Мне удалось завершить это уникальное путешествие в течение долгих ночей только благодаря поддержке моей жены Элеоноры. Я посвящаю эту книгу ей, с самого начала верившей в этот проект.

Содержание

От издательства

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 «Отзывы 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. Издательство «ДМК Пресс» очень серьезно относится 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.

Введение

Без сомнения, индустрия высоких технологий продолжает оказывать все большее влияние на нашу повседневную жизнь. Изменения столь же стремительны, сколь и постоянны, и происходят повсюду вокруг нас – в наших телефонах, автомобилях, интеллектуальных динамиках и микрогаджетах, которые мы используем для повышения эффективности, комфорта и возможностей коммуникации. Машинное обучение (*machine learning*, ML) – одна из самых преобразующих технологий нашего времени. Предприятия, ученые и инженерные сообщества продолжают углублять понимание, развивать и исследовать возможности этой невероятной технологии и служат все большему раскрытию ее потенциала для создания новых вариантов использования во многих отраслях.

Я менеджер по продуктам машинного обучения в компании ARM. В этой роли я нахожусь в центре революции ML, которая происходит в смартфонах, автомобильной промышленности, играх, AR, VR[1] и других областях. Мне ясно, что в ближайшем будущем функциональность ML будет в каждом отдельном электронном устройстве, – от крупнейших в мире суперкомпьютеров до самых маленьких и маломощных микроконтроллеров. Работа в области ML познакомила меня с некоторыми из самых блестящих и ярких умов в области технологий – теми, кто бросает вызов традиции, задает сложные вопросы и открывает новые ценности благодаря использованию ML.

Когда я впервые встретил Джана Марко, я едва мог произнести «ML», но в то время он уже был ветераном в этом космосе. Я был поражен широтой и глубиной его знаний и его способностью решать сложные проблемы. Вместе с командой ARM он работал над созданием Arm Compute Library (ACL) – самой эффективной библиотеки, доступной для ML на платформе ARM. Успех ACL не имеет себе равных. Он развернут на миллиардах устройств по всему миру – от серверов и флагманских смартфонов до интеллектуальных духовых шкафов.

Когда Джан Марко сказал мне, что пишет книгу о ML, моей немедленной реакцией было: «Какую часть?» Экосистема ML настолько разнообразна, что необходимо учитывать множество различных технологий, платформ и фреймворков[2]. Я знал, что благодаря обширным знаниям всех аспектов ML он был подходящим человеком для этой работы. Кроме того, Джан Марко обладает удивительной способностью объяснять вещи прямо и логично.

Книга Джана Марко раскрывает мир TinyML, проводя нас через ряд практических примеров из реального мира. Каждый пример изложен в форме рецепта, в четком и последовательном стиле, обеспечивающем простое по-

---

[1] AR (*augmented reality*) – дополненная реальность; VR (*virtual reality*) – виртуальная реальность. – *Здесь и далее прим. перев.*

[2] Фреймворк (*framework*) – программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

шаговое руководство. Начиная с базовых принципов, он растолковывает основы электронных или программных технологий, которые будут использоваться в примере. Затем в книге рассказывается об используемых платформах и технологиях, за которыми следуют основы ML, – разработка моделей нейронных сетей, проведение обучения и разворачивания моделей на целевом устройстве. Это действительно стиль кулинарного руководства по приготовлению супа с орехами. Каждый пример немного сложнее предыдущего, и в них удачно сочетаются традиционные и новые технологии. Вы не просто узнаете «как», вы также получите понимание «почему». Когда дело доходит до периферийных устройств, эта книга действительно дает панорамный обзор области ML.

Машинное обучение продолжает менять все аспекты технологий, и разработчикам программного обеспечения необходимо начинать его изучение. Эта книга позволяет быстро адаптироваться благодаря использованию легкодоступных и недорогих аппаратных средств. Независимо от того, являетесь ли вы новичком в ML или имеете некоторый опыт, каждый пример будет содержать новые знания и оставлять достаточно возможностей для саморазвития и дальнейших экспериментов. Независимо от того, используете ли вы эту книгу в качестве учебника или справочника, вы создадите прочную основу в области ML для будущего развития. Это даст вашей команде возможность достичь повышения эффективности и производительности, получить новые идеи и новые возможности для ваших продуктов.

Ронан Нотон,
старший менеджер по продуктам машинного обучения в ARM

Составители

Об авторе

Джан Марко Йодиче (Gian Marco Iodice) – командный и технологический лидер группы Machine Learning Group в компании ARM. В 2017 году он стал соавтором библиотеки ARM Compute Library (ACL). ACL в настоящее время является самой эффективной библиотекой для ML на процессорах ARM, она развернута на миллиардах устройств по всему миру – от серверов до смартфонов.

Джан Марко получил степень магистра с отличием в области электронной инженерии в Пизанском университете (Италия) и имеет многолетний опыт разработки алгоритмов ML и компьютерного зрения на периферийных устройствах. Теперь он руководит оптимизацией производительности ML на графических процессорах (GPU) ARM Mali.

В 2020 году Джан Марко стал соучредителем английского сообщества TinyML, возникшего с целью поощрения обмена знаниями, обучения и мотивации следующего поколения разработчиков ML на малогабаритных и энергоэффективных устройствах.

О рецензентах

Алессандро Гранде (Alessandro Grande) – физик, инженер, коммуникатор и технологический лидер, страстно желающий объединить людей и расширить их возможности для создания более эффективных и устойчивых технологий. Алессандро является директором по продуктам Edge Impulse и соучредителем сообщества TinyML в Великобритании и Италии. До прихода в Edge Impulse Алессандро работал в ARM евангелистом-разработчиком и менеджером экосистемы, уделяя особое внимание созданию основ более умного и эффективного интернета вещей (IoT). Он имеет степень магистра в области ядерной и электронной физики от Римского университета «Сапиенца».

Дакш Трехан (Daksh Trehan) начал свою карьеру в качестве аналитика, руководствуясь особой любовью к статистике и обработке данных. Различные

статистические методы познакомили его с миром ML и *data science*[3]. Хотя Дакш Трехан сосредоточен на анализе данных, он любит заниматься прогнозами с использованием ML. Он понимает значение данных в современном мире и постоянно пытается изменить мир, используя различные методы ML и свои навыки визуализации данных.

Дакш Трехан любит писать статьи о ML и искусственном интеллекте, и они принесли ему более 100 000 просмотров на сегодняшний день. Он также внес свой вклад в качестве консультанта по ML в книгу о TikTok, написанную доктором Маркусом Рейчем и доступную в магазине электронных книг Amazon.

[3] Data science (букв. наука о данных) – общее название широкого спектра дисциплин, занимающихся анализом и обработкой данных в различных областях бизнеса, науки или инженерных разработок. Само по себе название «Data scientist» в смысле обозначения профессии вы можете встретить крайне редко – обычно такие специализации называются по своим конкретным областям (напр., специалист по математической статистике – обработке результатов эксперимента и проверке статистических гипотез – или специалист по анализу данных). К «наукам о данных» относится в том числе и создание и внедрение алгоритмов машинного обучения, о которых идет речь в этой книге.

Предисловие

Эта книга о TinyML, быстрорастущей области на пересечении **машинного обучения (ML)** и встраиваемых систем, позволяющей искусственному интеллекту (ИИ) работать на устройствах на основе микроконтроллеров с чрезвычайно низким энергопотреблением.

TinyML – захватывающая область, полная возможностей. При небольшом бюджете мы можем создавать устройства, разумно взаимодействующие с окружающим миром и меняющие наш образ жизни к лучшему. Однако к этой области может быть трудно подступиться, если мы исходим из области ML и мало знакомы с микроконтроллерами. Цель этой книги – разрушить подобные барьеры и на практических примерах сделать TinyML доступным для разработчиков, не имеющих опыта программирования встраиваемых систем. Каждая глава будет представлять собой самостоятельный проект, позволяющий узнать, как использовать технологии, лежащие в основе TinyML, для взаимодействия с электронными компонентами (например, датчиками), и развертывать модели ML на устройствах с ограниченным объемом памяти.

«*Поваренная книга TinyML*» начинается с практического введения в эту междисциплинарную область, чтобы ознакомить вас с некоторыми основами развертывания интеллектуальных приложений на Arduino Nano 33 BLE Sense и Raspberry Pi Pico. По мере продвижения вы будете решать различные задачи, с которыми можете столкнуться при создании прототипов устройств на микроконтроллерах, вроде управления состоянием светодиода с помощью GPIO-вывода и кнопки или подачей питания на устройство с помощью батарей. После этого мы поговорим о примерах, касающихся измерителя температуры-влажности или датчиков **три-V** (**голос** (*voice*), **зрение** (*view*) и **вибрация** (*vibration*)), чтобы получить необходимые навыки для внедрения комплексных интеллектуальных приложений в различных сценариях. Затем вы изучите рекомендации по созданию уменьшенных моделей для микроконтроллеров с ограниченным объемом памяти. Наконец, вы познакомитесь с двумя самыми последними технологиями, microTVM и microNPU, которые помогут усовершенствовать ваши игры с TinyML.

К концу этой книги вы будете хорошо разбираться в лучших практиках и фреймворках ML, позволяющих разрабатывать ML-приложения на микроконтроллерах, и будете иметь четкое представление о ключевых аспектах, которые следует учитывать на этапе разработки.

Для кого предназначена эта книга

Эта книга предназначена для инженеров-разработчиков ML, заинтересованных в быстрой разработке приложений ML на микроконтроллерах с помощью практических примеров. Книга поможет вам расширить знания о революции TinyML, получить навыки создания комплексных интеллектуальных

проектов с использованием реальных датчиков данных на Arduino Nano 33 BLE Sense и Raspberry Pi Pico. Требуется базовое знакомство с C/C++, программированием на Python и **интерфейсом командной строки** (**CLI**). Однако никаких предварительных знаний о микроконтроллерах не требуется.

О ЧЕМ РАССКАЗЫВАЕТ ЭТА КНИГА

В главе 1 «*Начало работы с TinyML*» представлен обзор TinyML, а также возможности и проблемы, связанные с внедрением ML на микроконтроллерах с чрезвычайно низким энергопотреблением. В этой главе основное внимание уделяется фундаментальным элементам ML, энергопотреблению и микроконтроллерам, отличиям TinyML от обычного ML в облаке, на настольных компьютерах или даже смартфонах.

В главе 2 «*Прототипирование на микроконтроллерах*» представлены краткие и простые проекты, позволяющие разобраться с соответствующими основами программирования микроконтроллеров. Мы разберемся с отладкой кода и тем, как передавать данные на монитор последовательного порта Arduino. После этого мы узнаем, как программировать GPIO-периферию, управляющую выводами микроконтроллера, с помощью ARM Mbed API и использовать макетную плату для подключения внешних компонентов, таких как светодиоды и кнопки. В конце мы узнаем, как питать Arduino Nano 33 BLE Sense и Raspberry PI Pico от батареек.

Глава 3 «*Создание метеостанции с помощью библиотеки TensorFlow Lite for microcontrollers*» проведет вас через все этапы разработки приложения на основе библиотеки TensorFlow Lite for microcontrollers[4] и научит получать данные датчиков температуры и влажности. Приложение, которое будет разработано в этой главе, представляет собой метеостанцию на базе ML для прогнозирования снегопада.

Сначала мы сосредоточимся на подготовке набора данных путем получения исторических данных о погоде из WorldWeatherOnline. После этого мы представим соответствующие основы обучения и тестирования модели на основе библиотеки TensorFlow. В конце концов мы развернем модель на Arduino Nano 33 BLE Sense и Raspberry Pi Pico с библиотекой TensorFlow Lite for microcontrollers.

В главе 4 «*Голосовое управление светодиодами с помощью Edge Impulse*» показано, как разработать сквозное приложение для определения ключевых слов (**keyword spotting, KWS**) с помощью Edge Impulse и ознакомиться со сбором аудиоданных и аналого-цифровыми преобразователями (**АЦП**). Приложение, рассмотренное в этой главе, управляет цветом светодиода (красный, зеленый и синий) и количеством миганий (один, два и три).

---

[4] TensorFlow – библиотека машинного обучения, разработанная Google и ориентированная на обычные среды выполнения (серверы, облачные хранилища, ПК и смартфоны). В 2017 году была представлена специальная облегченная версия TensorFlow Lite для мобильных и малопотребляющих устройств с небольшим объемом памяти.

Сначала мы сосредоточимся на подготовке набора данных, показав, как получать аудиоданные с помощью мобильного телефона. После этого мы разработаем модель с использованием функций **MFCC**[5] и оптимизируем производительность с помощью EON Tuner. В конце концов мы доработаем приложение KWS на Arduino Nano 33 BLE Sense и Raspberry Pi Pico.

Глава 5 «*Распознавание интерьеров помещений с помощью TensorFlow Lite for microcontrollers и Arduino Nano*» призвана показать вам, как применять трансфертное обучение[6] с помощью TensorFlow и ознакомиться с лучшими практиками использования модуля камеры с микроконтроллером. Для целей этой главы мы разработаем приложение для распознавания интерьеров с помощью Arduino Nano 33 BLE Sense и модуля камеры OV7670.

В первой части мы увидим, как получать изображения с модуля камеры OV7670. После этого мы сосредоточимся на дизайне модели, применяя трансфертное обучение с помощью Keras[7] для распознавания кухонных и ванных комнат. В конце концов мы развернем квантизованную модель на Arduino Nano 33 BLE Sense с помощью TensorFlow Lite for microcontrollers.

Глава 6 «*Создание интерфейса на основе жестов для управления воспроизведением на YouTube*» направлена на разработку сквозного приложения для распознавания жестов с помощью Edge Impulse и Raspberry Pi Pico. Оно познакомит вас с инерциальными датчиками, научит использовать периферийные устройства I2C и создавать многопоточные приложения в Arm Mbed OS.

Сначала мы соберем данные акселерометра с помощью сборщика данных Edge Impulse data forwarder. После этого разработаем модель с использованием функций в частотной области для распознавания трех жестов. В конце концов мы развернем приложение на Raspberry Pi Pico и внедрим программу на Python с библиотекой PyAutoGUI для создания бесконтактного интерфейса для управления воспроизведением видео на YouTube.

В главе 7 «*Запуск модели TinyML CIFAR-10 на виртуальной платформе ОС Zephyr*» приведены рекомендации по созданию уменьшенных моделей для микроконтроллеров с ограниченным объемом памяти. В этой главе мы будем разрабатывать модель на основе набора данных для классификации изображений CIFAR-10 на виртуальном микроконтроллере на базе ARM Cortex-M3.

Сначала мы установим Zephyr, основной фреймворк, используемый в этой главе для выполнения нашей задачи. После этого разработаем малую квантизованную модель CIFAR-10 с библиотекой TensorFlow. Эта модель подойдет для микроконтроллера с объемом программной памяти всего 256 Кбайт и оперативной памятью 64 Кбайт. В конце концов мы создадим приложение

---

[5] MFCC (Mel-frequency cepstral coefficients) – сложный алгоритм анализа звуковых фрагментов с целью выделения характерных частот для распознавания речи. Подробнее об MFCC на русском языке см. https://habr.com/ru/post/140828/.

[6] Подробно о трансфертном обучении рассказывается в главе 5.

[7] Keras – высокоуровневый API, надстройка над библиотеками машинного обучения (в том числе Tensorflow), позволяющий реализовать их функциональность наиболее простым образом.

для классификации изображений с помощью TensorFlow Lite for microcontrollers и ОС Zephyr и запустим его на виртуальной платформе с помощью Quick Emulator (QEMU).

Глава 8 «*К следующему поколению TinyML с microNPU*» поможет вам ознакомиться с microNPU, новым классом процессоров для работы ML на периферийных устройствах. В этой главе мы будем запускать квантизованную модель CIFAR-10 на виртуальном контроллере ARM Ethos-U55 microNPU с помощью оптимизирующего компилятора TVM.

Сначала мы узнаем, как работает микропроцессор ARM Ethos-U55, и установим программные средства для сборки и запуска модели на фиксированной виртуальной платформе ARM Corstone-300. После этого мы будем использовать компилятор TVM для преобразования предварительно обученной модели TensorFlow Lite в код на языке Си. В конце мы покажем, как скомпилировать и развернуть код, сгенерированный TVM, в ARM Corstone-300 для выполнения вычислений с помощью ARM Ethos-U55 microNPU.

Как извлечь максимум пользы из этой книги

Вам понадобится компьютер (ноутбук или настольный компьютер) с архитектурой x86-64 и по крайней мере одним USB-портом для программирования плат Arduino Nano 33 BLE Sense и Raspberry Pi Pico. Для первых шести глав вы можете использовать Ubuntu 18.04 (или более позднюю версию) или Windows (например, Windows 10) в качестве операционной системы. Однако вам понадобится Ubuntu 18.04 (или более поздняя версия) для главы 7, посвященной запуску малой модели CIFAR-10 на виртуальной платформе с ОС Zephyr, и главы 8, посвященной следующему поколению TinyML с microNPU.

Необходимо иметь на вашем компьютере следующие программы:

- Python (рекомендуется Python 3.7),
- текстовый редактор (например, gedit в Ubuntu),
- медиаплеер (например, VLC),
- средство просмотра изображений (например, приложение по умолчанию в Ubuntu или Windows 10),
- веб-браузер (например, Google Chrome).

Для путешествия по TinyML нам понадобятся различные программные средства для разработки ML и программирования встроенных систем. Благодаря Arduino, Edge Impulse и Google эти инструменты будут находиться в облаке с доступом на основе браузера и с бесплатным планом использования.

Программы Arduino Nano 33 BLE Sense и Raspberry Pi Pico будут разрабатываться непосредственно в веб-браузере с помощью веб-редактора Arduino (https://create.arduino.cc). Однако бесплатный веб-редактор Arduino имеет ограничение в 200 с времени компиляции в день. Поэтому вы можете рассмотреть возможность перехода на любой платный тарифный план или на использование бесплатного локального редактора Arduino IDE (https://www.arduino.cc/en/software), чтобы получить неограниченное время компиляции. Если вас интересует бесплатная локальная среда разработки Arduino IDE, мы

предоставили на GitHub (https://github.com/PacktPublishing/TinyML-Cookbook/blob/main/Docs/setup_local_arduino_ide.md) инструкции по ее настройке.

В следующей таблице суммированы сведения об аппаратных устройствах и программных средствах, рассмотренных в каждой главе.

Глава	Аппаратные устройства	Программные средства
1	Arduino Nano 33 BLE Sense Raspberry Pi Pico	Arduino Web Editor
2	Arduino Nano 33 BLE Sense Raspberry Pi Pico	Arduino Web Editor, Google Colaboratory
3	Arduino Nano 33 BLE Sense Raspberry Pi Pico	Arduino Web Editor, Google Colaboratory
4	Arduino Nano 33 BLE Sense Raspberry Pi Pico	Arduino Web Editor, Edge Impulse Python 3.6 (local)
5	Arduino Nano 33 BLE Sense	Arduino Web Editor, Google Colaboratory Python 3.6 (local)
6	Raspberry Pi Pico	Arduino Web Editor, Edge Impulse Python 3.6 (local)
7	Виртуальная платформа	Google Colaboratory, Python 3.6 (local) Zephyr SDK
8	Виртуальная платформа	ARM Corstone-300, Python 3.6 (local) TVM/microTVM

Для проектов могут потребоваться датчики и дополнительные электронные компоненты для создания реалистичных прототипов TinyML и полного процесса разработки. Все компоненты перечислены в начале каждой главы и в файле *README.md* на GitHub (https://github.com/PacktPublishing/TinyML-Cookbook). Поскольку вы будете создавать настоящие электронные схемы, нам потребуется комплект электронных компонентов, который включает в себя по крайней мере беспаечную макетную плату, разноцветные светодиоды, резисторы, кнопки и соединительные провода-перемычки. Не волнуйтесь, если вы новичок в электронике, – вы познакомитесь с этими компонентами в первых двух главах этой книги. Кроме того, мы подготовили список покупок для начинающих на GitHub, чтобы вы точно знали, что купить: https://github.com/PacktPublishing/TinyML-Cookbook/blob/main/Docs/shopping_list.md[8].

Если вы используете цифровую версию этой книги, мы советуем вам вводить код самостоятельно или получить доступ к коду через репозиторий GitHub (ссылка доступна ниже). Это поможет вам избежать любых потенциальных ошибок, связанных с копированием и вставкой кода.

[8] Указанный по ссылке список ориентирован на реалии США и Европы. Рекомендации по приобретению в российских условиях оборудования или компонентов будут размещаться в разделах «*Технические требования*» соответствующих глав. Отсутствие такой рекомендации означает, что компонент широко доступен и может быть приобретен без длительного поиска в основных отечественных интернет-магазинах электроники (http://www.chipdip.ru, http://iarduino.ru, https://www.electronshik.ru, https://dip8.ru и др.), а также на https://aliexpress.ru.

Загрузка файлов с примерами кода

Вы можете загрузить файлы с примерами кода для этой книги с GitHub по адресу https://github.com/PacktPublishing/TinyML-Cookbook. В случае обновления кода он также будет обновлен в существующем репозитории GitHub.

У нас также есть другие пакеты кода из нашего богатого каталога книг и видео, доступных по адресу https://github.com/PacktPublishing. Ознакомьтесь с ними!

Загрузка цветных изображений

Мы предоставляем PDF-файл, содержащий цветные изображения скриншотов и графиков, используемых в этой книге. Вы можете скачать его здесь: https://static.packt-cdn.com/downloads/9781801814973_ColorImages.pdf.

Текстовые соглашения

В этой книге используется ряд текстовых соглашений.

Гиперссылки, домены и интернет-адреса: выделяются жирным курсивом, если они не представляют собой законченный интернет-адрес с указанием протокола (URL). В последнем случае ссылка выделена синим шрифтом: https://github.com.

Имя файла: курсивом указаны имена папок, имена файлов, расширения файлов, пути.

Пример: «Войдите в папку *~/project_npu* и создайте три папки с именами *binaries*, *src* и *sw_libs*».

Также *курсивом* указываются названия глав и разделов и указываемые в скобках эквиваленты переводных терминов (напр. «узел (*node*)»).

`Код в тексте`: моноширинный шрифт указывает кодовые обозначения в тексте, команды, имена таблиц базы данных, фиктивные URL-адреса, вводимые пользователем, и дескрипторы Twitter.

`Блок кода` задается следующим образом:

```
export PATH=~/project_npu/binaries/FVP_Corstone_SSE-300/models/Linux64_GCC-6.4:$PATH
```

Когда мы хотим привлечь внимание к определенной части блока кода, соответствующие строки или элементы выделяются **`жирным`**:

```
[default]
exten => s,1,Dial(Zap/1|30)
exten => s,2,Voicemail(u100)
exten => s,102,Voicemail(b100)
exten => i,1,Voicemail(s0)
```

Любой ввод или вывод из командной строки записывается следующим образом:

```
$ cd ~/project_npu
$ mkdir binaries
$ mkdir src
```

Жирный шрифт: также обозначает новый термин, важное слово или названия, которые вы видите на экране.

Например, названия в меню или диалоговых окнах отображаются в тексте следующим образом: «Нажмите на **Corstone-300 Ecosystem FVPs**, а затем нажмите на кнопку **Download Linux**». Русский перевод пунктов меню и названий кнопок, если это необходимо для лучшего понимания текста, указывается курсивом в скобках. Например, указанная фраза может заканчиваться следующим образом: «нажмите на кнопку **Download Linux** (*Загрузить Linux*)».

Важные примечания выглядят так.

Примечания выглядят вот так.

Подсказки выглядят вот так.

Часто используемые разделы

В этой книге вы найдете несколько часто появляющихся заголовков. Подробности инструкций по выполнению примеров помещены в следующие разделы.

Подготовка

В этом разделе рассказывается, чего ожидать от примера, и описывается, как настроить программное обеспечение или выполнить предварительные настройки, необходимые для его выполнения.

Как это делается...

В этом разделе приведены шаги, необходимые для выполнения примера.

Дополнительно

Этот раздел содержит дополнительную информацию о примере, чтобы вы лучше ориентировались.