

*Спасибо, папа и мама, что открыли мне свет,
именуемый жизнью, и всегда были рядом.*

Фред, ты крутой чувак.

Ты всегда понуждаешь меня сделать больше.

Спасибо, брат!

Содержание

Об авторе	12
Предисловие	13
Часть I. АЛГОРИТМЫ И ОКРУЖАЮЩИЕ СРЕДЫ	18
Глава 1. Ландшафт обучения с подкреплением	19
Введение в ОП.....	20
Сравнение ОП и обучения с учителем	22
История ОП	23
Глубокое обучение	25
Элементы ОП	26
Стратегия	26
Функция ценности.....	28
Вознаграждение.....	29
Модель.....	30
Применение ОП.....	30
Игры.....	30
Робототехника и индустрия 4.0	31
Машинное обучение.....	32
Экономика и финансы	32
Здравоохранение	32
Интеллектуальные транспортные системы.....	33
Оптимизация энергопотребления и умные сети электроснабжения.....	33
Резюме.....	33
Вопросы.....	33
Для дальнейшего чтения.....	34
Глава 2. Реализация цикла ОП и OpenAI Gym	35
Настройка окружающей среды	36
Установка OpenAI Gym	36
Установка Roboschool	37
OpenAI Gym и цикл ОП.....	37
Разработка цикла ОП.....	38
Привыкаем к пространствам.....	41
Разработка моделей МО с помощью TensorFlow.....	42
Тензоры	43
Создание графа.....	45
Простой пример линейной регрессии	46

Введение в TensorBoard.....	49
Типы окружающих сред ОП.....	51
Зачем нужны различные среды?.....	51
Окружающие среды с открытым исходным кодом.....	52
Резюме.....	54
Вопросы.....	55
Для дальнейшего чтения.....	55

Глава 3. Решение задач методом динамического программирования..... 56

МППР.....	56
Стратегия.....	58
Доход.....	58
Функции ценности.....	59
Уравнение Беллмана.....	60
Классификация алгоритмов ОП.....	61
Безмодельные алгоритмы.....	62
Алгоритмы ОП, основанные на модели.....	63
Разнообразие алгоритмов.....	64
Динамическое программирование.....	64
Оценивание и улучшение стратегии.....	65
Итерация по стратегиям.....	66
Итерация по ценности.....	70
Резюме.....	72
Вопросы.....	73
Для дальнейшего чтения.....	73

Часть II. БЕЗМОДЕЛЬНЫЕ АЛГОРИТМЫ ОП..... 74

Глава 4. Применение Q-обучения и алгоритма SARSA..... 75

Обучение без модели.....	76
Порядок действий.....	76
Оценивание стратегии.....	77
Проблема исследования.....	77
TD-обучение.....	78
TD-обновление.....	79
Улучшение стратегии.....	79
Сравнение методов Монте-Карло и TD-методов.....	79
SARSA.....	80
Алгоритм.....	80
Применение SARSA к игре Taxi-v2.....	81
Q-обучение.....	86
Теория.....	86
Алгоритм.....	87
Применение Q-обучения к игре Taxi-v2.....	87
Сравнение SARSA и Q-обучения.....	89

Резюме.....	91
Вопросы.....	92
Глава 5. Глубокая Q-сеть.....	93
Глубокие нейронные сети и Q-обучение.....	93
Аппроксимация функций.....	94
Q-обучение с нейронными сетями.....	95
Неустойчивость глубокого Q-обучения.....	96
DQN.....	97
Решение.....	97
Алгоритм DQN.....	98
Архитектура модели.....	101
Применение DQN к игре Pong.....	102
Игры Atari.....	102
Предварительная обработка.....	103
Реализация DQN.....	105
Результаты.....	112
Вариации на тему DQN.....	113
Double DQN.....	114
Dueling DQN.....	117
<i>n</i> -шаговый DQN.....	118
Резюме.....	120
Вопросы.....	120
Для дальнейшего чтения.....	121
Глава 6. Стохастическая оптимизация и градиенты	
стратегии.....	122
Методы градиента стратегии.....	122
Градиент стратегии.....	123
Теорема о градиенте стратегии.....	124
Вычисление градиента.....	125
Стратегия.....	126
Алгоритм ГС с единой стратегией.....	127
Устройство алгоритма REINFORCE.....	127
Реализация REINFORCE.....	129
Посадка космического корабля с помощью алгоритма REINFORCE.....	132
REINFORCE с базой.....	134
Реализация REINFORCE с базой.....	136
Обучение алгоритма исполнитель–критик.....	137
Как критик помогает обучаться исполнителю.....	137
<i>n</i> -шаговая модель AC.....	138
Реализация AC.....	139
Посадка космического корабля с помощью алгоритма AC.....	141
Дополнительные улучшения AC и полезные советы.....	142
Резюме.....	143
Вопросы.....	143
Для дальнейшего чтения.....	143

Глава 7. Реализация TRPO и PPO	144
Roboschool	144
Управление непрерывной системой.....	145
Метод естественного градиента стратегии	148
Интуитивное описание NPG	149
Немного математики	150
Осложнения в методе естественного градиента	152
Оптимизация стратегии в доверительной области	152
Алгоритм TRPO	153
Реализация алгоритма TRPO	156
Применение TRPO	160
Проксимальная оптимизация стратегии.....	163
Краткое описание	163
Алгоритм PPO	163
Реализация PPO	164
Применение PPO	166
Резюме.....	168
Вопросы.....	168
Для дальнейшего чтения.....	169
Глава 8. Применения алгоритмов DDPG и TD3	170
Сочетание оптимизации градиента стратегии с Q-обучением	170
Детерминированный градиент стратегии.....	171
Алгоритм DDPG	174
Реализация DDPG	176
Применение DDPG к среде VipedalWalker-v2.....	180
Алгоритм TD3	182
Проблема смещения оценки в сторону завышения.....	182
Уменьшение дисперсии	184
Применение TD3 к среде VipedalWalker-v2.....	186
Резюме.....	187
Вопросы.....	188
Для дальнейшего чтения.....	188
Часть III. ЗА ПРЕДЕЛАМИ БЕЗМОДЕЛЬНЫХ АЛГОРИТМОВ	189
Глава 9. ОП на основе модели	190
Методы на основе модели.....	190
Общая картина обучения на основе модели	191
Достоинства и недостатки	195
Сочетание безмодельного и основанного на модели обучения	196
Полезная комбинация.....	196
Построение модели из изображений.....	198
Применение алгоритма ME-TRPO к задаче об обратном маятнике	199

Принцип работы ME-TRPO	200
Реализация ME-TRPO	200
Эксперименты в среде RoboSchool.....	204
Резюме.....	206
Вопросы.....	207
Для дальнейшего чтения.....	207
Глава 10. Подражательное обучение и алгоритм DAgger	208
Технические требования.....	208
Установка Flappy Bird	209
Подход на основе подражания	209
Пример: помощник водителя.....	210
Сравнение подражательного обучения и обучения с подкреплением.....	211
Роль эксперта в подражательном обучении	211
Структура IL	212
Игра Flappy Bird	214
Порядок взаимодействия с окружающей средой.....	215
Алгоритм агрегирования набора данных	216
Алгоритм DAgger	217
Реализация DAgger	217
Анализ результатов игры в Flappy Bird	221
Обратное обучение с подкреплением.....	222
Резюме.....	223
Вопросы.....	223
Для дальнейшего чтения.....	224
Глава 11. Оптимизация методом черного ящика	225
За рамками ОП.....	225
Краткий обзор ОП.....	226
Альтернатива	226
Основы эволюционных алгоритмов	227
Генетические алгоритмы	230
Эволюционные стратегии.....	230
Масштабируемые эволюционные стратегии.....	232
Основной принцип.....	233
Масштабируемая реализация.....	234
Применение масштабируемой ЭС к среде LunarLander	239
Резюме.....	241
Вопросы.....	241
Для дальнейшего чтения.....	242
Глава 12. Разработка алгоритма ESBAS	243
Исследование и использование.....	244
Задача о многоруком бандите	245
Подходы к исследованию.....	246
ϵ -жадная стратегия	246

Алгоритм UCB	247
Сложность исследования	248
Алгоритм ESBAS.....	249
Что такое выбор алгоритма	249
ESBAS изнутри	250
Реализация.....	252
Тестирование в среде Acrobot.....	255
Резюме.....	257
Вопросы.....	258
Для дальнейшего чтения.....	258
Глава 13. Практические подходы к решению проблем ОП.....	259
Рекомендуемые практики глубокого ОП.....	259
Выбор подходящего алгоритма	260
От простого к сложному.....	261
Проблемы глубокого ОП.....	263
Устойчивость и воспроизводимость результатов	263
Эффективность	264
Обобщаемость.....	265
Передовые методы	266
ОП без учителя.....	266
Перенос обучения.....	268
ОП в реальном мире	270
Лицом к лицу с реальным миром.....	270
Преодоление разрыва между имитационной моделью и реальным миром	271
Создание собственной окружающей среды.....	272
Будущее ОП и его влияние на общество	272
Резюме.....	273
Вопросы.....	274
Для дальнейшего чтения.....	274
Ответы на вопросы.....	275
Предметный указатель	281

Об авторе

Андреа Лонца занимается глубоким обучением, одержим искусственным интеллектом и страстью создавать машины, действующие «разумно». Знания в области обучения с подкреплением, обработки естественного языка и компьютерного зрения приобрел в ходе работы над проектами по машинному обучению в университете и в промышленности. Также участвовал в нескольких конкурсах Kaggle и достигал высоких результатов. Всегда ищет интересные задачи и любит доказывать, на что способен.

О РЕЦЕНЗЕНТЕ

Грег Уолтерс занимается компьютерами и программированием с 1972 года. Отлично владеет языками Visual Basic, Visual Basic .NET, Python и SQL (диалектами MySQL, SQLite, Microsoft SQL Server, Oracle), C++, Delphi, Modula-2, Pascal, C, ассемблером 80x86, COBOL и Fortran. Обучает программированию, через его руки прошло множество людей, которых он учил таким продуктам, как MySQL, Open Database Connectivity, Quattro Pro, Corel Draw!, Paradox, Microsoft Word, Excel, DOS, Windows 3.11, Windows for Workgroups, Windows 95, Windows NT, Windows 2000, Windows XP и Linux. Сейчас на пенсии и в свободное время музицирует и обожает готовить, но всегда готов поработать фрилансером над разными проектами.

Предисловие

Обучение с подкреплением (ОП) – популярное и многообещающее направление искусственного интеллекта, в рамках которого изучается построение моделей и агентов, способных находить идеальное поведение в условиях изменяющихся требований. Эта книга поможет вам овладеть алгоритмами ОП и научиться реализовывать их при создании самообучающихся агентов.

Книга начинается с введения в инструменты, библиотеки и процедуру установки, необходимые для работы в среде ОП, затем рассматриваются различные элементы ОП и применение методов, основанных на понятии ценности, в частности алгоритмов Q-обучения и SARSA. Вы научитесь сочетать Q-обучение с нейронными сетями для решения сложных задач. Рассматриваются также методы градиента стратегии, TRPO и PPO, позволяющие повысить производительность и устойчивость, а затем детерминированные алгоритмы DDPG и TD3. Объясняется, как работает техника подражательного обучения и как алгоритм Dagger позволяет обучить агента летать. Вы узнаете об эволюционных стратегиях и оптимизации методом черного ящика. И наконец познакомитесь с алгоритмами исследования на базе верхней доверительной границы (UCB и UCB1) и метаалгоритмом ESBAS.

Прочитав книгу до конца, вы научитесь применять основные алгоритмы ОП для решения реальных задач и станете членом сообщества ОП.

ПРЕДПОЛАГАЕМАЯ АУДИТОРИЯ

Если вы занимаетесь исследованиями в области ИИ, применяете в работе глубокое обучение или просто хотите изучить ОП с нуля, то эта книга для вас. Она будет полезна и тем, кто хочет узнать о последних достижениях в этой области. Необходимо владение языком Python на рабочем уровне.

СТРУКТУРА КНИГИ

В главе 1 «Ландшафт обучения с подкреплением» описываются проблемы, которые ОП с успехом решает, и приложения, в которых алгоритмы ОП уже нашли применение. Здесь же рассматриваются инструменты, библиотеки и процедуры их установки и настройки, необходимые для выполнения обсуждаемых в книге проектов.

В главе 2 «Реализация цикла ОП и OpenAI Gym» описывается главный цикл алгоритмов ОП, инструментарий для разработки алгоритмов и различные типы сред. Мы разработаем с помощью интерфейса к OpenAI Gym агента, который будет играть в игру CartPole, совершая случайные действия. Мы также научимся использовать OpenAI Gym для работы в других средах.

Глава 3 «Решение задач методом динамического программирования» содержит введение в основные идеи, терминологию и подходы, применяемые в ОП.

Вы узнаете о главных составных частях ОП и составите общее представление о том, как алгоритмы ОП применяются к решению задач. Вы также узнаете о различиях между основанными на модели и безмодельными алгоритмами и о классификации алгоритмов обучения с подкреплением. Мы применим динамическое программирование для решения игры FrozenLake.

В главе 4 «Применения Q-обучения и алгоритма SARSA» речь пойдет о методах на основе ценности, в частности Q-обучении и SARSA, двух алгоритмах, которые отличаются от динамического программирования и хорошо масштабируются на задачи большого размера. Чтобы лучше разобраться в этих алгоритмах, мы применим их к игре FrozenLake и сравним результаты с динамическим программированием.

В главе 5 «Глубокие Q-сети» описывается использование нейронных сетей и, в частности, **сверточных нейронных сетей (СНС)** к Q-обучению. Вы узнаете, почему сочетание Q-обучения и нейронных сетей дает поразительные результаты и как это открывает путь к гораздо более широкому кругу задач. Кроме того, мы применим глубокую Q-сеть к игре Atari, воспользовавшись интерфейсом к OpenAI Gym.

В главе 6 «Стохастическая оптимизация и градиенты стратегии» вводится новое семейство безмодельных алгоритмов: методы градиента стратегии. Мы узнаем о различии между методами градиента стратегии и методами на основе ценности, а также об их сильных и слабых сторонах. Затем реализуем алгоритмы REINFORCE и исполнитель–критик для решения игры LunarLander.

В главе 7 «Реализация TRPO и PPO» предлагается модификация методов градиента стратегии с использованием новых механизмов контроля над улучшением стратегии. Эти механизмы позволяют улучшить устойчивость и сходимость алгоритмов градиента стратегии. В частности, мы опишем и реализуем два основных метода градиента стратегии, в которых используются эти подходы: TRPO и PPO. Они будут реализованы в семействе сред с непрерывным пространством действий RoboSchool.

В главе 8 «Применения алгоритмов DDPG и TD3» вводится новая категория алгоритмов с детерминированной стратегией, которые сочетают идею градиента стратегии с Q-обучением. Вы узнаете об идее и реализации двух глубоких детерминированных алгоритмов, DDPG и TD3, в новой окружающей среде.

В главе 9 «ОП на основе модели» иллюстрируются алгоритмы ОП, которые обучаются модели окружающей среды для планирования будущих действий, т. е. обучения стратегии. Вы узнаете, как они работают, в чем их плюсы и почему они предпочтительны во многих ситуациях. Чтобы лучше разобраться в них, мы реализуем основанный на модели алгоритм в среде RoboSchool.

В главе 10 «Подражательное обучение и алгоритм DAgger» объясняется, как работает подражательное обучение и как его можно адаптировать к конкретной задаче. Будет рассмотрен самый известный алгоритм подражательного обучения, DAgger. Ради лучшего освоения мы реализуем его и тем ускорим процесс обучения агента в игре FlappyBird.

В главе 11 «Оптимизация методом черного ящика» изучаются эволюционные алгоритмы – класс алгоритмов оптимизации методом черного ящика, которые не опираются на обратное распространение. Интерес к этим алгоритмам растет, потому что они, во-первых, быстро обучаются, а во-вторых, легко

распараллеливаются на сотни и тысячи процессорных ядер. В этой главе подводится теоретический фундамент под эти алгоритмы и приводится практическая реализация на примере алгоритма эволюционной стратегии.

В главе 12 «Разработка алгоритма ESBAS» описывается важная дилемма исследования–использования, специфичная для ОП. Она демонстрируется на примере задачи о многоруких бандитах и решается методами верхней доверительной границы: UCB и UCB1. Затем мы узнаем о проблеме выбора алгоритма и разработаем метаалгоритм ESBAS, в котором UCB1 используется для выбора наиболее подходящего алгоритма ОП в конкретной ситуации.

В главе 13 «Практические подходы к решению проблем ОП» мы рассмотрим основные проблемы, возникающие в этой области, и объясним, как их преодолевать. Вы узнаете о некоторых трудностях применения ОП к реальным задачам, о будущем глубокого ОП и о его влиянии на общество.

Что необходимо для чтения этой книги

Необходимо владение языком Python на рабочем уровне. Знакомство с ОП и различными инструментами, используемыми в этой области, также было бы полезно.

Графические выделения

В этой книге для выделения семантически различной информации применяются различные стили. Ниже приведены примеры стилей с пояснениями.

Код в тексте: фрагменты кода, имена таблиц базы данных, папок и файлов, URL-адреса, адреса в Twitter, например: «В этой книге используется версия Python 3.7, но должны работать все версии начиная с 3.5. Предполагается также, что пакеты `numpy` и `matplotlib` уже установлены».

Отдельно стоящие фрагменты кода набраны так:

```
import gym

# создать окружающую среду
env = gym.make("CartPole-v1")
# привести среду в исходное состояние перед началом работы
env.reset()

# повторить 10 раз
for i in range(10):
    # предпринять случайное действие
    env.step(env.action_space.sample())
    # отрисовать игру
    env.render()

# закрыть среду
env.close()
```

Текст, который вводится на консоли или выводится на консоль, напечатан следующим образом:

```
$ git clone https://github.com/pybox2d/pybox2d
$ cd pybox2d
$ pip install -e .
```

Новые термины, важные слова и слова на экране набраны **полужирным шрифтом**. Также выделяются элементы интерфейса, например пункты меню и поля в диалоговых окнах, например: «В **обучении с подкреплением (ОП)** алгоритм называется агентом, он обучается на данных, поступающих от окружающей среды».

 Предупреждения и важные замечания оформлены так.

 Советы и рекомендации выглядят так.

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 http://dmkpress.com/authors/publish_book/ или напишите в издательство: 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 данной 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.

СКАЧИВАНИЕ ИСХОДНОГО КОДА

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

НАРУШЕНИЕ АВТОРСКИХ ПРАВ

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Packt Publishing 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 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

АЛГОРИТМЫ И ОКРУЖАЮЩИЕ СРЕДЫ

Эта часть представляет собой введение в обучение с подкреплением. В ней закладывается теоретический фундамент и подготавливается почва для последующих глав. Эта часть состоит из следующих глав:

- глава 1 «Ландшафт обучения с подкреплением»;
- глава 2 «Реализация цикла ОП и OpenAI Gym»;
- глава 3 «Решение задач методами динамического программирования».

Глава 1

Ландшафт обучения с подкреплением

Люди и животные обучаются методом проб и ошибок. Этот процесс основан на механизмах вознаграждения в ответ на то или иное поведение. Его цель – посредством многократного повторения побудить к выбору таких действий, которые порождают положительные отклики, и отвлечь от действий, порождающих отрицательные отклики. Посредством механизма проб и ошибок мы обучаемся взаимодействовать с людьми и окружающим нас миром, преследовать сложные осмысленные цели, а не просто стремиться к получению немедленного удовольствия.

Обучение на опыте и взаимодействии принципиально важно. Представьте, что вы должны научиться играть в футбол, только глядя на то, как играют другие. Если, основываясь на таком опыте, вы выйдете на поле, то, скорее всего, будете играть отвратительно.

Это было продемонстрировано в середине XX века в известной работе Ричарда Хелда (Richard Held) и Алана Хейна (Alan Hein) 1963 года, в которой два котенка с рождения были посажены в некое подобие карусели. Один котенок мог двигаться свободно (был активен), а второй не мог производить никаких движений и перемещался пассивно – его возил первый. Впоследствии только у котенка, который мог свободно двигаться, развилось восприятие глубины и двигательные навыки. Это было доказано отсутствием у пассивного котенка мигательного рефлекса на приближающиеся объекты. Таким образом, в этом довольно жестоком эксперименте было продемонстрировано, что для обучения животного необходимо физическое взаимодействие с окружающей средой, а не только зрительная стимуляция.

В основе **обучения с подкреплением (ОП)** лежит идея об активном взаимодействии со средой методом проб и ошибок, рассмотренная у обучающихся людей и животных. В частности, в ОП агент постепенно обучается в процессе взаимодействия с окружающим миром. Это позволяет наделять компьютер рудиментарными навыками обучения, взяв за образец поведение человека.

Эта книга целиком посвящена обучению с подкреплением. Ее цель – помочь вам разобраться в данной области на практических примерах. В начальных главах мы познакомимся с базовыми понятиями обучения с подкреплением, а затем приступим к разработке первых алгоритмов. Постепенно мы будем создавать все более сложные и мощные алгоритмы для решения более интересных

и интригующих задач. Вы увидите, что обучение с подкреплением – очень широкий предмет и что в нем существует много алгоритмов, предназначенных для решения разных задач разными способами. Однако мы постараемся дать простое, но полное описание всех идей, а также ясные и практически полезные реализации алгоритмов.

В этой главе мы познакомимся с фундаментальными концепциями ОП, узнаем о различиях между имеющимися подходами и о таких ключевых понятиях, как функция ценности, вознаграждение и модель окружающей среды. Попутно мы расскажем об истории ОП и ее приложениях.

В этой главе рассматриваются следующие вопросы:

- введение в ОП;
- элементы ОП;
- приложения ОП.

ВВЕДЕНИЕ В ОП

ОП – часть машинного обучения, в которой изучается последовательное принятие решений для достижения поставленной цели. Задача ОП состоит из **агента**, принимающего решения, и физического или виртуального мира, с которым агент взаимодействует, – **окружающей среды**. Взаимодействие агента с окружающей средой сводится к **действиям**, имеющим некоторые последствия. В результате агент получает от среды обратную связь в форме нового **состояния** и **вознаграждения**. Оба этих сигнала – последствия действия, предпринятого агентом. Точнее, вознаграждение показывает, насколько хорошим или плохим было действие, а состояние – это текущее представление агента и окружающей среды. Этот цикл показан на рис. 1.1.

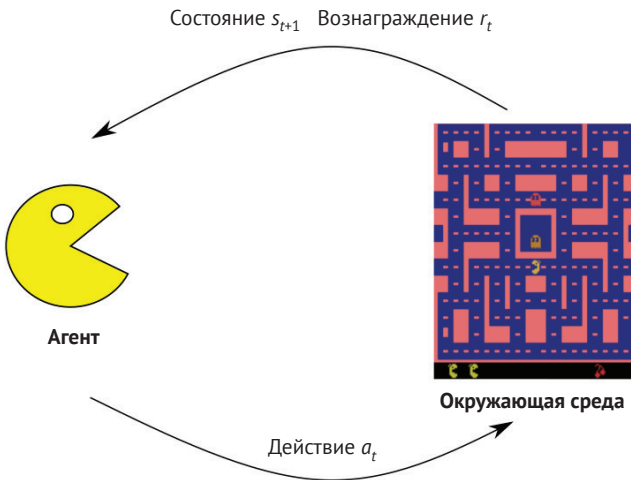


Рис. 1.1

На этом рисунке агент представлен значком пакмана¹, который в зависимости от текущего состояния окружающей среды выбирает следующее действие. Его поведение влияет на окружающую среду, а именно на его положение и положения врагов. В ответ на поведение агента среда возвращает новое состояние и вознаграждение. Этот цикл повторяется, пока игра не завершится.

Конечная цель агента – максимизировать полное вознаграждение, полученное за все время существования. Введем обозначения: если a_t – действие в момент t , а r_t – вознаграждение в момент t , то агент предпринимает действия a_0, a_1, \dots, a_t , стремясь максимизировать сумму $\sum_{i=0}^t r_i$.

Чтобы максимизировать полное вознаграждение, агент должен обучиться наилучшему поведению в каждой ситуации. Для этого агенту необходимо произвести оптимизацию на длинном горизонте, принимая во внимание каждое действие. В окружающей среде с большим количеством дискретных или непрерывных состояний и действий обучение затруднено тем, что агент вынужден учитывать все возможные ситуации. Мало того, вознаграждение может поступать очень редко и запаздывать во времени, что дополнительно усложняет процесс обучения.

Чтобы привести пример задачи ОП и заодно объяснить, в чем сложность разреженного вознаграждения, рассмотрим хорошо известную сказку о Гензеле и Гретель. Родители завели их в лес, чтобы там оставить, но Гензель, знавший об их намерении, захватил с собой из дому ломоть хлеба и оставил дорожку из хлебных крошек, которая должна была привести его и сестру обратно домой. В системе ОП агентами являются Гензель и Гретель, а окружающей средой – лес. За каждую встреченную на обратном пути крошку агент получает вознаграждение +1, а за выход к дому – вознаграждение +10. В этом случае чем чаще хлебная дорожка, тем легче детям будет найти дорогу домой, поскольку для перехода от одной крошки к другой нужно исследовать сравнительно небольшую область. К сожалению, в реальном мире разреженные вознаграждения встречаются куда чаще плотных.

Важная характеристика ОП заключается в возможности достижения результата, даже когда окружающая среда динамическая, неопределенная и недетерминированная. Ниже приведены примеры реальных проблем, которые можно поставить как задачи ОП.

- Беспилотные автомобили – популярная идея, которая с трудом поддается ОП. Дело в том, что при движении по дороге нужно принимать во внимание много факторов (пешеходы, другие автомобили, велосипеды, светофоры и т. д.), а среда в высшей степени неопределенная. В данном случае беспилотный автомобиль – это агент, который может воздействовать на руль, акселератор и тормоза. Средой же является окружающий мир. Очевидно, что агент не может воспринять весь мир вокруг себя целиком, ему доступна лишь ограниченная информация от датчиков (например, камеры, радара и системы навигации). Цель беспилотного автомобиля – добраться до места назначения за минимальное время, соблюдая правила дорожного движения и не нанеся никакого вреда. Сле-

¹ Персонаж старой компьютерной игры PacMan. – Прим. перев.

довательно, агент может получить отрицательное вознаграждение, если случится какое-то неприемлемое событие, а величина положительного пропорциональна времени поездки.

- В шахматах цель состоит в том, чтобы поставить мат противнику. В терминах ОП игрок является агентом, а текущая позиция на доске – окружающей средой. Агенту разрешается переставлять фигуры согласно правилам. В итоге окружающая среда возвращает положительное вознаграждение, если агент выиграл, и отрицательное, если проиграл. Во всех остальных случаях вознаграждение нулевое, а следующим состоянием является позиция на доске после хода противника. В отличие от примера беспилотного автомобиля, здесь состояние агента совпадает с состоянием окружающей среды. Иными словами, агент знает все об окружающей среде.

Сравнение ОП и обучения с учителем

ОП и обучение с учителем – похожие, но все же различные парадигмы обучения на данных. Многие задачи можно решить как с помощью ОП, так и обучения с учителем, но в большинстве случаев эти методы предназначены для решения разных задач.

Цель обучения с учителем – научиться обобщать, располагая лишь фиксированным набором данных с ограниченным количеством примеров. Каждый пример состоит из входа и желательного выхода (или метки), так что реакция на выбор агента следует незамедлительно.

Напротив, в ОП акцент ставится на последовательных действиях, которые можно предпринять в конкретной ситуации. В данном случае единственное, что дает учитель, – сигнал вознаграждения. Какое действие правильно при данных условиях, неизвестно, в отличие от обучения с учителем.

ОП можно рассматривать как более общую и полную систему обучения. Перечислим основные характеристики ОП:

- вознаграждение может быть плотным, разреженным или приходиться с большим запаздыванием. Во многих случаях вознаграждение становится известно только в конце задания (например, в шахматах);
- задача последовательная и зависящая от времени – выбранные действия могут влиять на следующие действия, которые, в свою очередь, влияют на возможные вознаграждения и состояния;
- агент должен совершать действия, которые с высокой вероятностью приведут к достижению цели (использование), но в то же время должен пробовать иные действия, чтобы другие части окружающей среды не остались неисследованными (исследование). Эту двойственность называют дилеммой (или компромиссом) исследования–использования, она призвана решить трудную проблему поиска баланса между исследованием и использованием окружающей среды. Она важна также и потому, что, в отличие от обучения с учителем, агент ОП может влиять на окружающую среду, т. к. вправе собирать новые данные, коль скоро считает это полезным;
- окружающая среда стохастическая и недетерминированная, и агент должен учитывать это в процессе обучения и для предсказания следующего

действия. Мы увидим, что многие компоненты ОП можно спроектировать так, что они будут выдавать либо только одно детерминированное значение, либо диапазон значений, каждому из которых сопоставлена вероятность.

Третий вид обучения – **обучение без учителя** – применяется для выявления закономерностей в данных при отсутствии какой-либо информации от учителя. Сжатие данных, кластеризация и порождающие модели – все это примеры обучения без учителя. Его можно инкорпорировать в систему ОП, чтобы исследовать окружающую среду и получать информацию о ней. Комбинация обучения без учителя и ОП называется **ОП без учителя**. В этом случае никакое вознаграждение не присуждается, а агент может генерировать внутренние мотивы, побуждающие отдать предпочтение новым ситуациям, которые можно исследовать.



Отметим, что задачи, связанные с беспилотными автомобилями, можно решать и как задачи обучения с учителем, но результаты получаются неудовлетворительные. Основная проблема в том, что распределение данных, с которым агент сталкивается на практике, может сильно отличаться от того, что он видел в процессе обучения.

История ОП

Первое математическое обоснование ОП появилось в 1960–1970-е годы в области оптимального управления. Тем самым была решена задача минимизации некоторой меры поведения динамической системы, изменяющейся во времени. Этот метод подразумевал решение системы уравнений при известной динамике системы. Тогда было сформулировано ключевое понятие **марковского процесса принятия решений (МППР)**. Оно легло в основу общего подхода к моделированию принятия решений в стохастических условиях. В эти годы был разработан метод решения задач оптимального управления, получивший название **динамического программирования (ДП)**. Смысл ДП в том, чтобы разбить сложную задачу на несколько более простых с целью решить систему уравнений МППР.

Отметим, что ДП упрощает поиск оптимального управления лишь для систем с известной динамикой, ни о каком обучении речи не идет. Кроме того, ему свойственна проблема **проклятия размерности**, поскольку требования к вычислительным ресурсам экспоненциально возрастают по мере роста количества состояний.

Но, как заметили Ричард Саттон и Эндрю Бартон, хотя эти методы не подразумевают обучения, мы все равно должны считать методы решения задач оптимального управления, в т. ч. ДП, методами ОП.

В 1980-х годах наконец-то появилась концепция обучения на основе предсказаний в соседние моменты времени – метод **обучения на основе временных различий (temporal difference learning – TD-обучение)**. Открытие TD-обучения принесло с собой новое семейство эффективных алгоритмов, которые будут рассмотрены в этой книге.

Первые задачи, решенные с помощью TD-обучения, были настолько малы, что их можно было представить с помощью таблиц или массивов. Соответ-

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru