

*Моим родителям, которые поощряли все мои увлечения,
буквально окружали меня великолепными книгами
и подарили мне первый компьютер в те времена,
когда он все еще оставался предметом роскоши
(и использовался в основном для развлечений)*

Оглавление

Часть I ПЕРВЫЕ ШАГИ	23
1 ■ Когда и зачем используется JAX	25
2 ■ Первая программа в JAX.....	46
Часть II ЯДРО JAX.....	79
3 ■ Работа с массивами.....	81
4 ■ Вычисление градиентов	132
5 ■ Компиляция кода	178
6 ■ Векторизация кода.....	223
7 ■ Распараллеливание вычислений.....	254
8 ■ Использование сегментирования тензоров	305
9 ■ Случайные числа в JAX.....	333
10 ■ Работа с pytree	368
Часть III ЭКОСИСТЕМА	393
11 ■ Высокоуровневые библиотеки поддержки нейронных сетей.....	395
12 ■ Другие члены экосистемы JAX	445

Содержание

Оглавление	6
Предисловие	12
Благодарности	14
О книге	16
Об авторе	21
Об иллюстрации на обложке	22
Часть I ПЕРВЫЕ ШАГИ	23
1 Когда и зачем используется JAX	25
1.1 Почему нужно использовать JAX	30
1.1.1 Производительность вычислений	30
1.1.2 Функциональный подход	34
1.1.3 Экосистема JAX	35
1.2 Чем JAX отличается от NumPy	37
1.2.1 JAX как NumPy	38
1.2.2 Компонуемые трансформации	39
1.3 Чем JAX отличается от TensorFlow и PyTorch	42
Резюме	45
2 Первая программа в JAX	46
2.1 Учебная задача машинного обучения: классификация рукописных цифр	47
2.2 Общий обзор проекта глубокого обучения с использованием JAX	49
2.3 Загрузка и подготовка набора данных	51
2.4 Простая нейронная сеть с использованием JAX	54
2.4.1 Инициализация нейронной сети	56
2.4.2 Нейронная сеть с прямой связью	58

2.5	<code>vmarr</code> : автоматически векторизованные вычисления для обработки пакетов.....	61
2.6	Autodiff: как вычислять градиенты, ничего не зная о производных	65
2.6.1	Функция потерь	67
2.6.2	Определение градиентов	68
2.6.3	Шаг обновления градиента.....	68
2.6.4	Цикл тренировки	70
2.7	JIT: компиляция кода для ускорения его выполнения	72
2.8	Сохранение и развертывание модели	74
2.9	Чистые функции и компонуемые трансформации: почему они так важны	76
	Упражнение 2.1.....	77
	Резюме	78
Часть II	ЯДРО JAX	79
3	Работа с массивами	81
3.1	Обработка изображений с использованием массивов NumPy	82
3.1.1	Загрузка изображения в массив NumPy.....	84
3.1.2	Выполнение простых операций предварительной обработки с изображением	88
3.1.3	Добавление шума в изображение.....	90
3.1.4	Реализация фильтрации изображения	92
3.1.5	Сохранение тензора как файла изображения	98
3.2	Массивы в JAX	99
3.2.1	Переход на NumPy-подобный API JAX.....	100
3.2.2	Что такое <code>Aggaf</code> ?	102
3.2.3	Операции, связанные с устройствами	104
3.2.4	Асинхронная диспетчеризация	111
3.2.5	Выполнение вычислений на TPU.....	112
3.3	Отличия от NumPy	117
3.3.1	Неизменяемость	117
3.3.2	Типы	121
3.4	Интерфейсы высокого и низкого уровней: <code>jax.numpy</code> и <code>jax.lax</code>	126
3.4.1	Базисные элементы управления потоком выполнения.....	127
3.4.2	Повышение (расширение) типа	129
	Упражнение 3.1.....	130
	Резюме	130
4	Вычисление градиентов	132
4.1	Различные способы вычисления производных	134
4.1.1	Дифференцирование вручную	136
4.1.2	Символьное дифференцирование	137
4.1.3	Численное дифференцирование.....	139
4.1.4	Автоматическое дифференцирование	141
4.2	Вычисление градиентов с использованием autodiff	144
4.2.1	Работа с градиентами в TensorFlow	146

4.2.2	Работа с градиентами в PyTorch	148
4.2.3	Работа с градиентами в JAX	149
4.2.4	Производные более высоких порядков	158
4.2.5	Вариант со многими переменными	160
4.3	Прямой и обратный режимы autodiff	164
4.3.1	Трассировка оценок	165
4.3.2	Прямой режим и <code>jvp()</code>	166
4.3.3	Обратный режим и <code>vjp()</code>	171
4.3.4	Материалы для более глубокого изучения	175
	Резюме	176
5	Компиляция кода	178
5.1	Использование компиляции	180
5.1.1	Использование JIT-компиляции	181
5.1.2	Чистые функции и процесс компиляции	190
5.2	Внутренний механизм JIT	193
5.2.1	<i>Jaxpr</i> – промежуточное представление для программ JAX	193
5.2.2	<i>XLA</i>	204
5.2.3	Использование AOT-компиляции	211
5.3	Ограничения JIT-компиляции	216
5.3.1	Чистые функции и функции, не являющиеся чистыми	216
5.3.2	Точные числовые данные	216
5.3.3	Условные выражения, использующие значения входных параметров	216
5.3.4	Медленная компиляция	217
5.3.5	Методы класса	219
5.3.6	Простые функции	221
	Упражнение 5.1	221
	Резюме	221
6	Векторизация кода	223
6.1	Различные способы векторизации функции	224
6.1.1	Простейшие методики	226
6.1.2	Векторизация вручную	229
6.1.3	Автоматическая векторизация	230
6.1.4	Сравнение скорости выполнения	231
6.2	Управление поведением <code>vmap()</code>	233
6.2.1	Управление осями массива для выполнения преобразования	233
6.2.2	Управление осями выходного массива	237
6.2.3	Использование именованных аргументов	238
6.2.4	Использование стиля декоратора	241
6.2.5	Использование коллективных операций	241
6.3	Варианты использования <code>vmap()</code> из реальной практики	243
6.3.1	Обработка пакетов данных	244
6.3.2	Пакетная обработка моделей нейронных сетей	246
6.3.3	Поэлементные градиенты	247
6.3.4	Векторизация циклов	249
	Резюме	253

7	Распараллеливание вычислений	254
7.1	Распараллеливание вычислений с помощью <code>rmap()</code>	256
7.1.1	Установка начальных условий задачи	257
7.1.2	Использование <code>rmap</code> (почти) так же, как <code>imap</code>	260
7.2	Управление поведением <code>rmap()</code>	268
7.2.1	Управление отображением осей входных и выходных данных	268
7.2.2	Использование именованных осей и коллективных операций	276
7.3	Пример программы тренировки нейронной сети с распараллеливанием по данным	285
7.3.1	Подготовка данных и структуры нейронной сети	286
7.3.2	Реализация процедуры тренировки с распараллеливанием по данным	290
7.4	Использование конфигураций с несколькими хостами	296
	Резюме	303
8	Использование сегментирования тензоров	305
8.1	Основы сегментирования тензоров	307
8.1.1	Сетка устройств	310
8.1.2	Позиционное сегментирование	310
8.1.3	Пример с применением двумерной сетки	311
8.1.4	Использование репликации	316
8.1.5	Ограничения сегментирования	319
8.1.6	Сегментирование с именованием	320
8.1.7	Стратегия размещения устройств и ошибки	322
8.2	Многослойный перцептрон с применением сегментирования тензоров	325
8.2.1	Восьмиканальное распараллеливание данных	325
8.2.2	Четырехканальное распараллеливание по данным, двухканальное распараллеливание тензора	327
	Резюме	332
9	Случайные числа в JAX	333
9.1	Генерация случайных данных	335
9.1.1	Загрузка набора данных	337
9.1.2	Генерация случайного шума	340
9.1.3	Выполнение случайного расширения данных	344
9.2	Отличия от NumPy	347
9.2.1	Как работает NumPy	347
9.2.2	Начальное число и состояние в NumPy	349
9.2.3	PRNG в JAX	353
9.2.4	Более подробная конфигурация JAX PRNG	361
9.3	Генерация случайных чисел в реальных приложениях	362
9.3.1	Создание полного конвейера расширения данных	363
9.3.2	Генерация случайных инициализаций для нейронной сети	364
	Резюме	366

10	Работа с pytree	368
10.1	Представление сложных структур данных в форме pytree	370
10.2	Функции для работы с pytree	376
10.2.1	Использование <code>tree_map()</code>	376
10.2.2	Преобразование pytree в плоскую структуру и восстановление древовидной формы	380
10.2.3	Использование <code>tree_reduce()</code>	383
10.2.4	Транспонирование pytree	384
10.3	Создание специализированных узлов pytree	387
	Резюме	391
Часть III ЭКОСИСТЕМА		393
11	Высокоуровневые библиотеки поддержки нейронных сетей	395
11.1	Классификация изображений MNIST с использованием многослойного перцептрана	397
11.1.1	Многослойный перцептрон в Flax	397
11.1.2	Библиотека трансформации градиентов Optax	405
11.1.3	Тренировка нейронной сети с применением Flax	408
11.2	Классификация изображений с использованием ResNet	413
11.2.1	Управление состоянием в Flax	415
11.2.2	Сохранение и загрузка модели с использованием Orbax	422
11.3	Использование экосистемы Hugging Face	425
11.3.1	Использование предварительно натренированной модели из хранилища Hugging Face Model Hub	427
11.3.2	Более подробное изучение процессов точной настройки и предварительной тренировки	435
11.3.3	Использование библиотеки диффузоров	438
	Резюме	443
12	Другие члены экосистемы JAX	445
12.1	Экосистема глубокого обучения	446
12.1.1	Высокоуровневые библиотеки поддержки нейронных сетей	446
12.1.2	Большие языковые модели в JAX	448
12.1.3	Библиотеки утилит	451
12.2	Модули машинного обучения	454
12.2.1	Обучение с подкреплением	454
12.2.2	Прочие библиотеки машинного обучения	455
12.3	Модули JAX для других сфер деятельности	457
	Резюме	459
	<i>Приложение A. Установка JAX</i>	460
	<i>Приложение B. Использование Google Colab</i>	464
	<i>Приложение C. Использование Google Cloud TPU</i>	467
	<i>Приложение D. Экспериментальные средства распараллеливания</i>	472
	<i>Предметный указатель</i>	513

Предисловие

JAX – это мощная библиотека на языке Python, созданная компанией Google для глубокого обучения и высокопроизводительных вычислений. Библиотека широко используется в научных исследованиях в области машинного обучения и позиционируется как третий по известности и распространенности фреймворк глубокого обучения, уступая только TensorFlow и PyTorch. В частности, это практически «штатный» фреймворк в таких компаниях, как DeepMind, а исследования Google все в большей степени основываются на JAX.

Что мне действительно нравится в JAX, так это его акцент на функциональном программировании в глубоком обучении. Этот фреймворк предоставляет надежные функциональные преобразования, включая вычисление градиента, JIT-компиляцию посредством XLA, автоматическую векторизацию и возможности распараллеливания. JAX поддерживает графические (GPU) и тензорные (TPU) процессоры, обеспечивая впечатляющую производительность.

Сейчас самое подходящее время для подробного и глубокого изучения JAX, так как его экосистема стремительно расширяется. Несмотря на то что фреймворк существует уже несколько лет, все же наблюдается очевидная недостаточность всеобъемлющих ресурсов для новичков. Хотя веб-сайт JAX предлагает обширную документацию и поддерживающее сообщество, объединение всех источников информации, особенно при интеграции других библиотек, может показаться обескураживающим.

Эта книга создана для тех, кто стремится освоить JAX. Ее цель – объединить самую важную информацию в одном месте и помочь читателям понять концепции JAX, улучшить их навыки и способность применять JAX в проектах и исследованиях.

Предполагается знание базовых принципов глубокого обучения и наличие опыта практической работы с языком Python. В книге не рассматриваются основы глубокого обучения, поскольку по этой

теме существует множество информационных ресурсов. Вместо этого все внимание сосредоточено исключительно на JAX, хотя при необходимости главные концепции глубокого обучения все же кратко излагаются. Такой подход должен оказаться полезным для тех, кто не имеет опыта работы в сфере глубокого обучения, например физикам.

JAX – это нечто большее, нежели просто фреймворк глубокого обучения. Диапазон его модулей постоянно расширяется и выходит за рамки глубокого обучения, создавая и укрепляя потенциальные возможности в области дифференцируемого программирования, крупномасштабных физических имитаций и многих других. Надеюсь, что эта книга также будет полезна тем, кто интересуется подобными приложениями.

JAX продолжает развиваться, поэтому несколько глав книги существенно обновились. Но не стоит беспокоиться о возможных изменениях в будущем, потому что основополагающие знания, которые вы получите при чтении, останутся применимыми в следующих версиях JAX.

Благодарности

Написание этой книги заняло больше времени, чем я предполагал. В процессе работы над ней я сменил несколько стран, да и версии JAX менялись. Некоторые главы пришлось переписывать. Но теперь все готово!

Прежде всего хочу поблагодарить мою семью – жену Милу, сыновей Даню и Федю. Вы так долго скучали без моего внимания! Но все время поддерживали меня.

Хочу поблагодарить народ Армении, где мы жили некоторое время, за его доброту и гостеприимство. Особая благодарность Ереванскому стартап-сообществу за помощь и поддержку. Грант Хачатрян (Hrant Khachatrian), Завен Навоян (Zaven Navoyan), Арсен Егиазарян (Arsen Yeghiazaryan), Андраник Хачатрян (Andranik Khachatryan), Ашот Арзуманян (Ashot Arzumanyan), Аш Варданян (Ash Vardanian), Адам Биттлингмайер (Adam Bittlingmayer), Артур Алексанян (Artur Aleksanyan), Эрик Аракелян (Erik Arakelyan), Карен Гюльбудагян (Karén Gyulbudaghyan) – огромное вам спасибо!

Спасибо организации Enterprise Armenia, национальному агентству по привлечению инвестиций Армении (National Investment Promotion Agency of Armenia). Вы делаете огромную работу, и ваша помощь бесцenna.

Благодарю редакторов издательства Manning Патрика Барба (Patrick Barb), Бекки Уитни (Becky Whitney) и Франсис Лефковитц (Frances Lefkowitz). Даже несмотря на то, что при работе над книгой сменились три редактора и в нее было внесено множество изменений, каждый из вас внес свой вклад в ее ценность. Также благодарю Майка Стивенса (Mike Stephens) и Марьяна Баце (Marjan Bace), которые с самого начала, с моих первых замыслов верили в то, что книга будет написана.

Спасибо моему техническому редактору Нику МакГрейви (Nick McGreivy), который, помимо того что является докторантом Прин-

стонского университета, где он изучает физику плазмы, использует JAX в своих исследованиях для оптимизации научных экспериментов, а также для интеграции методов глубокого обучения в численное моделирование. Также благодаря техническому корректору Костаса Пассадиса (Kostas Passadis) и рецензентов Арслана Габдулхакова (Arslan Gabdulkhakov), Чан Сун Пака (Chansung Park), Филиппа Дорнеля (Fillipe Dornelas), Джеймса Блэка (James Black), Джеймса Ванга (James Wang), Цзюнь Цзян (Jun Jiang), Кейт Ким (Keith Kim), Люсиана-Поля Торье (Lucian-Paul Torje), Максима Волгина (Maxim Volgin), Наджиба Арифа (Najeeb Arif), Ору Голану (Or Golan), Ритобрата Гошу (Ritobrata Ghosh), Сен Юн Ли (Seunghyun Lee), Симоне Де Бони (Simone De Bonis), Стивену Оутсу (Stephen Oates), Тони Холдройду (Tony Holdroyd), Видья Винаю (Vidhya Vinay) и Войте Тума (Vojta Tuma). Вы все предоставили множество полезных комментариев и предложений, которые помогли улучшить эту книгу. С учетом вышесказанного, все оставшиеся в книге ошибки целиком и полностью лежат на моей совести.

И наконец, благодарю моих друзей, экспертов GDE (Google Developer Experts) и компанию Google за поддержку столь крупномасштабной инициативы. Сообщество GDE достойно наивысшей похвалы! Многие эксперты GDE просматривали ранние версии и давали полезную обратную связь. Особая благодарность Дэвиду Кардоzo (David Cardozo) за его весьма ценные отзывы и замечания.

О книге

Книга «Глубокое обучение с JAX» написана для того, чтобы помочь читателям понять JAX и начать практически применять этот фреймворк в проектах и исследованиях. В книге собрана вся наиболее важная информация, которая позволит понять концепции JAX. Многочисленные простые для понимания примеры облегчают восприятие этой темы.

Для кого предназначена эта книга

Книга «Глубокое обучение с JAX» ориентирована на специалистов-практиков и исследователей в области глубокого обучения, знакомых с такими фреймворками, как PyTorch и TensorFlow, и желающих начать использовать JAX. Исследователи в других областях (например, в физике или в сфере оптимизации) или аспиранты, специализирующиеся в глубоком обучении, численных методах оптимизации или в распределенных вычислениях, также найдут эту книгу полезной для обучения и практической деятельности.

Как организована эта книга: общая схема

Книга состоит из трех частей, содержащих 12 глав.

В части I представлено введение и демонстрация возможностей JAX:

- глава 1 отвечает на самый важный вопрос: «Почему именно JAX?» Здесь объясняется, что такое JAX, описываются его сильные и слабые стороны по сравнению с другими фреймворками, такими как TensorFlow и PyTorch, а также особо отмечается, когда JAX может стать самым лучшим инструментом для вашего проекта;

- в главе 2 вы получите первый практический опыт использования JAX. Мы создадим простую нейронную сеть для классификации изображений, а кроме того, будут представлены основные концепции: преобразования JAX для автоматической векторизации, вычисления градиентов и JIT-компиляции. Вы также узнаете, как сохранять и загружать модели, и поймете различие между чистыми функциями и функциями с побочным эффектом в JAX.

В части II рассматриваются основные функциональные средства и возможности JAX:

- в главе 3 используется «рабочая лошадка» глубокого обучения: тензоры, или многомерные массивы. Сравниваются массивы NumPy и JAX, обсуждается работа с ними на разнообразных аппаратных устройствах, таких как центральные процессоры (CPU), графические процессоры (GPU) и тензорные процессоры (TPU), объясняются нюансы адаптации исходного кода с рассмотрением различий при использовании NumPy и JAX;
- в главе 4 рассматривается чрезвычайно важная задача вычисления градиентов, решение которой является необходимым условием для тренировки нейронных сетей. Сравниваются разнообразные методы дифференцирования, очень подробно рассматриваются возможности автоматического дифференцирования в JAX, а также использование режимов прямого и обратного автоматического дифференцирования;
- в главе 5 показано, как оптимизировать код для повышения производительности, используя JIT-компиляцию. Рассматриваются внутреннее устройство и работа механизма JIT и его взаимодействие с компилятором XLA, а также способы устранения потенциальных ограничений;
- в главе 6 представлена автоматическая векторизация, мощная методика эффективной обработки пакетов данных. Рассматриваются разнообразные методы векторизации, объясняется, как управлять JAX-преобразованием `vtar()`, а также анализируются сценарии из реальной практики, где автоматическая реализация показывает себя во всем блеске;
- в главе 7 основное внимание уделено распараллеливанию, позволяющему одновременно выполнять вычисления на нескольких устройствах. Объясняется, как использовать преобразование `rtar()` для параллельного выполнения, как управлять его поведением. Рассматривается распараллеливание данных для тренировки нейронной сети. Также используется код для реальной работы на конфигурациях с несколькими хостами для выполнения крупномасштабных задач;
- в главе 8 представлено сегментирование тензоров, инновационная эффективная методика распараллеливания в JAX. По-

казано, как применять XLA для автоматического распараллеливания. Рассматривается реализация параллельного режима работы с данными и тензорами для тренировки нейронной сети, а также преимущества этой методики;

- в главе 9 разбирается важная тема – генерация случайных чисел в JAX. Описываются различия между JAX и NumPy в этом аспекте, обсуждается роль ключей в представлении состояния генераторов случайных чисел, объясняется, как применить эти концепции в реальных приложениях;
- глава 10 знакомит читателей с pytrees, мощным инструментальным средством для представления сложных структур данных в JAX. Рассматриваются методы эффективной работы с pytrees, использование функций для обработки деревьев и даже создание специализированных узлов pytree для особых потребностей.

Часть III представляет богатую функциями и характеризующуюся большим разнообразием экосистему библиотек, созданных на основе JAX:

- в главе 11 представлены библиотеки поддержки нейронных сетей более высокого уровня, такие как Flax и Optax, предоставляющие удобные абстракции для создания и тренировки сложных моделей. Мы будем использовать Flax для создания простого MLP (многослойного перцептрона) и более продвинутой остаточной нейросети для классификации изображений, а также узнаем, как применять библиотеки Hugging Face для работы с трансформерами и диффузионными моделями;
- глава 12 дает более широкий обзор экосистемы JAX с описанием библиотек для решения разнообразных задач машинного обучения, в том числе для тренировки больших языковых моделей (LLM), обучения с подкреплением и эволюционных вычислений. Также рассматриваются модули JAX для использования в других областях науки, таких как физика, химия и т. д.

Если вы административный работник, то рекомендуется прочитать первые две главы, чтобы узнать о сильных сторонах JAX, о его отличиях от PyTorch и TensorFlow и о том, как выглядит типичный проект машинного обучения с применением JAX. Глава 12 также не содержит технической информации и может дать представление о том, где JAX проявляет себя с самой лучшей стороны.

Для разработчиков, желающих как можно быстрее приступить к созданию нейронных сетей с помощью JAX, рекомендуется сосредоточить основное внимание на главе 2, чтобы рассмотреть простой пример глубокого обучения, на главах 3–6 для изучения основополагающих концепций JAX и на главе 11 для обзора библиотек высокого уровня в экосистеме. Остальную часть книги можно читать

в любом порядке в зависимости от ваших конкретных интересов. Можно пропустить главы 7 и 8, если распаралеливание пока не входит в ваши планы, – вы можете вернуться к ним в любое время. Если вас интересует генерация случайных чисел и использование pytrees, то обратите особое внимание на главы 9 и 10, хотя предыдущие главы предоставляют вполне достаточную базовую информацию, для того чтобы начать практическое использование фреймворка JAX.

Исходный код примеров

Книга содержит множество примеров исходного кода – в пронумерованных листингах и в строках обычного текста. В обоих случаях исходный код выделяется монотонным шрифтом, для того чтобы можно было отличить его от обычного текста. Иногда фрагменты или строки исходного кода дополнительно выделяются **полужирным шрифтом**, чтобы выделить код, изменившийся по сравнению с предыдущими примерами текущей главы, например при добавлении новой функциональности в существующую строку кода.

Во многих случаях оригинальный исходный код был переформатирован – добавлялись разрывы строк и корректировалось выравнивание для размещения в доступном пространстве страницы книги. В некоторых случаях даже такие меры оказывались недостаточными, поэтому в листингах включались маркеры продолжения текущей строки (➡). Кроме того, комментарии часто удалялись из исходного кода, когда код был подробно описан в тексте. Многие листинги сопровождаются примечаниями, особо выделяющими важные концепции.

Готовые для выполнения фрагменты кода можно получить из (онлайновой) версии liveBook этой книги: <https://livebook.manning.com/book/deep-learning-with-jax>. Полные коды примеров из этой книги доступны для загрузки с веб-сайта издательства Manning www.manning.com, а также из репозитория GitHub: <https://github.com/che-shr-cat/JAX-in-Action>.

Почти для каждой главы существует соответствующий блокнот Colab (или несколько блокнотов). Код протестирован для JAX версии 0.4.14.

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 на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзы-

вы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и 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 обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Manning Publications очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

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