

Краткое содержание

Об авторе	16
О редакторах	17
Предисловие к русскоязычному изданию	18
Предисловие	20
Глава 1. Что такое обучение с подкреплением	25
Глава 2. OpenAI Gym	49
Глава 3. Глубокое обучение с помощью PyTorch	70
Глава 4. Метод кросс-энтропии	96
Глава 5. Динамическое программирование и уравнение Беллмана	115
Глава 6. Глубокие Q-сети	133
Глава 7. Расширения для DQN	166
Глава 8. Торговля акциями с использованием обучения с подкреплением	221
Глава 9. Градиенты по стратегиям	243
Глава 10. Метод актора-критика	264
Глава 11. Асинхронный метод актора-критика	281
Глава 12. Тренировка чат-ботов с помощью обучения с подкреплением	298
Глава 13. Веб-навигация	343
Глава 14. Непрерывное пространство действий	385
Глава 15. Доверительные области — TRPO, PPO и ACKTR	412
Глава 16. Оптимизация методом черного ящика в RL	427
Глава 17. Методы, основанные на моделях среды: воображение	449
Глава 18. AlphaGo Zero	471
Заключение	492

Оглавление

Краткое содержание	5
Об авторе	16
О редакторах	17
Предисловие к русскоязычному изданию	18
Предисловие	20
Для кого эта книга	21
Структура издания	21
Извлеките максимум из этой книги	22
Скачивание кода для примеров	23
Скачивание цветных изображений	23
Условные обозначения	23
От издательства	24
Глава 1. Что такое обучение с подкреплением	25
Обучение с учителем, без учителя и с подкреплением	26
Зависимости и отношения в обучении с подкреплением	29
Вознаграждение	30
Агент	31
Среда	32
Действия	32
Наблюдения	33
Марковские процессы принятия решений	36
Марковский процесс	36
Марковский процесс с вознаграждением	41
Марковский процесс принятия решений	44
Резюме	48

Глава 2. OpenAI Gym	49
Структура агента	49
Аппаратные и программные требования	51
OpenAI Gym API	53
Пространство действий	53
Пространство наблюдений	54
Среда	56
Создание среды	57
Сеанс CartPole	59
Агент CartPole, действующий случайным образом	61
Дополнительный функционал Gym — обертки и мониторы	63
Обертки	63
Класс Monitor	65
Резюме	69
Глава 3. Глубокое обучение с помощью PyTorch	70
Тензоры	70
Создание тензоров	71
Скалярные тензоры	73
Операции над тензорами	74
Тензоры с поддержкой графического процессора	74
Градиенты	75
Базовые элементы нейронных сетей	79
Пользовательские слои	81
Последние связующие элементы — функции потерь и оптимизаторы	84
Функции потерь	84
Оптимизаторы	85
Мониторинг с TensorBoard	87
Основы TensorBoard	88
Вывод графиков	89
Пример: GAN на изображениях Atari	90
Резюме	95
Глава 4. Метод кросс-энтропии	96
Классификация методов глубокого обучения	96
Практическое применение метода кросс-энтропии	98

Метод кросс-энтропии в CartPole.....	100
Метод кросс-энтропии в FrozenLake	108
Теоретические основы метода кросс-энтропии	113
Резюме	114
Глава 5. Динамическое программирование и уравнение Беллмана	115
Ценность, состояние и оптимальность	115
Уравнение Беллмана для оптимального управления	117
Ценность действия	120
Метод итерации по ценностям	122
Итерация по ценностям на практике.....	124
Q-обучение для FrozenLake	130
Резюме	132
Глава 6. Глубокие Q-сети	133
Итерация по ценностям в реальности	133
Табличное Q-обучение	135
Глубокое Q-обучение.....	139
Взаимодействие со средой	140
Оптимизация SGD	141
Корреляция между переходами.....	142
Марковское свойство	143
Окончательный вид обучения DQN	143
DQN в Pong.....	144
Обертки	145
Модель DQN.....	150
Обучение	152
Запуск и выполнение	161
Модель в действии.....	162
Резюме	165
Глава 7. Расширения для DQN	166
Библиотека PyTorch Agent Net.....	167
Агент	167
Опыт агента.....	169
Буфер примеров	170
Обертки среды Gym	170

Базовая DQN.....	170
N-шаговые DQN	177
Реализация	180
Двойные DQN	182
Реализация	182
Результаты	185
Зашумленные сети.....	187
Реализация	187
Результаты	191
Приоритизированный буфер примеров	192
Реализация	193
Результаты	198
Дуальная DQN	198
Реализация	200
Результаты	201
Категориальные DQN.....	201
Реализация	204
Результаты	211
Объединение всех методов.....	213
Реализация	214
Результаты	218
Резюме	219
Литература.....	219
Глава 8. Торговля акциями с использованием обучения с подкреплением	221
Торговля.....	221
Данные.....	222
Постановка задачи и ключевые решения	223
Торговая среда	225
Модели	232
Код обучения.....	234
Результаты	234
Полносвязная модель	234
Сверточная модель	238
Дальнейшие эксперименты	241
Резюме	242

Глава 9. Градиенты по стратегиям	243
Ценности и стратегия	243
Почему стратегия?	244
Представление стратегии.....	244
Градиенты по стратегиям.....	245
Метод REINFORCE	246
Пример с CartPole.....	247
Результаты	251
Методы, основанные на стратегиях, в сравнении с методами, основанными на ценностях.....	252
Ограничения метода REINFORCE.....	253
Требование полных эпизодов	253
Высокая дисперсия градиентов	254
Исследование	254
Корреляция между примерами.....	255
Градиенты по стратегиям в CartPole.....	255
Результаты	258
Градиенты по стратегиям в Pong.....	260
Результаты	261
Резюме	263
Глава 10. Метод актора-критика	264
Понижение дисперсии	264
Дисперсия в CartPole.....	266
Актор-критик	268
A2C в Pong.....	271
Результаты A2C в Pong	276
Настройка гиперпараметров	278
Скорость обучения.....	279
β -энтропия.....	279
Количество сред	280
Размер обучающего набора	280
Резюме	280

Глава 11. Асинхронный метод актора-критика	281
Корреляция и эффективность использования данных.....	281
Добавление еще одного A в A2C.....	282
Многопроцессорная обработка в Python.....	285
Параллелизм на уровне данных в A3C	285
Результаты	291
Параллелизм на уровне градиентов в A3C	291
Результаты	296
Резюме	297
Глава 12. Тренировка чат-ботов с помощью обучения с подкреплением	298
Обзор чат-ботов	298
Основы глубокого NLP	301
Рекуррентные нейронные сети.....	301
Эмбединги	303
Кодировщик-декодировщик	304
Обучение seq2seq	305
Обучение с использованием максимального правдоподобия	305
Оценка Bilingual evaluation understudy	308
RL в seq2seq	308
Самокритичное обучение на последовательностях.....	310
Пример чат-бота	311
Структура примера	311
Модули cornell.py и data.py	312
Оценка BLEU и utils.py.....	314
Модель	314
Обучение: перекрестная энтропия	321
Выполнение обучения.....	325
Проверка данных	327
Тестирование обученной модели	328
Обучение: SCST	330
Обучение SCST	337
Результаты	338
Бот для Telegram.....	339
Резюме	342

Глава 13. Веб-навигация	343
Навигация в Интернете.....	343
Автоматизация браузеров и RL	344
Бенчмарк Mini World of Bits.....	345
Universe от OpenAI.....	347
Установка	348
Действия и наблюдения.....	348
Создание рабочей среды.....	349
Стабильность MiniWoB	352
Метод «одного клика»	352
Действия с сеткой.....	352
Разбор примера	354
Модель	355
Код обучения.....	355
Запуск контейнеров	360
Процесс обучения.....	362
Проверка полученной в результате обучения стратегии	364
Проблемы метода одного щелчка	365
Демонстрационные примеры, выполненные человеком	367
Запись демонстрационных примеров	368
Формат записи.....	370
Обучение с использованием демонстрационных примеров.....	373
Результаты	374
Игра в крестики-нолики	375
Добавление текстового описания	377
Результаты	382
Стоит попробовать	383
Резюме	384
Глава 14. Непрерывное пространство действий	385
Почему непрерывное пространство?.....	385
Пространство действий.....	386
Среды.....	386
Метод актора-критика (A2C)	389
Реализация	390
Результаты	393

Использование моделей и видеозаписей.....	394
Градиенты по детерминированным стратегиям.....	395
Исследование	397
Реализация	397
Результаты	402
Запись видео	403
Дистрибутивные градиенты по стратегиям	403
Архитектура.....	404
Реализация	405
Результаты	409
Стоит попробовать	410
Резюме	411
Глава 15. Доверительные области — TRPO, PPO и ACKTR	412
Введение	412
Roboschool	413
Производительность A2C	413
Результаты	415
Запись видео	416
Проксимальная оптимизация стратегии	416
Реализация	417
Результаты	421
Оптимизация стратегии по доверительной области	422
Реализация	422
Результаты	423
A2C с использованием ACKTR	424
Реализация	425
Результаты	425
Резюме	426
Глава 16. Оптимизация методом черного ящика в RL	427
Методы черного ящика	427
Эволюционные стратегии	428
Эволюционные стратегии в CartPole.....	429
Результаты	433

Эволюционные стратегии в HalfCheetah	434
Результаты	439
Генетические алгоритмы	440
Генетические алгоритмы в CartPole.....	441
Результаты	443
Модификации генетических алгоритмов	444
Глубокий ГА.....	444
Поиск новизны.....	444
Генетический алгоритм в Cheetah	445
Результаты	447
Резюме	448
Литература.....	448
Глава 17. Методы, основанные на моделях среды: воображение	449
Сравнение безмодельных методов и методов, основанных на моделях.....	449
Недостатки моделей	451
Агент, дополненный воображением	452
Модель среды	454
Стратегия развертывания	454
Кодировщик развертываний.....	455
Результаты статьи.....	455
I2A в Breakout из Atari.....	455
Базовый агент A2C.....	456
Обучение EM.....	457
Агент с воображением	460
Результаты эксперимента	465
Базовый агент.....	465
Обучение весов EM	467
Обучение с моделью I2A.....	468
Резюме	470
Литература.....	470
Глава 18. AlphaGo Zero	471
Настольные игры	471
Метод AlphaGo Zero	472
Обзор	472

Поиск по дереву Монте-Карло.....	474
Самостоятельная игра.....	475
Обучение и оценка	476
Бот для Connect4	477
Модель игры	478
Реализация MCTS.....	480
Модель	485
Обучение.....	487
Тестирование и сравнение.....	488
Результаты для Connect4	488
Резюме	491
Литература.....	491
Заключение	492

1

Что такое обучение с подкреплением

Обучение с подкреплением (Reinforcement Learning, RL) — способ машинного обучения (Machine Learning, ML), при котором выполняется автоматическое обучение процессу принятия решений во времени. Эта задача получила широкое распространение во многих научных и инженерных областях.

В нашем изменчивом мире даже задачи, кажущиеся стационарными, со временем приобретают динамический характер. Рассмотрим в качестве примера классическую задачу обучения с учителем, когда нужно классифицировать изображения домашних животных по двум категориям: собаки и кошки. У вас есть тренировочный набор данных и классификатор, реализованный с помощью вашего любимого инструментария для глубокого обучения. Спустя некоторое время модель сошла и дает превосходные результаты. Хорошо? Разумеется, да! Вы развернули ее на боевых серверах и оставили работать. Затем, после отдыха на морском побережье, вы вдруг обнаружили, что сменилась мода на стрижки собак, и значительная часть ваших запросов теперь классифицируется ошибочно. Поэтому вам нужно обновить тренировочные изображения и повторить весь процесс заново. Хорошо? Разумеется, нет!

Этот пример демонстрирует тот факт, что даже у простых задач машинного обучения есть скрытое измерение времени, которое обычно не учитывается, но может стать причиной проблем в промышленных системах.

Обучение с подкреплением — подход, который изначально включает это дополнительное измерение (чаще всего это время) в процесс обучения, что делает его значительно ближе к человеческому восприятию искусственного интеллекта. В данной главе вы узнаете:

- ❑ о связи и различиях между обучением с подкреплением и другими областями машинного обучения (обучением с учителем и без учителя);
- ❑ об основных формализмах и моделях обучения с подкреплением и их взаимосвязях;
- ❑ теоретические основы обучения с подкреплением — разберем марковские процессы принятия решений.

Обучение с учителем, без учителя и с подкреплением

Возможно, вам знакомо понятие обучения с учителем. Это наиболее изученная и широко известная задача машинного обучения. Основная идея этого метода заключается в том, чтобы автоматически построить функцию, сопоставляющую входным данным выходные данные при заданном наборе примеров. В такой формулировке эта задача кажется простой, но с ней связано множество сложных частных случаев, с которыми компьютеры только недавно стали более или менее справляться.

Существует множество задач обучения с учителем, включая следующие.

- ❑ **Классификация текстов.** Является ли полученное по электронной почте сообщение спамом?
- ❑ **Классификация изображений и определение местоположения объектов.** На этой картинке изображена кошка, собака или кто-нибудь еще?
- ❑ **Предсказания.** Располагая историей наблюдений с разных датчиков, можно ли сделать прогноз погоды на завтра?
- ❑ **Анализ эмоций.** Как определить степень удовлетворенности клиента по тексту отзыва?

Вопросы могут показаться разноплановыми, но их объединяет одна идея: есть множество примеров входных данных и желаемых результатов и нужно научиться генерировать выходные данные по неизвестным в настоящий момент входным данным. Из самого термина «обучение с учителем» следует, что обучение строится на известных данных, которые были получены от эксперта, предоставившего правильные ответы.

С другой стороны, существует так называемое обучение без учителя, которое предполагает, что поскольку эксперта нет, то нет и известных меток. Следовательно, необходимо изучить скрытую структуру предоставленного набора данных. Одним из широко известных примеров такого подхода к обучению является кластеризация. В этом случае алгоритм пытается объединить данные в набор кластеров в соответствии с некоторыми зависимостями между отдельными примерами.

Еще один метод обучения без учителя, который набирает все большую популярность, — это *генеративно-состязательные сети* (Generative Adversarial Networks, GAN). Основная идея заключается в том, что у нас есть две соревнующиеся сети, первая из которых пытается сгенерировать поддельные данные с целью ввести в заблуждение вторую, в то время как вторая старается отличить искусственно сгенерированные данные от настоящих. С течением времени обе сети становятся все более и более искусными в выполнении своих заданий за счет того, что улавливают неочевидные характерные структуры в наборе данных.

Обучение с подкреплением является третьим подходом и находится где-то между полным контролем и совершенным отсутствием предопределенных меток. С одной стороны, в нем используются многие устоявшиеся методы обучения с учителем, такие как глубокие нейронные сети для аппроксимации функций, сто-

хастический градиентный спуск и метод обратного распространения для обучения представлению данных. С другой стороны, они чаще всего применяются несколько иным образом, чем в обучении с учителем.

В этой главе мы рассмотрим особенности обучения с подкреплением, включая формализмы и абстракции в более или менее четком виде. А пока, чтобы сравнить обучение с подкреплением, обучение с учителем и без учителя, обратимся к более наглядному примеру. Предположим, что у вас есть агент, которому нужно предпринимать действия, находясь в определенной среде. Робомышь в лабиринте на рис. 1.1 послужит хорошим примером, но вы также можете представить вертолет с автопилотом или программу для игры в шахматы. Для простоты остановимся на робомыши.



Рис. 1.1. Мир робомыши в лабиринте

Ее средой является лабиринт, в одних точках которого можно найти еду, а в других — получить удар электрическим током. Робомышь может совершать такие действия, как поворот налево или направо и движение вперед. Она может наблюдать полное состояние лабиринта для принятия решения о дальнейших действиях. Цель робота состоит в том, чтобы найти как можно больше еды, по возможности избегая ударов электрическим током. Эти сигналы о еде и электрическом токе являются вознаграждением, полученным агентом от среды, для дополнительной оценки его действий. Вознаграждение является весьма важной концепцией в обучении с подкреплением, и мы будем обсуждать его далее. На текущий момент достаточно понимать, что конечная цель агента заключается в том, чтобы получить как можно большее суммарное вознаграждение. В данном случае мышь может немного пострадать от удара электрическим током, чтобы добраться до еды, что будет лучше, чем если она будет просто стоять и ничего не получит.

В то же время нам нужно избегать жесткого прописывания в памяти робота информации о среде и лучших действиях в конкретной ситуации, так как это слишком трудоемко и может стать бесполезным даже при незначительном изменении

лабиринта. Чего бы нам хотелось, так это иметь некий магический набор методов, который позволит нашему роботу самостоятельно обучаться тому, как избегать ударов электрошоком и собирать как можно больше еды.

Обучение с подкреплением как раз и есть тот самый магический набор инструментов, который действует иным образом, чем методы обучения с учителем и без учителя. Он не работает с заранее определенными метками, как это делает обучение с учителем. Никто не помечает картинки, которые видит робот, как *плохие* или *хорошие*, и никто не задает для него наилучшее направление.

Тем не менее мы не действуем полностью вслепую, как на стадии оптимизации при обучении без учителя, — у нас есть система вознаграждений. Вознаграждения могут быть положительными при нахождении еды, отрицательными при получении ударов электрическим током и нейтральными, если ничего не происходит. Наблюдая подобные вознаграждения и связывая их с предпринятыми действиями, наш агент учится выполнять действия лучше, собирать большее количество еды и реже получать удары электрическим током.

Конечно же, за такую универсальность и гибкость обучения с подкреплением приходится платить. RL считается куда более сложным, чем обучение с учителем или без учителя. Вкратце рассмотрим, в чем заключается его сложность.

Первое, что следует отметить, — наблюдение в RL зависит от поведения агента и в некоторой мере является его *результатом*. Если ваш агент решает действовать неэффективно, то из наблюдений вы ничего не поймете о том, что было сделано неправильно и как следует поступить, чтобы улучшить результат (агент просто будет постоянно получать отрицательные вознаграждения). Если агент с завидным упрямством продолжает идти по неверному пути, то наблюдения могут дать ложное представление о том, что способа получить большее вознаграждение не существует (жизнь — это страдания), а это может оказаться абсолютно неверным. В терминах машинного обучения это может быть перефразировано как наличие не-*i.i.d.*-данных. Аббревиатура *i.i.d* расшифровывается как *independent and identically distributed* («независимые и одинаково распределенные») — очень важное требование для большинства методов обучения с учителем.

Другие трудности в жизни нашего агента связаны с тем, что ему нужно не только использовать стратегию (политику), которой он обучился, но и активно исследовать среду, ведь, кто знает, может быть, если мы будем действовать по-другому, то сможем значительно улучшить полученный результат. Проблема заключается в том, что если исследований среды будет слишком много, то наше вознаграждение может значительно уменьшиться (не говоря уж о том, что агент может вообще забыть, чему он научился ранее). То есть нам нужно найти некоторый баланс между двумя этими видами деятельности. Проблема выбора между использованием стратегии и исследованием среды — одна из фундаментальных проблем обучения с подкреплением.

Люди постоянно сталкиваются с подобным выбором: пойти поужинать в уже известное место или заглянуть в новый модный ресторан? Как часто нужно менять работу? Что лучше: заняться изучением новой области или продолжить работу в прежней? На эти вопросы нет универсальных ответов.

Третьим усложняющим фактором является то, что вознаграждение и действия могут значительно отстоять друг от друга. В шахматах это может быть сильный ход

в середине игры, решивший ход партии. Во время обучения нам нужно выявлять подобные ситуации и делать выводы, что может оказаться трудновыполнимым.

Тем не менее, несмотря на все эти препятствия и сложности, интерес к обучению с подкреплением растет в области как теории, так и практического применения.

Итак, если вам это интересно, двинемся дальше и рассмотрим основные абстракции обучения с подкреплением, при этом немного углубимся в детали.

ЗАВИСИМОСТИ И ОТНОШЕНИЯ В ОБУЧЕНИИ С ПОДКРЕПЛЕНИЕМ

В каждой научной и инженерной сфере делаются свои предположения и вводятся ограничения. В предыдущем разделе мы рассмотрели обучение с учителем, в котором подобным предположением является знание пар входных и выходных данных (меток). У ваших данных нет меток? Простите, но вам необходимо придумать, как их получить, или использовать какой-нибудь другой метод обучения. Это не говорит о том, что обучение с учителем плохое или хорошее, это просто делает его неприменимым к вашей задаче. Важно знать и понимать правила игры для каждого метода, тогда можно сэкономить много времени. Да, известно множество теоретических и практических прорывов, совершенных, когда кто-то пытался бросить вызов правилам, подойдя к делу творчески. Но все-таки сначала вам нужно разобраться в этих ограничениях.

Конечно же, подобные формализмы существуют и для обучения с подкреплением, и сейчас лучшее время познакомиться с ними, раз уж оставшаяся часть книги посвящена их анализу с различных точек зрения.

На рис. 1.2 вы можете видеть две основные составляющие обучения с подкреплением — *агента* и *среду*, а также способы их взаимодействия — *действия*, *вознаграждения* и *наблюдения*.

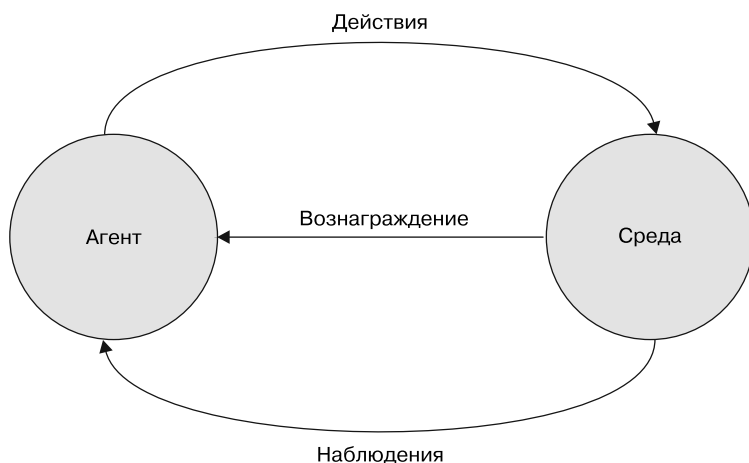


Рис. 1.2. Составляющие обучения с подкреплением и их взаимодействие

Вознаграждение

В первую очередь нужно рассмотреть понятие вознаграждения. В обучении с подкреплением это просто скалярное значение, периодически получаемое нами от среды. Оно может быть положительным или отрицательным, большим или маленьким — это просто число. Цель вознаграждения состоит в том, чтобы сообщить нашему агенту, насколько хорошим было его поведение. Частота, с которой агент получает вознаграждение, никак не задана — это может происходить ежесекундно либо один раз за всю жизнь, тем не менее распространенной практикой является получение вознаграждения через равные промежутки времени или при каждом взаимодействии со средой, просто для удобства. В случае, когда вознаграждение выдается однократно, все награды, за исключением последней, будут нулевыми.

Как уже упоминалось, цель вознаграждения в том, чтобы обеспечить агента обратной связью, информирующей о его успехах, и это важнейший принцип обучения с подкреплением. Сам термин «подкрепление» основан на том, что полученное агентом вознаграждение должно подкреплять его поведение положительным или отрицательным образом. Вознаграждение локально, это означает, что на его получение влияет только недавняя активность агента, а не успехи, достигнутые им за все время. Разумеется, получение значительного вознаграждения вовсе не означает, что секундой позже вы не встретитесь с катастрофическими последствиями ваших предыдущих решений. Это как ограбление банка, которое может показаться весьма неплохой идеей, пока не задумаешься о том, что за этим последует.

Основная цель агента — получить как можно большее вознаграждение за свои действия. Вот несколько конкретных примеров, поясняющих суть вознаграждения.

- ❑ **Торговля на финансовых рынках.** Итоговая прибыль является для участника торгов вознаграждением за покупку и продажу акций.
- ❑ **Шахматы.** В данном случае вознаграждение в конце игры принимает форму победы, проигрыша или ничьей и в значительной степени зависит от конкретной ситуации. Для меня, к примеру, сыграть вничью с гроссмейстером было бы серьезным достижением. На практике следует четко указывать фактическую ценность вознаграждения, притом что оценить ее может быть довольно сложно. Так, в шахматах вознаграждение может быть пропорциональным мастерству противника.
- ❑ **Дофаминовая система в головном мозге.** В мозге есть участок (лимбическая система), который вырабатывает дофамин, если нужно отправить положительный сигнал головному мозгу. Высокие концентрации дофамина вызывают удовольствие, что подкрепляет действия, которые данная система считает хорошими. К несчастью, лимбическая система очень древняя в том смысле, что она расценивает как хорошее еду, размножение и доминирование, поэтому то, что хорошо для выживания с точки зрения лимбической системы, может быть совершенно неприемлемо в социальном плане.