, за которыми мы так пристально следили. Ирония заключалась в том, что наблюдающие за матчами миллионы людей, по-видимому, делились на две группы:

- те, кто понимает и любит игру го, но мало знает о машинном обучении;
- те, кто понимает и ценит машинное обучение, но практически незнаком с правилами игры в го.

Для далекого от этих тем человека обе сферы могут казаться одинаково туманными, сложными и трудными для освоения. Несмотря на то что в последние годы все большее количество разработчиков программного обеспечения задействует методы машинного обучения и, в частности, *глубокого обучения*, игра в го остается в значительной степени неизвестной многим жителям стран Запада. Мы считаем это весьма прискорбным и искренне надеемся, что данная книга позволит сблизить две вышеупомянутые группы людей.

Мы убеждены в том, что использованию принципов, лежащих в основе алгоритма AlphaGo, можно на практике обучить широкую аудиторию разработчиков программного обеспечения. Наслаждение игрой го и ее понимание приходит в процессе игры и экспериментов. То же самое можно сказать о машинном обучении и любой другой дисциплине.

Если в ходе изучения этой книги вы проникнетесь энтузиазмом в отношении игры го или машинного обучения (надеемся, и того, и другого!), мы будем считать, что выполнили свою задачу. Если, помимо этого, вы научитесь создавать и разворачивать боты для игры в го, а также проводить собственные эксперименты, вам станет доступно множество других интересных ИИ-приложений. Наслаждайтесь путешествием!

Благодарности

Мы хотели бы поблагодарить всех сотрудников издательства Manning, сделавших публикацию этой книги возможной. В частности, мы благодарим наших неутомимых редакторов: Марину Майклз за то, что она помогла нам преодолеть первые 80 % пути, и Дженни Стаут за помощь в преодолении вторых 80 %. Выражаем благодарность нашему техническому редактору Чарльзу Федукке и техническому корректору Тане Вилке за проверку кода.

Благодарим всех рецензентов, предоставивших ценные отзывы: Александра Ерофеева, Алессандро Пузиелли, Алекса Орланди, Бурка Хуфнагеля, Крейга С. Коннелла, Даниэля Береца, Дениса Крайса, Доминго Салазара, Хельмута Хаушильда, Джеймса А. Худа, Джасбу Симпсона, Джин Лазароу, Мартина Мёллера, Скарбиникса Педерсена, Матиаса Поллигкайта, Ната Луенгнарюмитчая, Пьерлуиджи Рити, Сэма Де Костера, Шона Линдсея, Тайлера Коваллиса и Урсины Стосса.

Также спасибо всем, кто экспериментировал или участвовал в разработке нашего проекта BetaGo, особенно Эллиоту Герчаку и Кристоферу Мэлону.

Наконец, благодарим всех, кто когда-либо пытался научить компьютер играть в го и поделился результатами своих исследований.

Я хотел бы поблагодарить Карли за ее терпение и поддержку, а также папу и Джиллиан за то, что научили меня писать.

– Кевин Фергюсон

Особая благодарность Кевину за помощь в разъяснении материала, Андреасу за множество плодотворных дискуссий и Энн за ее постоянную поддержку.

– Макс Памперла

Об авторах

Макс Памперла является специалистом по работе с данными и инженером, занимающимся глубоким обучением в компании-разработчике ИИ-систем Skymind. Также он является сооснователем платформы глубокого обучения **aetros.com**.

Кевин Фергюсон на протяжении 18 лет работал в области создания распределенных систем и анализа данных. Он является специалистом по анализу данных в компании Honor и имеет опыт работы в таких компаниях, как Google и Meebo. Вместе Макс и Кевин разработали BetaGo, один из очень немногих го-ботов с открытым исходным кодом, созданных на языке Python.

Об иллюстрации на обложке

На обложке книги «Глубокое обучение и игра в го» изображен император Монтоку, правивший Японией с 850 по 858 год. Этот портрет был написан акварелью на шелке неизвестным художником. В 2006 году его репродукция была включена в раздел «Императоры и императрицы прошлого» японского исторического журнала *Bessatsu Rekishi Dokuhon*.

Подобные изображения напоминают нам об уникальности и индивидуальности древних городов и регионов мира. В то время по одежде можно было однозначно сказать, к какому из двух городов, разделенных несколькими десятками километров, принадлежит тот или иной человек.

С тех пор дресс-код изменился, и от былой самобытности разных уголков мира не осталось и следа. Сейчас жителей разных континентов бывает трудно отличить друг от друга. Возможно, мы променяли культурное и визуальное разнообразие на более разнообразную и интересную личную и интеллектуальную жизнь, или на более разнообразную жизнь в плане технологий. Мы, сотрудники издательства Manning, стремимся дополнить изобретательность, новаторство и увлеченность, пробуждаемые книгами по компьютерной тематике, обложками, отражающими богатый и разнообразный местный колорит прошлого.

Об этой книге

Книга «Глубокое обучение и игра в го» призвана познакомить читателя с современными концепциями машинного обучения на практическом примере создания искусственного интеллекта для игры в го. К концу главы 3 у вас будет готовая, хоть и очень примитивная программа, играющая в эту игру. В каждой следующей главе будет представлен новый метод улучшения ИИ вашего бота. В ходе экспериментов вы узнаете о преимуществах и недостатках каждого из этих методов. А в последних главах мы покажем, как алгоритмы AlphaGo и AlphaGo Zero позволяют интегрировать все представленные в книге методы в невероятно мощный ИИ.

Для кого предназначена эта книга

Эта книга рассчитана на разработчиков программного обеспечения, желающих приступить к экспериментам в области машинного обучения и предпочитающих практический подход математическому. Мы предполагаем, что вы обладаете практическим опытом программирования на языке Python, однако описанные алгоритмы можно реализовать на любом современном языке. Мы не предполагаем, что вы знакомы с игрой го. Если вы предпочитаете шахматы или какую-либо другую игру подобного рода, то сможете адаптировать к ней большинство описанных методов. Если же вы сами являетесь игроком в го, то получите массу удовольствия, наблюдая за прогрессом вашего бота!

Структура книги

Книга состоит из трех частей, включающих 14 глав и 5 приложений.

Часть I «Основы» знакомит читателя с основными концепциями, подробно описанными в остальных частях книги.

- Глава 1 «На пути к глубокому обучению» предоставляет беглый высокоуровневый обзор таких дисциплин, как искусственный интеллект, машинное обучение и глубокое обучение. В этой главе мы объясняем, как они связаны между собой и что вы можете и не можете сделать с помощью методов, применяемых в этих областях.
- Глава 2 «Игра го как проблема машинного обучения» знакомит читателя с правилами игры го и объясняет, чему мы намерены обучить компьютер.
- В главе 3 «Реализация первого бота для игры в го» мы реализуем доску для игры в го, раскладываем камни и играем в игры с помощью средств языка Python. В конце этой главы вы сможете создать простейший ИИ для игры в го.

Часть II «Машинное обучение и игровой ИИ» посвящена техническим и методологическим основам создания мощного ИИ. В частности, там описаны три метода, которые очень эффективно используются в алгоритме AlphaGo: *поиск по дереву, нейронные сети и обучение с подкреплением*.

Поиск по дереву

- Глава 4 «Игры и поиск по дереву» содержит обзор алгоритмов, которые отвечают за поиск и оценку последовательностей игрового процесса. Мы начнем с простого поиска путем минимаксного перебора, а затем перейдем к таким сложным алгоритмам, как альфа-бета-отсечение и метод Монте-Карло.

Нейронные сети

- Глава 5 «Знакомство с нейронными сетями» представляет собой практическое введение в тему искусственных нейронных сетей. Вы научитесь предсказывать рукописные цифры, создав нейронную сеть с нуля средствами Python.
- Глава 6 «Проектирование нейронной сети для данных го» объясняет сходство данных го с данными изображения, а также представляет сверточные нейронные сети для прогнозирования ходов. В этой главе мы начнем использовать для построения моделей популярную библиотеку глубокого обучения Keras.
- В главе 7 «Глубокое обучение бота на основе данных» мы применим знания, полученные в предыдущих двух главах, чтобы создать бота для игры в го на основе глубоких нейронных сетей. Мы обучим этого бота на фактических игровых данных сильных любительских партий и поговорим об ограничениях данного подхода.
- В главе 8 «Использование ботов» вы узнаете о том, как обеспечить игру ботов с противниками-людьми через пользовательский интерфейс. Кроме того, вы узнаете, как организовать игру ботов с другими ботами локально и на сервере для игры в го.

Обучение с подкреплением

- Глава 9 «Обучение на практике: обучение с подкреплением» посвящена основам обучения с подкреплением и описанию способов его использования для игры в го с самим собой.
- Глава 10 «Обучение с подкреплением и градиенты политики» знакомит читателя с градиентами политики, жизненно важным методом для повышения эффективности прогнозирования ходов, о котором мы говорили в главе 7.
- Глава 11 «Обучение с подкреплением и методы на основе значений» демонстрирует процесс оценки состояний доски с помощью так называемых методов на основе значений, которые являются мощным инструментом в сочетании с поиском по дереву, описанным в главе 4.
- Глава 12 «Обучение с подкреплением и методы типа “актор–критик”» знакомит читателя с методами, позволяющими предсказать долгосрочное значение конкретной позиции на доске и конкретного следующего хода для более эффективного выбора ходов.

В последней части III «Больше, чем сумма всех частей» мы соберем все описанные ранее строительные блоки, чтобы создать приложение, принципом работы напоминающее программу AlphaGo.

- Глава 13 «AlphaGo: Собираем все вместе», по сути, является технической и математической кульминацией этой книги. В ней мы обсуждаем, как

тренировка нейронной сети на данных го (главы 5–7) и последующая игра с самим собой (главы 8–11) в сочетании с поиском по дереву (глава 4) позволяют создать бота для игры в го сверхчеловеческого уровня.

- Глава 14 «AlphaGo Zero: Интеграция поиска по дереву и обучения с подкреплением» посвящена описанию современного состояния ИИ для настольных игр. В ней мы подробно поговорим об инновационной комбинации поиска по дереву и обучения с подкреплением, лежащей в основе программы AlphaGo Zero.

В приложениях мы рассмотрим следующие темы:

- в приложении А «Математические основы» кратко изложены некоторые базовые концепции линейной алгебры и математического анализа и приведены способы представления некоторых структур линейной алгебры с помощью библиотеки Python NumPy;
- в приложении Б «Алгоритм обратного распространения ошибки» более подробно объясняется процедура обучения большинства нейронных сетей, которые мы начнем использовать в главе 5;
- в приложении В «Программы и серверы для игры го» перечислены некоторые ресурсы для читателей, желающих больше узнать о данной игре;
- приложение Г «Обучение и развертывание ботов с помощью Amazon Web Services» представляет собой краткое руководство по запуску вашего бота на облачном сервере Amazon;
- в приложении Д «Отправка бота на онлайн-сервер для игры в го» говорится о том, как подключить бота к популярному серверу для игры в го, где вы можете протестировать его в игре с игроками со всего мира.

Структура книги схематически представлена на рисунке на следующей странице.

О коде

Эта книга содержит много примеров исходного кода как в листингах с пронумерованными строками, так и непосредственно в тексте. В обоих случаях код отформатирован моноширинным шрифтом, что позволяет отличить его от обычного текста. **Полужирным моноширинным шрифтом** выделяются изменения в коде, например новая функция, добавленная в уже существующую строку.

Во многих случаях мы переформатировали исходный код, добавив разрывы строк и изменив отступы с учетом доступного места на странице. В редких случаях даже этого оказалось недостаточно, поэтому мы включили в листинги символы продолжения строки (↵). Кроме того, мы убрали из листинга комментарии, относящиеся к коду, описанному в тексте. Многие листинги сопровождаются аннотациями, содержащими описание важных понятий.

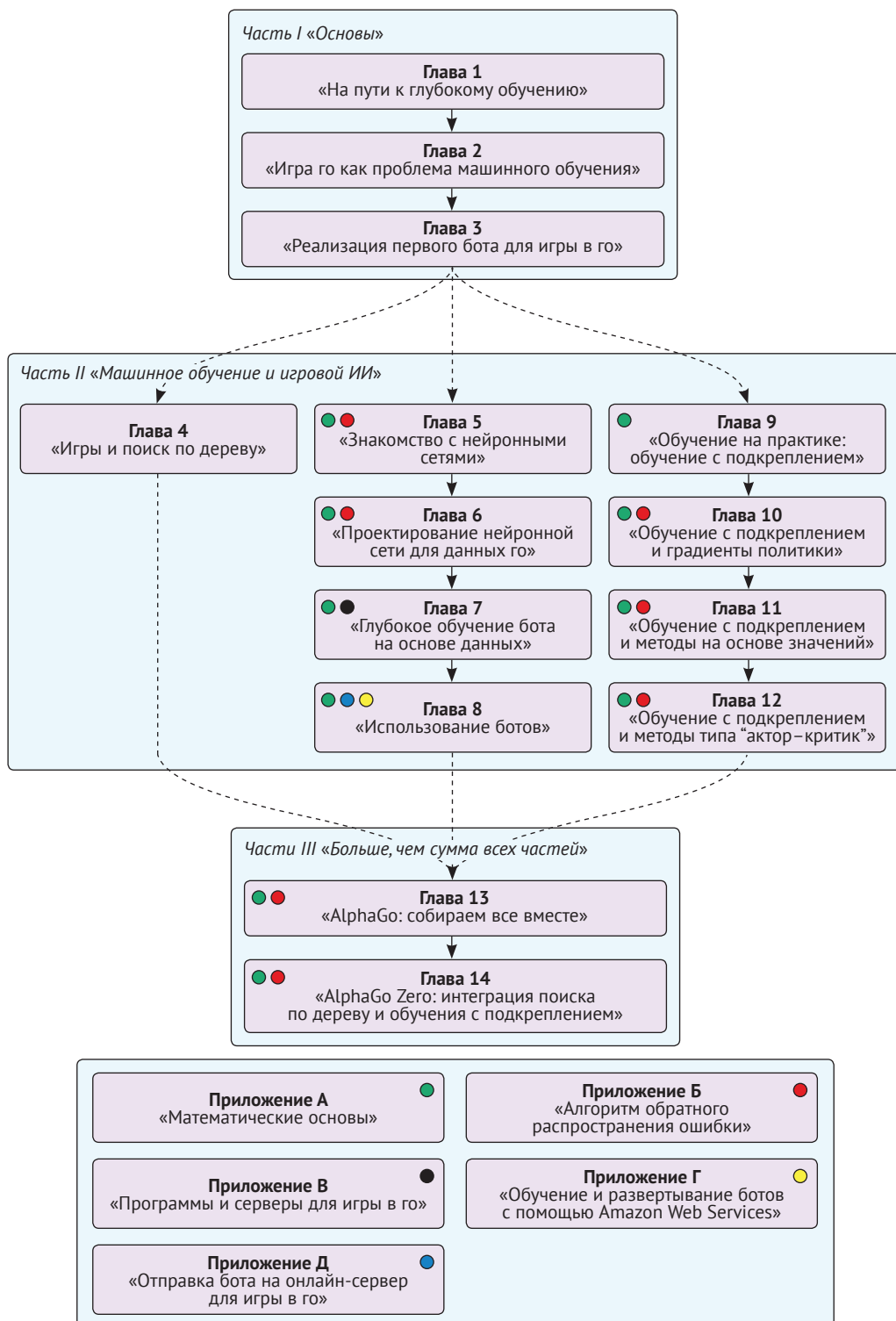
Все примеры кода, а также некоторые дополнительные фрагменты связующего кода вы можете найти на сайте GitHub по адресу: github.com/maxpumperla/deep_learning_and_the_game_of_go.



Такая пиктограмма обозначает совет или рекомендацию.



Такая пиктограмма обозначает указание или примечание общего характера.



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**.

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.

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.

НАРУШЕНИЕ АВТОРСКИХ ПРАВ

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Manning очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции.

Пожалуйста, свяжитесь с нами по адресу электронной почты **dmkpress@gmail.com** со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

Часть I

ОСНОВЫ

Что такое машинное обучение? Что собой представляет игра в го, и почему она является важной вехой на пути развития игрового ИИ? Чем обучение компьютера игре в го отличается от его обучения игре в шахматы или шашки?

В этой части мы ответим на все эти вопросы, после чего вы создадите гибкую библиотеку игровой логики го, которая станет основой для упражнений, приведенных в остальной части книги.

Глава 1

На пути к глубокому обучению: введение в машинное обучение

В этой главе:

- машинное обучение и его отличия от традиционного программирования;
- проблемы, которые можно и нельзя решить с помощью машинного обучения;
- связь машинного обучения и искусственного интеллекта;
- структура системы машинного обучения;
- дисциплины, относящиеся к области машинного обучения.

На протяжении всего срока существования компьютеров программисты интересовались *искусственным интеллектом* (ИИ), который позволил бы реализовать человеческое поведение с помощью компьютера. С давних пор популярным предметом исследований в области ИИ являются игры. В эпоху персональных компьютеров ИИ побеждал людей в игре в шашки, нарды, шахматы и почти во все остальные классические настольные игры. Однако на протяжении многих десятилетий древняя стратегическая игра го оставалась недостижимой для компьютеров. В 2016 году ИИ AlphaGo компании Google DeepMind бросил вызов 14-кратному чемпиону мира Ли Седолю и одержал победу в четырех играх из пяти. Следующая версия AlphaGo оказалась совершенно недосягаемой для игроков-людей: она выиграла 60 игр подряд, победив практически всех известных игроков в го.

Прорыв AlphaGo заключался в расширении классических алгоритмов ИИ с помощью машинного обучения. В частности, программа AlphaGo использовала современные методы *глубокого обучения* – алгоритмы, способные организовывать необработанные данные в полезные уровни абстракции. Применение этих методов не ограничивается играми. Глубокое обучение используется в приложениях для распознавания изображений и речи, системах машинного перевода и управления роботами. Изучение основ глубокого обучения поможет вам разобраться в принципах работы всех этих приложений.

Зачем посвящать целую книгу компьютеру, играющему в го? Авторы не сошли с ума, дело в том, что, в отличие от ИИ для игры в шахматы или нарды, мощный ИИ для игры в го требует применения методов глубокого обучения. Первоклассный шахматный движок вроде Stockfish содержит обширные знания о логике

Конец ознакомительного фрагмента.

Приобрести книгу можно
в интернет-магазине «Электронный универс»
(e-Univers.ru)