

Содержание

Предисловие	6
Об авторе	6
О рецензентах	6
Введение	6
Для кого предназначена эта книга	15
Что охватывает эта книга	16
Как получить максимальную отдачу от этой книги	18
Когда читать данную книгу	19
Скачивание файлов с примерами кода	21
Условные обозначения	21
Комментарии переводчика	22
Глава 1. Начало работы с платформой TensorFlow Mobile	22
Настройка платформы TensorFlow	26
Настройка TensorFlow в MacOS	28
Настройка TensorFlow в Ubuntu с поддержкой GPU	29
Настройка среды разработки Xcode	34
Настройка среды разработки Android Studio	35
TensorFlow Mobile против TensorFlow Lite	37
Выполнение примеров приложений TensorFlow для iOS	38
Выполнение примеров приложений TensorFlow для Android	39
Резюме	40
Глава 2. Классифицирование изображений с помощью трансферного обучения	6
Трансферное обучение – что это такое и почему	43
Вторичная тренировка с использованием модели Inception v3	44
Вторичная тренировка с использованием моделей MobileNet	53
Использование вторично натренированных моделей в примере приложения для iOS	56
Использование вторично натренированных моделей в примере приложения для Android	58
Добавление платформы TensorFlow в свое собственное приложение для iOS	60

Добавление платформы TensorFlow в свое собственное приложение для iOS на языке Objective-C.....	60
Добавление платформы TensorFlow в свое собственное приложение для iOS на языке Swift	66
Добавление платформы TensorFlow в свое собственное приложение для Android.....	73
Резюме.....	78

Глава 3. Обнаружение и локализация объектов.....6

Обнаружение объектов – краткий обзор	80
Настройка API TensorFlow обнаружения объектов.....	82
Быстрая установка и пример	83
Использование предварительно натренированных моделей	84
Вторичная тренировка моделей на основе SSD-MobileNet и более быстрого RCNN-детектора	88
Использование моделей обнаружения объектов в iOS	94
Ручная сборка библиотек TensorFlow для iOS	94
Использование библиотек TensorFlow для iOS в приложении	95
Добавление функционала обнаружения объектов в приложение для iOS.....	98
Использование YOLO2 – еще одной модели обнаружения объектов	104
Резюме.....	108

Глава 4. Трансформирование рисунков с помощью художественных стилей.....6

Нейронный перенос стиля – краткий обзор.....	110
Тренировка моделей быстрого нейронного переноса стиля	111
Использование моделей быстрого нейронного переноса стиля в iOS	114
Добавление и тестирование с моделями быстрого нейронного переноса.....	115
Анализ программного кода iOS с использованием моделей быстрого нейронного переноса	117
Использование моделей быстрого нейронного переноса стиля в Android....	119
Использование многостилевой модели TensorFlow Magenta в iOS.....	125
Использование многостилевой модели TensorFlow Magenta в Android	133
Резюме.....	137

Глава 5. Понимание простых речевых команд.....6

Распознавание речи – краткий обзор	140
Тренировка простой модели распознавания команд.....	142
Использование простой модели распознавания речи в Android	146
Создание нового приложения с использованием модели.....	147
Вывод результатов распознавания.....	152

Использование простой модели распознавания речи в iOS на языке Objective-C.....	155
Создание нового приложения с использованием модели.....	155
Исправление ошибок загрузки модели с помощью файла tf_op_files.txt	161
Использование простой модели распознавания речи в iOS на языке Swift... ..	162
Резюме.....	166

Глава 6. Описание изображений на естественном языке..... 6

Аннотирование изображений – как оно работает	168
Тренировка и замораживание модели аннотирования изображений	170
Тренировка и тестирование генерирования аннотаций	170
Замораживание модели аннотирования изображений.....	174
Трансформация и оптимизация модели аннотирования изображений	180
Исправление ошибок в трансформированных моделях.....	180
Оптимизация трансформированной модели	184
Использование модели аннотирования изображений в iOS.....	185
Использование модели аннотирования изображений в Android	195
Резюме.....	201

Глава 7. Распознавание рисунков с помощью CNN- и LSTM-сетей..... 6

Классификация рисунков – как это работает	204
Тренировка, предсказание и подготовка модели классификации рисунков.....	206
Тренировка модели классификации рисунков.....	207
Предсказание с помощью модели классификации рисунков	208
Подготовка модели классификации рисунков	210
Использование модели классификации рисунков в iOS	215
Сборка пользовательской библиотеки TensorFlow для iOS	216
Разработка приложения для iOS с использованием модели	217
Использование модели классификации рисунков в Android	225
Сборка пользовательской библиотеки TensorFlow для Android	225
Разработка приложения для Android с целью применения модели	227
Резюме.....	236

Глава 8. Предсказание биржевой цены с помощью RNN-сети..... 6

RNN-сеть и предсказание биржевой цены – что это такое и как это делается.....	238
Использование API TensorFlow RNN для предсказания биржевой цены.....	240
Тренировка RNN-модели в TensorFlow	241
Тестирование модели TensorFlow RNN	245

Использование API RNN LSTM библиотеки Keras для предсказания биржевой цены	247
Тренировка LSTM-модели в библиотеке Keras	248
Тестирование модели Keras RNN	251
Выполнение моделей TensorFlow и Keras в iOS	254
Выполнение моделей TensorFlow и Keras в Android	260
Резюме	265

Глава 9. Генерирование и улучшение изображений

с помощью GAN-сети	6
GAN-сеть – что это такое и почему	267
Построение и тренировка GAN-моделей с помощью TensorFlow	269
Базовая GAN-модель генерирования рукописных цифр	269
Продвинутая GAN-модель улучшения разрешающей способности изображения	272
Использование GAN-моделей в iOS	277
Использование базовой GAN-модели	279
Использование продвинутой GAN-модели	281
Использование GAN-моделей в Android	284
Использование базовой GAN-модели	287
Использование продвинутой GAN-модели	289
Резюме	291

Глава 10. Создание мобильного игрового

AlphaZero-подобного приложения	6
Алгоритм AlphaZero – как он работает?	294
Тренировка и тестирование AlphaZero-подобной модели для игры «Четыре в ряд»	296
Тренировка модели	297
Тестирование модели	301
Анализ программного кода построения модели	304
Заморозка модели	305
Использование модели игры «Четыре в ряд» в iOS	306
Использование модели игры «Четыре в ряд» в Android	320
Резюме	332

Глава 11. Применение платформ TensorFlow Lite и Core ML на мобильных устройствах

на мобильных устройствах	6
Платформа TensorFlow Lite – краткий обзор	334
Использование платформы TensorFlow Lite в iOS	335
Выполнение примеров приложений TensorFlow Lite для iOS	335
Использование готовой модели TensorFlow Lite в iOS	337

Использование вторично натренированной модели TensorFlow для платформы TensorFlow Lite в iOS.....	342
Использование пользовательской модели TensorFlow Lite в iOS.....	343
Использование платформы TensorFlow Lite в Android	345
Платформа Core ML для iOS – краткий обзор.....	349
Использование платформы Core ML с машинным обучением на основе библиотеки Scikit-Learn	350
Построение и конвертирование моделей Scikit-Learn.....	350
Использование конвертированных в формат Core ML моделей в iOS.....	352
Использование платформы Core ML с Keras и TensorFlow.....	354
Резюме.....	359

Глава 12. Разработка приложений TensorFlow на компьютере Raspberry Pi 361

Настройка компьютера Raspberry Pi и приведение его в движение.....	362
Настройка материнской платы Raspberry Pi	364
Приведение компьютера Raspberry Pi в движение	367
Настройка платформы TensorFlow на компьютере Raspberry Pi.....	369
Распознавание изображений и речевое воспроизведение текста.....	371
Распознавание звука и движение робота	374
Самообучение с подкреплением на компьютере Raspberry Pi.....	377
Описание симулируемой среды CartPole.....	378
Начало с простой интуитивно понятной линии поведения	383
Использование нейронных сетей для построения более оптимальной линии поведения.....	385
Резюме.....	393

Послесловие.....	6
-------------------------	----------

Предисловие

В последнее десятилетие происходит бурное развитие как машинного обучения, так и смартфонов; сегодня эти технологии наконец сливаются воедино, в результате чего образуется невероятное разнообразие приложений, от которых всего несколько лет назад вы бы отмахнулись, как от научной фантастики далекого будущего. Только подумайте: вы уже привыкли разговаривать с телефоном, спрашивать у него дорогу или поручать ему назначать время встречи в вашем графике. Фотокамера телефона отслеживает лица и распознает объекты. Игры становятся все интереснее и сложнее, поскольку боты становятся умнее и умнее. И бесчисленные программные приложения используют под капотом ту или иную форму искусственного интеллекта все менее очевидными способами, например рекомендуя контент, который вам понравится, предугадывая ваш следующий маршрут, чтобы сообщить, когда следует уйти, предлагая, что печатать дальше, и т.д.

До недавнего времени весь интеллект располагался на стороне сервера, а это означало, что пользователь должен был оставаться подключенным к интернету, в идеале, быстрым и стабильным соединением. Неизбежные задержки и сбои в обслуживании были тормозом для многих приложений. Но сегодня интеллект находится прямо у вас на ладони благодаря огромным аппаратным улучшениям и усовершенствованным библиотекам машинного обучения.

Самое главное, что эти технологии теперь полностью демократизированы: практически любой инженер-программист может научиться программировать интеллектуальное мобильное приложение на основе глубоких нейронных сетей, используя TensorFlow, мощную библиотеку глубокого обучения компании Google с открытым исходным кодом. Замечательная и уникальная книга Джеффа Танга покажет вам, как разрабатывать локальные приложения на основе TensorFlow для iOS, Android и Raspberry Pi, проведя вас через множество конкретных примеров с пошаговыми инструкциями и с большим трудом добытыми советами по устранению неполадок: начиная с классифицирования изображений, обнаружения объектов, аннотирования изображений и распознавания рисунков и заканчивая распознаванием речи, предсказанием временных рядов, генеративно-состязательными сетями, самообучением на основе максимизации вознаграждения и даже построением интеллектуальных игр с использованием AlphaZero – усовершенствованной технологии, построенной поверх программы AlphaGo, которая одержала верх над чемпионами мира по игре го Ли Седодем и Ке Цзе.

Эта книга станет очень популярной, поскольку в ней затронута чрезвычайно важная тема, о которой трудно получить хорошую достоверную информацию. Так что пора засучить рукава. Впереди вас ждет захватывающее путешествие! Ну, и какое же интеллектуальное мобильное приложение вы будете строить?

*Орелиен Жерон,
экс-лидер команды классифицирования видеороликов YouTube и автор книги Hands-On Machine Learning with Scikit-Learn and TensorFlow («Практическое машинное обучение с помощью Scikit-Learn и TensorFlow», O'Reilly, 2017)*

Париж, 11 мая 2018 г.

Об авторе

Джефф Танг влюбился в классический ИИ более двух десятилетий назад. После получения степени магистра наук в области информатики он в течение 2 лет работал над машинным переводом и затем, чтобы пережить долгую зиму ИИ¹, трудился над корпоративными, голосовыми, веб- и мобильными приложениями в стартапах, компаниях AOL, Baidu и Qualcomm. Он разработал пользующееся исключительным спросом приложение iOS с миллионами скачиваний и был удостоен звания ведущего разработчика Android Market компанией Google. Он вновь присоединился к современному ИИ в 2015 году, зная наперед, что ИИ будет его страстью и приверженностью в течение следующих двух десятилетий. Одна из его любимых тем – сделать ИИ доступным в любое время в любом месте, в результате чего и появилась настоящая книга.

Хотел бы поблагодарить Лариссу Пинто за то, что подсказала мне идею книги, Флавиана Ваза и Ахила Наира за отзывы во время редактирования черновика. Большое спасибо Питу Уордену, руководителю направления мобильных приложений TensorFlow в Google, за его помощь до и после того, как он стал техническим рецензентом книги, и Амите Капур, еще одному рецензенту книги, которая предоставила мне ценные отзывы. Особая благодарность Орелиену Жерону, автору бестселлера по практическому МО, за любезные ответы на все мои электронные письма, свои соображения и предисловие к книге, для которого он нашел время, несмотря на свое упакованное расписание, – Мерси боку, Орелиен.

Я также искренне ценю понимание и поддержку всей моей семьи, помимо Лизы и Воли, во время курортного сезона и все месяцы, пока я как сумасшедший день и ночь работал над книгой, – спасибо Эми, Анне, Дженни, Софии, Марку, Сэнди и Бену.

¹ Зима ИИ – это период сокращения фондирования и падения интереса к исследованиям в области искусственного интеллекта. См. https://en.wikipedia.org/wiki/AI_winter. – Прим. перев.

О рецензентах

Пит Уорден является техническим руководителем направления мобильных и встроенных приложений TensorFlow в команде Google Brain¹.

Амита Капур является доцентом кафедры электроники Университета Дели SRCASW с 1996 года. Получатель престижной стипендии DAAD для продолжения части своей исследовательской работы в Технологическом институте Карлсруэ, Германия. Награждена лучшей презентационной наградой на международной конференции Photonics 2008. Член различных профессиональных организаций и имеет более 40 публикаций в международных журналах и конференциях. В настоящее время ее научные направления включают МО, ИИ, нейронные сети, робототехнику, буддизм и этику в ИИ.

¹ Google Brain – это исследовательский проект Google по изучению искусственного интеллекта на основе глубокого обучения. См. https://ru.wikipedia.org/wiki/Google_Brain. – *Прим. перев.*

Введение

Искусственный интеллект (ИИ), симулирование человеческого интеллекта в компьютерах, имеет долгую историю. С момента своего официального рождения в 1956 году ИИ пережил несколько периодов подъема и спада. Нынешнее возрождение интереса к ИИ, или новая революция ИИ, началось в 2012 году вместе с инновационным прорывом в области глубокого обучения, отрасли машинного обучения, которая сегодня стала самым популярным направлением ИИ, после того как **глубокие сверточные нейронные сети** (Deep Convolutional Neural Network, DCNN) одержали верх в конкурсной инициативе ImageNet по распознаванию изображений (Large Scale Visual Recognition Challenge, ILSVRC) с частотой ошибок всего 16,4% по сравнению с идущей на втором месте не DCNN-моделью с частотой ошибок 26,2%. Начиная с 2012 года улучшенные модели на основе глубокой сверточной нейронной сети выигрывают конкурс ImageNet ежегодно, а технология глубокого обучения применяется ко многим сложным задачам ИИ, находящимся за пределами компьютерного зрения, таким как распознавание речи, машинный перевод и игра в го, в результате чего происходил один инновационный прорыв за другим. В марте 2016 года программа AlphaGo компании Google DeepMind, построенная с применением глубокого самообучения на основе максимизации вознаграждения, победила 18-кратного чемпиона мира Ли Седоля со счетом 4:1. На ежегодном фестивале разработчиков Google I/O в 2017 году компания Google объявила, что они переходят от мобильного мира (mobile-first world), то есть технологий, предназначенных в первую очередь для мобильных устройств, к миру ИИ. Другие ведущие компании, такие как Amazon, Apple, Facebook и Microsoft, продолжали вкладывать значительные средства в ИИ и запускать множество продуктов на базе ИИ.

TensorFlow – это платформа Google с открытым исходным кодом для создания приложений машинного обучения. С момента своего первого выпуска в ноябре 2015 года, когда уже существовало несколько популярных платформ глубокого обучения с открытым исходным кодом, платформа TensorFlow быстро, менее чем за 2 года, стала самым популярным каркасом глубокого обучения с открытым исходным кодом. С тех пор еженедельно строились новые модели TensorFlow для решения всевозможных задач, которые требуют человеческого или даже сверхчеловеческого интеллекта, были опубликованы десятки книг по TensorFlow, и еще больше появилось сетевых журналов, учебных пособий, курсов и видеороликов, посвященных TensorFlow. Популярность ИИ и TensorFlow совершенно очевидна, но для чего нужна еще одна книга с термином «TensorFlow» в названии?

Эта книга уникальна, и уникальность ее в том, что она объединяет ИИ, приводимый в действие платформой TensorFlow, с мобильным устройством, соединяя мир яркого будущего с миром самого процветающего настоящего.

За последнее десятилетие мы все были свидетелями революции в области смартфонов iOS и Android и испытали ее на себе. И при этом мы находимся лишь в начале революции в области ИИ, которая, вероятно, окажет еще более глубокое влияние на мир вокруг нас. Что может быть лучше, чем тема, которая объединяет лучшее из двух миров, книга, которая показывает, как создавать приложения ИИ с поддержкой TensorFlow на мобильном устройстве, в любое время и в любом месте?

Разумеется, вы можете создавать приложения с элементами ИИ, используя множество существующих облачных API искусственного интеллекта, и иногда это имеет смысл. Вместе с тем запуск приложений с элементами ИИ полностью на мобильных устройствах имеет свои преимущества, которые заключаются в том, что вы можете запускать приложения, даже если отсутствует подключение к сети, когда вы не можете обмениваться данными с облачным сервером либо когда пользователи не хотят отправлять данные со своих мобильных устройств кому-либо еще.

Надо признать также и то, что среди ресурсов проекта TensorFlow с открытым исходным кодом уже есть несколько примеров приложений для iOS и Android, которые могут помочь вам начать работу с мобильной платформой TensorFlow. Однако если вы когда-либо пытались запускать захватывающую модель с поддержкой TensorFlow, которая будет поражать вас своими способностями на своем устройстве iOS или Android, то вы, скорее всего, знаете, что это сопряжено со множеством преград, которые следует преодолеть, прежде чем вы сможете увидеть модель, которая будет успешно работать на вашем мобильном устройстве.

Эта книга поможет сэкономить вам уйму времени и усилий и покажет, как решать все самые распространенные проблемы, с которыми вы можете столкнуться при запуске моделей TensorFlow на мобильном устройстве. В настоящей книге вы увидите, как будут построены с нуля более 10 полноценных программных приложений TensorFlow для iOS и Android, выполняющих множество захватывающих моделей с поддержкой TensorFlow, включая новейшие и самые крутые – модели на основе генеративно-сопоставительной сети (Generative Adversarial Network, GAN) и алгоритма AlphaZero.

Для кого предназначена эта книга

Настоящая книга предназначена для тех читателей, которые являются разработчиками приложений для iOS и/или Android и интересуются построением и вторичной тренировкой других моделей TensorFlow и запуском их в своих мобильных приложениях, а также тех, кто является разработчиком приложений с использованием платформы TensorFlow и хочет запускать новые удивительные модели TensorFlow на мобильных устройствах. Если вы заинтересованы в платформах TensorFlow Lite, Core ML или TensorFlow на компьютере Raspberry Pi, то также извлечете пользу из данной книги.

ЧТО ОХВАТЫВАЕТ ЭТА КНИГА

В главе 1 «*Начало работы с платформой TensorFlow Mobile*» рассматривается настройка платформы TensorFlow в Mac и Ubuntu, а также графического процессора NVIDIA GPU в Ubuntu. Помимо этого, также рассматривается установка сред разработки Xcode и Android Studio. Мы обсудим разницу между платформами TensorFlow Mobile и TensorFlow Lite и рассмотрим случаи, когда вы должны их использовать. Наконец, мы покажем вам, как запускать примеры приложений для iOS и Android с поддержкой платформы TensorFlow.

В главе 2 «*Классифицирование изображений с помощью трансферного обучения*» рассматривается понятие трансферного обучения и дается объяснение, почему вы должны его использовать, демонстрируется, как вторично тренировать модели Inception v3 и MobileNet для более точного и быстрого распознавания пород собак и как использовать вторично натренированные модели в примерах приложений для iOS и Android. Затем мы покажем вам, как добавлять поддержку платформы TensorFlow в свое собственное приложение для iOS на языках программирования Objective-C и Swift и в приложение для Android для распознавания пород собак.

В главе 3 «*Обнаружение и локализация объектов*» дается краткий обзор технологии обнаружения объектов, а затем показывается настройка API Tensorflow обнаружения объектов и его применение для вторичной тренировки моделей SSD-MobileNet и моделей Faster RCNN. Мы также покажем вам, как использовать модели, которые использовались в примере приложения TensorFlow для Android, в приложении для iOS, выполнив сборку библиотеки TensorFlow для iOS вручную с целью поддержки операций TensorFlow, отличных от стандартных. Наконец, мы покажем, как тренировать YOLO2, еще одну популярную модель обнаружения объектов, которая также используется в примере приложения TensorFlow для Android, и как ее использовать в приложении для iOS.

В главе 4 «*Трансформирование рисунков с помощью художественных стилей*» сначала дается обзор технологии нейронного переноса художественного стиля, получившей ускоренное развитие в последние несколько лет. Затем в ней демонстрируется процедура тренировки модели быстрого нейронного переноса стиля и их применения в приложениях для iOS и Android. После этого мы расскажем, как использовать многостилевую модель TensorFlow Magenta в своих собственных приложениях для iOS и Android, позволяющую легко создавать удивительные художественные стили.

В главе 5 «*Понимание простых речевых команд*» дается описание технологии распознавания речи и демонстрируется тренировка простой модели распознавания речевых команд. Затем мы покажем вам, как использовать такую модель в Android, а также в iOS с применением языков программирования Objective-C и Swift. Мы также дадим ряд советов в отношении того, как исправлять возможные ошибки загрузки и запуска модели на мобильном устройстве.

В главе 6 «*Описание изображений на естественном языке*» дается описание принципа работы технологии аннотирования изображений. Затем в ней демонстрируется процедура тренировки и заморозки модели аннотирования изображений в TensorFlow. Далее мы обсудим вопрос конвертации и оптимизации многосложной модели с целью ее подготовки к работе на мобильных устройствах. Наконец, мы предложим полноценные приложения для iOS и Android, в которых данная модель используется для генерирования описания изображений на естественном языке.

В главе 7 «*Распознавание рисунков с помощью CNN- и LSTM-сетей*» объясняется принцип работы технологии классифицирования рисунков. Данная глава посвящена подготовке модели, ее тренировке и выведению предсказаний. Затем мы покажем, как создать еще одну пользовательскую библиотеку TensorFlow в iOS для использования данной модели в забавном приложении для iOS, выполняющем распознавание набросков. Наконец, мы покажем вам, как создавать пользовательскую библиотеку TensorFlow для Android, которая позволит исправлять ошибку загрузки новой модели, а затем продемонстрируем использование модели в своем собственном приложении для Android.

В главе 8 «*Предсказание биржевой цены с помощью RNN-сети*» дан пошаговый анализ технологии RNN-сети и продемонстрировано ее применение для предсказания биржевой цены. Затем мы покажем, как строить RNN-модель предсказания биржевой цены с помощью API TensorFlow и как строить LSTM RNN-модель с более простым в использовании API библиотеки Keras для достижения той же самой цели. Мы проведем тестирование этих моделей и посмотрим, смогут ли они превзойти случайную стратегию покупки или продажи. Наконец, мы покажем вам, как запускать модели TensorFlow и Keras в приложениях для iOS и Android.

В главе 9 «*Генерирование и улучшение изображений с помощью GAN-сети*» дается обзор технологии генеративно-сопоставительной сети (GAN-сети) и объясняется, почему она имеет такой большой потенциал. Затем в данной главе будет дано краткое описание процедуры построения и тренировки базовой GAN-модели, которая может использоваться для создания человекоподобных рукописных цифр, а также более продвинутой модели, которая способна улучшать качество изображения с низким разрешением. Наконец, мы расскажем, как использовать обе эти модели на основе GAN-сети в приложениях для iOS и Android.

В главе 10 «*Создание мобильного игрового AlphaZero-подобного приложения*» сначала дается описание работы новейшей и популярной программы AlphaZero, затем демонстрируется процедура тренировки и тестирования AlphaZero-подобной модели для игры в простую, но увлекательную игру под названием «Четыре в ряд» с использованием библиотеки Keras и платформы TensorFlow в качестве бэкенда. Далее мы покажем вам полноценные приложения для iOS и Android, которые позволяют использовать эту модель и играть в игру «Четыре в ряд» на ваших мобильных устройствах.

В главе 11 «*Применение платформ TensorFlow Lite и Core ML на мобильном устройстве*» демонстрируется работа платформы TensorFlow Lite, а затем показывается применение в iOS готовой модели TensorFlow, вторично натренированной модели TensorFlow для платформы TensorFlow Lite и пользовательской модели TensorFlow Lite. Мы также покажем вам, как использовать платформу TensorFlow Lite в Android. После этого мы дадим краткий обзор платформы Core ML компании Apple и покажем, как использовать платформу Core ML со стандартным машинным обучением на основе библиотеки Scikit-Learn. Наконец, мы рассмотрим пример использования платформы Core ML с платформой TensorFlow и библиотекой Keras.

В главе 12 «*Разработка приложений TensorFlow на компьютере Raspberry Pi*» сначала рассматривается настройка компьютера Raspberry Pi и приведение его в движение, а также настройка платформы TensorFlow на компьютере Raspberry Pi. Затем мы расскажем, как использовать модели TensorFlow распознавания изображений и звука, а также API речевого воспроизведения текста (TTS) и API движения робота для создания робота Raspberry Pi, который может двигаться, видеть, слушать и говорить. Наконец, мы подробно обсудим вопрос использования библиотеки OpenAI Gym и платформы TensorFlow для создания и тренировки с нуля мощной нейросетевой модели линии поведения на основе самообучения с подкреплением в симулируемой среде, позволяющей роботу учиться сохранять равновесие.

КАК ПОЛУЧИТЬ МАКСИМАЛЬНУЮ ОТДАЧУ ОТ ЭТОЙ КНИГИ

Мы рекомендуем начать с чтения первых четырех глав по порядку, а также с запуска сопутствующих приложений для iOS и Android, доступных из репозитория исходного кода книги по адресу <http://github.com/jeffxtang/mobiletfbook>. Они позволят вам обеспечить наличие сред разработки, настроенных для разработки мобильных приложений с поддержкой TensorFlow, и сведений о том, как интегрировать TensorFlow в свои собственные приложения для iOS и/или Android. Если вы являетесь разработчиком iOS, то также узнаете, как использовать языки программирования Objective-C или Swift с платформой TensorFlow, когда и как использовать модуль TensorFlow либо пользовательскую библиотеку TensorFlow для iOS ручной сборки.

Затем, если вам необходимо выполнить ручную сборку пользовательской библиотеки TensorFlow для Android, перейдите к главе 7 «*Распознавание рисунков с помощью CNN- и LSTM-сетей*», и если вы хотите узнать, как использовать в своем мобильном приложении модель Keras, то обратитесь к главе 8 «*Предсказание биржевой цены с помощью RNN-сети*» и главе 10 «*Создание мобильного игрового AlphaZero-подобного приложения*».

Если вы больше заинтересованы в платформе TensorFlow Lite или Core ML, то прочтите главу 11 «*Применение платформ TensorFlow Lite и Core ML на мобильном устройстве*», а если же вы больше всего заинтересованы в платформе

TensorFlow на компьютере Raspberry Pi или реализацией самообучения с подкреплением в платформе TensorFlow, то перейдите к главе 12 «*Разработка приложений TensorFlow на компьютере Raspberry Pi*».

Кроме этого, прежде чем сразу переходить к подробному изучению реализаций моделей, вы можете просмотреть главы 5–10, чтобы узнать, как выполнять тренировку различных видов моделей – на основе CNN, RNN, LSTM, GAN и AlphaZero – и как их использовать на мобильных устройствах, возможно, запуская приложения для iOS и/или Android, приводимые в каждой главе. Кроме того, вы можете перейти непосредственно к любой главе с интересующей вас моделью; просто имейте в виду, что более поздняя глава может ссылаться на более раннюю главу относительно повторяющихся подробностей, таких как шаги добавления в свое приложение пользовательской библиотеки TensorFlow для iOS или исправление некоторых ошибок загрузки либо запуска модели путем ручной сборки пользовательской библиотеки TensorFlow. Так или иначе, будьте уверены, что вы не потеряетесь. По крайней мере, мы сделали все возможное, чтобы обеспечить вам удобное пошаговое руководство с периодическими ссылками на некоторые шаги из предыдущих инструкций, с тем чтобы помочь вам избежать всевозможных ловушек, с которыми вы можете столкнуться при создании мобильных приложений с поддержкой платформы TensorFlow, а также просто чтобы избежать повторов.

Когда читать данную книгу

В последние годы ИИ, точнее его самая популярная область – машинное обучение, а если еще точнее, его самая популярная подобласть – глубокое обучение, получили ускоренное развитие. Новые выпуски платформы TensorFlow, поддерживаемые компанией Google и самым популярным сообществом разработчиков на данной платформе из всех платформ машинного обучения с открытым исходным кодом, также выходили с более высокой скоростью. Когда мы приступили к написанию данной книги в декабре 2017 года, последний релиз платформы TensorFlow был 1.4.0, выпущенный 2 ноября 2017 года, и после этого версия 1.5.0 была выпущена 26 января 2018 года, версия 1.6.0–28 февраля 2018 года, версия 1.7.0–29 марта 2018 года, и версия 1.8.0–27 апреля 2018 года. Весь приводимый в книге программный код для iOS и Android и на языке Python был протестирован со всеми этими версиями платформы TensorFlow. Добавим, что к тому времени, когда вы будете читать эту книгу, последней версией TensorFlow, вероятно, будет уже версия выше 1.8.0.

Как оказалось, вам не нужно особо переживать по поводу частого выхода новых версий платформы TensorFlow; приводимый в книге программный код, скорее всего, будет легко работать в последних версиях данной платформы. Во время тестирования нами программных приложений на платформах TensorFlow 1.4, 1.5, 1.6, 1.7 и 1.8 мы не вносили никаких изменений в про-

граммный код. Вполне вероятно, что в более поздней версии по умолчанию будет поддерживаться еще больше операций платформы TensorFlow, и поэтому вам не придется выполнять ручную сборку пользовательской библиотеки TensorFlow, либо вы сможете выполнить ручную сборку пользовательской библиотеки TensorFlow более простым способом.

Конечно, нет никакой гарантии, что в будущих версиях платформы TensorFlow весь программный код будет работать без каких-либо изменений, но с учетом всех подробных инструкций и советов по устранению неполадок, изложенных в данной книге, независимо от того, когда вы читаете эту книгу, сейчас или по прошествии нескольких месяцев, у вас не должно возникнуть шероховатостей при ее чтении и запуске приводимых в ней приложений с помощью платформы TensorFlow 1.4–1.8 или более поздней версии.



Несмотря на то что в целях публикации данной книги мы в определенный момент вынуждены были остановиться на конкретной версии платформы TensorFlow, мы продолжим выполнять тестовый прогон всего программного кода книги после выхода каждой новой крупной версии платформы TensorFlow, а также соответствующим образом обновлять программный код и результаты тестов в репозитории исходного кода книги по адресу <http://github.com/jeffxtang/mobiletfbook>. Если у вас возникли вопросы по поводу программного кода или данной книги, то вы также можете отправить вопрос непосредственно в репозиторий.

Еще одной проблемой является выбор между платформами TensorFlow Mobile и TensorFlow Lite. В большинстве глав данной книги обсуждается использование платформы TensorFlow Mobile (главы с 1 по 10). Может быть, в будущем платформа TensorFlow Lite станет мобильной альтернативой настольной платформе TensorFlow, но пока, согласно конференции разработчиков Google I/O 2018, эта платформа находится в предварительной версии – именно поэтому компания Google ожидает, что вы будете «применять платформу TensorFlow Mobile для большинства производственных задач». Даже после того, как платформа TensorFlow Lite будет выпущена официально, по данным компании Google, «в обозримом будущем платформа TensorFlow Mobile не собирается уходить со сцены» – на самом деле с учетом последней версии TensorFlow 1.8.0, которую мы протестировали до публикации книги, мы обнаружили, что применение платформы TensorFlow Mobile стало еще проще.

Если же, наконец, наступит тот день, когда платформа TensorFlow Lite полностью заменит собой платформу TensorFlow Mobile во всех случаях использования, располагая большей производительностью и меньшим размером, то навыки, которые вы приобретете из этой книги, только лучше подготовят вас к этому дню. В то же время, прежде чем это непредвиденное будущее наступит, вы имеете возможность прочитать книгу и узнать, как применять ее более солидный аналог, платформу TensorFlow Mobile, для запуска всех этих удивительных и мощных моделей с поддержкой TensorFlow в мобильных приложениях.

СКАЧИВАНИЕ ФАЙЛОВ С ПРИМЕРАМИ КОДА

Файлы с примерами можно скачать с вашего аккаунта по адресу <http://www.packtpub.com/> для всех книг издательства Packt Publishing, которые вы приобрели. Если вы купили эту книгу в другом месте, то можно посетить <http://www.packtpub.com/support> и зарегистрироваться там, чтобы получить файлы прямо по электронной почте.

Скачать файлы с примерами можно, выполнив следующие шаги.

1. Войдите на наш веб-сайт или зарегистрируйтесь там, используя ваш адрес электронной почты и пароль.
2. Наведите указатель мыши на вкладку SUPPORT вверху страницы.
3. Щелкните по разделу Code Downloads & Errata, посвященному примерам программного кода и опечаткам.
4. Введите название книги в поле поиска.

Скачав файл, пожалуйста, убедитесь, что вы распаковали или извлекли папку, воспользовавшись последней версией указанных ниже архиваторов:

- WinRAR / 7-Zip для Windows;
- Zipeg / iZip / UnRarX для Mac OS;
- 7-Zip / PeaZip для Linux.

Помимо этого, комплект примеров программного кода, прилагаемый к данной книге, размещен на GitHub в разделе Packt по адресу <https://github.com/PacktPublishing/Intelligent-Mobile-Projects-with-TensorFlow>. В случае обновления программного кода он будет обновлен в существующем репозитории GitHub. Мы также располагаем другими комплектами примеров программного кода, которые можно выбрать из нашего богатого каталога книг и видеороликов, предлагаемого на странице <https://github.com/PacktPublishing/>. Можете убедиться сами!

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

В этой книге используется ряд условных обозначений.

КодВТексте: указывает кодовые слова в тексте, имена таблиц базы данных, имена папок, имена файлов, расширения файлов, имена путей, фиктивные URL-адреса, ввод данных пользователем и дескрипторы Twitter. Вот пример: «Установить библиотеки `matplotlib`, `pillow`, `lxml` и `ipython` в Ubuntu или Mac, которые вы можете запустить».

Блок кода выглядит следующим образом:

```
syntax = "proto2";
package object_detection.protos;
message StringIntLabelMapItem {
  optional string name = 1;
  optional int32 id = 2;
  optional string display_name = 3;
};
```

```
message StringIntLabelMap {  
  repeated StringIntLabelMapItem item = 1;  
};
```

Любой ввод или вывод из командной строки записывается следующим образом:

```
sudo pip install pillow  
sudo pip install lxml  
sudo pip install jupyter  
sudo pip install matplotlib
```

Полужирный шрифт: обозначает новый термин, важное слово или слова, отображаемые на экране. Например, слова в меню или диалоговых окнах отображаются в тексте следующим образом. Вот пример: «А теперь давайте выберем опцию **«Улучшить изображение»**, и вы увидите результат».



Данный элемент обозначает предупреждение или предостережение.



Данный элемент обозначает подсказку или совет.

КОММЕНТАРИИ ПЕРЕВОДЧИКА

В центре внимания машинного обучения и его подобласти, глубокого обучения, находится обучающаяся система, то есть система, способная с течением времени приобретать новые знания и улучшать свою работу, используя поступающую информацию¹. В зарубежной специализированной литературе для *передачи* знаний и *приобретения* знаний существуют отдельные термины – *train* (*натренировать*, обучить) и *learn* (*выучить*, обучиться). Приведенное ниже предложение из главы 2 настоящей книги четко это демонстрирует:

It'd take us many weeks of training of a modern CNN from scratch to learn all the weights.

Нам пришлось бы тренировать современную CNN-сеть на протяжении многих недель с нуля, пока она не заучит все веса,

где *training* (тренировка) – это работа, которую выполняет исследователь-проектировщик для получения обучившейся модели, в основе которой лежит обучающийся алгоритм, по сути искатель минимумов (или максимумов) для надлежащим образом сформулированных функций, а *learning* (самообучение, заучивание) – это работа, которую выполняет алгоритм-ученик по приобретению новых знаний или изменению и закреплению существующих знаний

¹ См. https://ru.wikipedia.org/wiki/Обучающаяся_система, а также <https://bigenc.ru/mathematics/text/1810335>. – *Прим. перев.*

и поведения¹. Когда же в русской спецлитературе используется термин «обучение», то он несет в себе двусмысленность, потому что под ним может подразумеваться и передача знаний, и получение знаний одновременно, как, например, в случае с термином «машинное обучение», который может означать и тренировку алгоритмических машин, и способность таких машин обучаться, что нередко вносит путаницу и терминологический разброд в переводной литературе при решении дилеммы «training-learning», в то время как появление в зарубежной технической литературе термина learning в любом виде подразумевает исключительно второе – *самообучение, заучивание* алгоритмом весов и других параметров. Отсюда вытекает одно важное следствие: английский термин machine learning обозначает *приобретение знаний алгоритмической машиной*, а следовательно, более соответствовать оригиналу будет термин *машинное самообучение или автоматическое обучение*. Весомым аргументом в пользу этого термина является и то, что с начала 60-х и до середины 80-х годов XX столетия в ходу был похожий термин – «обучающиеся машины». Проблематика обучающихся и самопроизводящихся машин изучалась в работах А. Тьюринга «*Может ли машина мыслить?*», 1960 (обучающиеся машины), К. Шеннона «*Работы по теории информации и кибернетике*» (самовоспроизводящиеся машины), Н. Винера «*Кибернетика, или Управление и связь в животном и машине*», 1961, Н. Нильсона «*Обучающиеся машины*», 1974, и Цыпкина Я.З. «*Основы теории обучающихся систем*», 1970.

В настоящем переводе далее за основу принят зарубежный подход, который неизбежно привел к некоторой корректировке терминологии. Применяемые в машинном обучении и глубоком обучении алгоритмы, модели и системы переведены как *обучающиеся, машинно-обучающиеся и глубоко обучающиеся*. То есть акцент делается не на классификации алгоритма в соответствующей иерархии, а на его характерном свойстве. Далее, методы, которые реализуются в обучающихся алгоритмах, переведены как *методы самообучения* (ср. методы обучения). Как известно, эти методы делятся на три широкие категории. Следуя принципу бритвы Оккама, они переведены как методы контролируемого самообучения (ср. обучение с учителем), методы неконтролируемого самообучения (ср. обучение без учителя) и методы самообучения на основе максимизации вознаграждения (ср. обучение с подкреплением). Последний термин обусловлен тем, что в его основе лежит алгоритм, который «*учится максимизировать некое понятие вознаграждения*», получаемого за правильно выполненное действие². Среди многих гиперпараметров, которые позволяют настроить работу обучающегося алгоритма, имеется rate of learning, который переведен как *скорость заучивания* (ср. темп обучения).

¹ См. <http://www.basicknowledge101.com/subjects/learningstyles.html#diy>. – *Прим. перев.*

² См. https://en.wikipedia.org/wiki/Reinforcement_learning. – *Прим. перев.*

Глава 1

Начало работы с платформой TensorFlow Mobile

Данная глава посвящена настройке среды разработки для создания всех тех приложений для iOS или Android с поддержкой TensorFlow, которые рассматриваются в остальной части книги. Мы не будем подробно обсуждать все поддерживаемые версии платформы TensorFlow, версии ОС, версии сред Xcode и Android Studio, которые могут использоваться для разработки, поскольку такую информацию можно легко найти на веб-сайте TensorFlow (<http://www.tensorflow.org>) или в Google. Вместо этого в данной главе мы кратко расскажем о примерах рабочих сред, которые позволят нам быстро углубиться в изучение и разработку всех этих удивительных приложений.

Если у вас уже установлена платформа TensorFlow, а также среды разработки Xcode и Android Studio, и вы можете запускать и тестировать примеры приложений TensorFlow для iOS и Android, и если у вас уже установлен графический процессор NVIDIA GPU, предназначенный для более быстрой тренировки глубоко обучающихся моделей, то вы можете пропустить эту главу либо перейти непосредственно к незнакомому разделу.

В данной главе мы рассмотрим следующие темы (настройка среды разработки в компьютере Raspberry Pi будет обсуждаться в главе 12 «*Разработка приложений TensorFlow на компьютере Raspberry Pi*»):

- настройка платформы TensorFlow;
- настройка среды разработки Xcode;
- настройка среды разработки Android Studio;
- TensorFlow Mobile против TensorFlow Lite;
- запуск примеров приложений TensorFlow для iOS;
- запуск примеров приложений TensorFlow для Android.

НАСТРОЙКА ПЛАТФОРМЫ TENSORFLOW

Платформа TensorFlow является ведущей платформой с открытым исходным кодом для машинного интеллекта. Когда компания Google выпустила

TensorFlow в качестве проекта с открытым исходным кодом в ноябре 2015 года, уже существовало несколько других подобных платформ с открытым исходным кодом для глубокого обучения: Caffe, Torch и Theano. Согласно Google I/O от 10 мая 2018 года, платформа TensorFlow на GitHub достигла 99к звезд, показав рост 14к звезд за 4 месяца, в то время как платформа Caffe увеличила свой показатель только на 2К до 24К звезд. По прошествии двух лет с момента первого релиза это уже самая популярная платформа с открытым исходным кодом для тренировки и развертывания глубоко обучающихся моделей (к тому же она имеет хорошую поддержку традиционного машинного обучения). По состоянию на январь 2018 года платформа TensorFlow имеет порядка 85к звезд (<https://github.com/tensorflow/tensorflow>) на GitHub, в то время как три остальные ведущие платформы с открытым исходным кодом для глубокого обучения, Caffe (<https://github.com/BVLC/caffe>), CNTK (<https://github.com/Microsoft/CNTK>) и Mxnet (<http://github.com/apache/incubator-mxnet>), имеют соответственно более 22к, 13к и 12к звезд.



Если вы немного запутались в технических жаргонизмах «машинное обучение», «глубокое обучение», «машинный интеллект» и «искусственный интеллект» (ИИ), то дадим краткое резюме: термины «машинный интеллект» и «ИИ» на самом деле означают одно и то же; термином «машинное обучение» обозначается область ИИ, к тому же самая популярная; термином «глубокое обучение» обозначается особый тип машинного обучения, а также современный и наиболее эффективный подход к решению сложных задач, таких как компьютерное зрение, распознавание и синтез речи и обработка естественного языка. Поэтому когда в данной книге мы говорим «ИИ», то мы в первую очередь подразумеваем глубокое обучение, настоящего спасителя, который перенес ИИ из долгой зимы в лето. Для получения подробной информации о зиме ИИ и о глубоком обучении обратитесь к веб-страницам https://en.wikipedia.org/wiki/AI_winter и <http://www.deeplearningbook.org>.

Мы исходим из того, что у вас уже имеется базовое понимание принципа работы платформы TensorFlow, но если это не так, то вы должны обратиться к разделу краткого руководства (https://www.tensorflow.org/get_started) и учебных пособий (<https://www.tensorflow.org/tutorials>) веб-сайта TensorFlow или учебным пособиям Awesome TensorFlow (<https://github.com/jtoy/awesome-tensorflow>). На эту тему имеется две хорошие книги: «*Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn*» («Python и машинное обучение: машинное обучение и глубокое обучение с помощью Python, scikit-learn и TensorFlow»), 2-е изд., Себастьян Рашка (Sebastian Raschka) и Вахид Мирджалили (Vahid Mirjalili), и «*Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*» («Практическое машинное обучение с помощью пакета Scikit-Learn и платформы TensorFlow: понятия, инструменты и методы построения интеллектуальных систем»), автор Орелиен Жерон (Aurélien Géron).

Платформа TensorFlow может быть установлена в MacOS, Ubuntu или Windows. Мы рассмотрим этапы установки платформы TensorFlow версии 1.4

из ее исходного кода в MacOS X El Capitan (10.11.6), в macOS Sierra (10.12.6) и в Ubuntu 16.04. Если у вас другая ОС или версия платформы, то для получения дополнительной информации вы можете обратиться к документации по установке платформы TensorFlow (<https://www.tensorflow.org/install>). К тому времени, когда вы будете читать эту книгу, вероятно, выйдет уже новая версия платформы TensorFlow. Хотя вы все равно сможете запускать приводимый в книге программный код с более новой версией, это утверждение не является гарантией того, что это будет именно так, поэтому для настройки платформы TensorFlow в Mac и Ubuntu мы используем исходный код TensorFlow версии 1.4; благодаря этому вы сможете легко проверить ее работоспособность и поиграть с приложениями данной книги.



С того времени (декабрь 2017 года), когда был написан приведенный выше абзац, было выпущено уже четыре новых официальных релиза TensorFlow (1.5, 1.6, 1.7 и 1.8), которые можно загрузить с веб-страницы релизов по адресу <https://github.com/tensorflow/tensorflow/releases> или из репозитория исходного кода TensorFlow (<https://github.com/tensorflow/tensorflow>), а также новая версия среды разработки Xcode (9.3 по состоянию на май 2018 года). Новые версии TensorFlow, такие как 1.8, по умолчанию поддерживают новые версии платформы NVIDIA CUDA и библиотеки cuDNN (см. раздел «*Настройка платформы TensorFlow в Ubuntu с поддержкой GPU*» относительно подробностей), однако мы рекомендуем обязательно просмотреть официальную документацию TensorFlow по установке последней версии TensorFlow с поддержкой GPU. В этой и следующих главах в качестве примера мы можем ссылаться на определенную версию TensorFlow, однако при этом будем поддерживать весь исходный код для iOS и Android и на Python в репозитории исходного кода книги по адресу <https://github.com/jeffxtang/mobiletfbook> в протестированном и, при необходимости, в обновленном виде до самой последней версии платформы TensorFlow, а также сред разработки Xcode и Android Studio.

В целом мы будем использовать платформу TensorFlow в Mac для разработки приложений для iOS и для Android и платформу TensorFlow в Ubuntu для тренировки глубоко обучающихся моделей, которые используются в приложениях.

НАСТРОЙКА TENSORFLOW В MacOS

Как правило, для установки платформы TensorFlow в изолированной среде следует использовать виртуальную среду `virtualenv`, контейнерную среду `Docker` или дистрибутив `Anaconda`. Но поскольку мы будем разрабатывать приложения TensorFlow для iOS и Android, используя исходный код платформы TensorFlow, то мы с тем же успехом можем выполнить сборку непосредственно самой платформы TensorFlow из ее исходного кода, и в этом случае использование нативной установки с помощью менеджера пакетов `pip` может быть проще, чем другие варианты. Если вы хотите поэкспериментировать с различными версиями TensorFlow, то рекомендуем установить другие версии TensorFlow с помощью одного из вариантов: `virtualenv`, `Docker` и `Anaconda`. Здесь мы уста-

новим TensorFlow 1.4 непосредственно в вашу систему MacOS с использованием менеджера пакетов `pip` и среды языка Python 2.7.10, устанавливаемых в ОС по умолчанию.

Выполните следующие ниже действия, чтобы скачать и установить TensorFlow 1.4 в MacOS.

1. Скачать исходный код TensorFlow 1.4.0 (zip или tar .gz) со страницы релизов платформы TensorFlow в GitHub: <https://github.com/tensorflow/tensorflow/releases>.
2. Распаковать скачанный файл и перетащить папку `tensorflow-1.4.0` в свой домашний каталог.
3. Убедиться, что у вас установлена среда разработки Xcode 8.2.1 или выше (если нет, то сначала следует прочитать раздел «*Настройка среды разработки Xcode*»).
4. Открыть новое окно терминала, затем выполнить команду `cd tensorflow-1.4.0`, чтобы перейти в указанный в команде каталог.
5. Выполнить команду `xcode-select --install`, чтобы установить инструменты командной строки.
6. Выполнить следующие ниже команды, чтобы установить другие инструменты и пакеты, необходимые для сборки платформы TensorFlow¹:

```
sudo pip install six numpy wheel
brew install automake
brew install libtool
./configure
brew upgrade bazel
```

7. Выполнить сборку из исходного кода платформы TensorFlow с поддержкой только CPU (мы рассмотрим поддержку GPU в следующем разделе) и сгенерировать пакетный файл менеджера пакетов `pip` с расширением `.whl`:

```
bazel build --config=opt //tensorflow/tools/pip_package:build_pip_package
bazel-bin/tensorflow/tools/pip_package/build_pip_package /tmp/tensorflow_pkg
```

8. Установить пакет TensorFlow 1.4.0 CPU:

```
sudo pip install --upgrade /tmp/tensorflow_pkg/tensorflow-1.4.0-cp27-cp27m-macosx_10_12_intel.whl
```

Если во время данного процесса вы столкнулись с какой-либо ошибкой, то, честно говоря, поиск сообщения об ошибке в Google должен быть самым лучшим способом для ее исправления, поскольку в данной книге мы намерены сосредоточиться на советах и знаниях, труднодоступных в другом месте и полученных после долгих часов сборки и отладки практических мобильных

¹ Bazel – это версия с открытым исходным кодом внутреннего для компании Google инструмента сборки. См. <http://biophysics.med.jhmi.edu/~yliu120/tensorflow.html>. – Прим. перев.

приложений TensorFlow. Одной из конкретных ошибок, которую вы можете увидеть, является `Operation not permitted` (Операция не разрешена) при выполнении команд `sudo pip install`. Для того чтобы ее исправить, вы можете отключить защиту целостности системы (System Integrity Protection, SIP) вашего Mac, перезапустив Mac и нажав клавиши `Cmd+R`, чтобы войти в режим восстановления, а затем перед перезапуском Mac в разделе Утилиты–Терминал, выполнить `csrutil disable`. Если вам не совсем удобно отключать SIP, то вы можете посмотреть документацию платформы TensorFlow, с тем чтобы попробовать один из более сложных методов установки, таких как виртуальная среда `virtualenv`.

Если все идет хорошо, то у вас должно получиться запустить Python или предпочтительно IPython в окне терминала, а затем выполнить инструкции `import tensorflow as tf` и `tf.__version__` и увидеть 1.4.0 на выходе.

НАСТРОЙКА TENSORFLOW В UBUNTU С ПОДДЕРЖКОЙ GPU

Одним из преимуществ использования хорошей платформы глубокого обучения, такой как TensorFlow, является ее полная поддержка использования графического процессора (GPU) во время тренировки модели. Тренировка нетривиальной модели TensorFlow на базе GPU проходит намного быстрее, чем на CPU, и в настоящее время NVIDIA предлагает самые лучшие и самые затратно-эффективные GPU, поддерживаемые платформой TensorFlow. При этом Ubuntu является самой лучшей ОС для совместной работы графических процессоров NVIDIA и платформы TensorFlow. Вы можете легко купить один GPU за несколько сотен долларов и установить его на недорогой настольный компьютер с операционной системой Ubuntu. Вы также можете установить NVIDIA GPU в Windows, но поддержка платформы TensorFlow в Windows не так хороша, как в Ubuntu.

Для тренировки моделей, развернутых в приложениях этой книги, мы используем видеокарту NVIDIA GTX 1070, которую вы можете приобрести в Amazon или eBay примерно за \$400. Есть хороший блог-пост Тима Дэттмерса (Tim Dettmers) относительно того, какой GPU использовать для глубокого обучения (<http://timdettmers.com/2017/04/09/which-gpu-for-deep-learning/>). После того как вы получите такой GPU и установите его в операционную систему Ubuntu, и перед тем, как вы установите платформу TensorFlow с поддержкой GPU, вам нужно установить параллельно-вычислительную платформу NVIDIA CUDA и 8.0 (или 9.0) и библиотеку cuDNN (библиотеку CUDA для глубоких нейронных сетей) 6.0 (или 7.0), обе из которых поддерживаются в TensorFlow 1.4.

! Альтернативой настройке собственной операционной системы Ubuntu с поддержкой GPU для работы с TensorFlow является использование TensorFlow в специальной облачной службе с поддержкой GPU, такой как Cloud ML Engine облачной платформы Google Cloud Platform (<https://cloud.google.com/ml-engine/docs/using-gpus>). У каждо-

го варианта есть свои плюсы и минусы. Облачные службы обычно применяют по-временное выставление счетов. Если ваша цель состоит в тренировке или вторичной тренировке моделей для развертывания на мобильных устройствах, имея в виду, что ваши модели будут не какими-то сверхсложными, и если вы планируете выполнять тренировку машинно-обучающихся моделей в течение длительного времени, то затратно-эффективнее и удовлетворительнее иметь свой собственный графический процессор.

Выполните следующие ниже действия, чтобы установить платформу CUDA 8.0 и библиотеку cuDNN6.0 в Ubuntu 16.04 (у вас должно получиться скачать и установить CUDA 9.0 и cuDNN7.0 аналогичным образом).

1. Найти релиз NVIDIA CUDA 8.0 GA2 на <https://developer.nvidia.com/cuda-80-ga2-download-archive>, и сделать соответствующий выбор, как показано на приведенном ниже скриншоте:

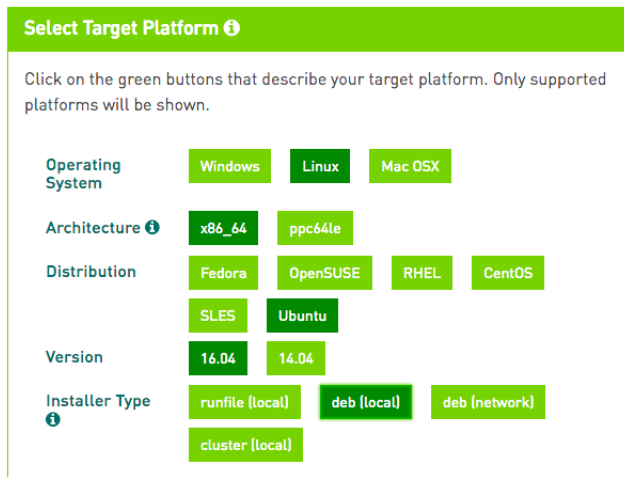


Рисунок 1.1. Подготовка к скачиванию платформы CUDA 8.0 в Ubuntu 16.04

2. Скачать базовый установщик (Base Installer), как показано на следующем ниже скриншоте:



Рисунок 1.2. Выбор установочного файла CUDA 8.0 для Ubuntu 16.04

- Открыть новый терминал и выполнить следующие ниже команды (вам также потребуется добавить две последние команды в файл `.bashrc`, благодаря чему две переменные окружения вступят в силу при следующем запуске нового терминала):

```
sudo dpkg -i /home/jeff/Downloads/cuda-repo-ubuntu1604-8-0-local-ga2_8.0.61-1_amd64.deb
sudo apt-get update
sudo apt-get install cuda-8-0
export CUDA_HOME=/usr/local/cuda
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
```

- Скачать библиотеку NVIDIA cuDNN6.0 для платформы CUDA 8.0 на <https://developer.nvidia.com/rdp/cudnn-download> – прежде чем вы сможете ее скачать, вам будет предложено (бесплатно) зарегистрироваться с использованием учетной записи разработчика NVIDIA, как показано на следующем ниже скриншоте (выберите выделенную библиотеку cuDNN v6.0 для Linux):

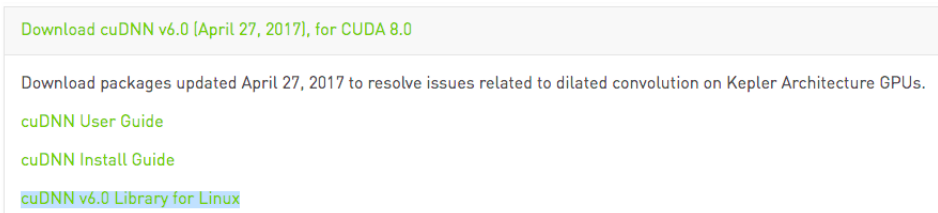


Рисунок 1.3. Выбор библиотеки cuDNN6.0 для платформы CUDA 8.0 в Linux

5. Распаковать скачанный файл, при этом он должен находиться в установленном по умолчанию каталоге `-/Downloads`, и в нем вы увидите папку с именем `cuda` с двумя подпапками `include` и `lib64`.
6. Скопировать файлы `cuDNN` в папках `include` и `lib64` в подпапки `include` и `lib64` папки `CUDA_HOME`:

```
sudo cp ~/Downloads/cuda/lib64/* /usr/local/cuda/lib64
sudo cp ~/Downloads/cuda/include/cudnn.h /usr/local/cuda/include
```

Теперь мы готовы установить TensorFlow 1.4 с поддержкой GPU в Ubuntu (первые приведенные здесь два шага являются такими же, как и те, которые описаны в разделе «*Настройка платформы TensorFlow в MacOS*»):

1. Скачать исходный код TensorFlow 1.4.0 (zip или tar.gz) со страницы релизов платформы TensorFlow на GitHub: <https://github.com/tensorflow/tensorflow/releases>.
2. Распаковать скачанный файл и перетащить папку в свой домашний каталог.
3. Скачать установщик `bazel`, `bazel-0.5.4-installer-linux-x86_64.sh` с <https://github.com/bazelbuild/bazel/releases>.
4. Открыть новое окно терминала, затем выполнить следующие ниже команды для установки инструментов и пакетов, необходимых для сборки платформы TensorFlow:

```
sudo pip install six numpy wheel
cd ~/Downloads
chmod +x bazel-0.5.4-installer-linux-x86_64.sh
./bazel-0.5.4-installer-linux-x86_64.sh --user
```

5. Выполнить сборку из исходного кода платформы TensorFlow с поддержкой GPU и сгенерировать пакетный файл менеджера пакетов `pip` с расширением `.whl`:

```
cd ~/tensorflow-1.4.0
./configure
bazel build --config=opt --config=cuda //tensorflow/tools/pip_package:build_pip_package
bazel-bin/tensorflow/tools/pip_package/build_pip_package /tmp/tensorflow_pkg
```

6. Установить пакет TensorFlow 1.4.0 GPU:

```
sudo pip install --upgrade /tmp/tensorflow_pkg/tensorflow-1.4.0-cp27-cp27mu-linux_x86_64.whl
```

Теперь, если все идет хорошо, вы можете запустить IPython и ввести следующие ниже инструкции, чтобы увидеть информацию о GPU, который используется платформой TensorFlow:

```
In [1]: import tensorflow as tf
In [2]: tf.__version__
Out[2]: '1.4.0'
```

```
In [3]: sess=tf.Session()
```

```
2017-12-28 23:45:37.599904: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:892]
successful NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero
2017-12-28 23:45:37.600173: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Found
device 0 with properties:
name: GeForce GTX 1070 major: 6 minor: 1 memoryClockRate(GHz): 1.7845
pciBusID: 0000:01:00.0
totalMemory: 7.92GiB freeMemory: 7.60GiB
2017-12-28 23:45:37.600186: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1120]
Creating TensorFlow device (/device: GPU:0) -> (device: 0, name: GeForce GTX 1070, pci bus
id: 0000:01:00.0, compute capability: 6.1)
```

Поздравляю! Теперь вы готовы к тренировке глубоко обучающихся моделей, используемых в приложениях этой книги. Прежде чем мы начнем играть с нашей новой игрушкой и использовать ее для тренировки наших моделей, а затем развертывать и запускать их на мобильных устройствах, давайте сначала посмотрим, что нужно для того, чтобы быть готовым к разработке мобильных приложений.

НАСТРОЙКА СРЕДЫ РАЗРАБОТКИ XCODE

Среда разработки Xcode используется для разработки приложений iOS. Вы можете ее скачать и установить, имея в распоряжении компьютер Mac и бесплатный идентификатор Apple ID. Если ваш Mac является относительно старым и имеет операционную систему OS X El-Capitan версии 10.11.6, то вы можете скачать Xcode 8.2.1 с <https://developer.apple.com/download/more>. Если же у вас операционная система macOS Sierra версии 10.12.6 или более поздней версии, то вы можете скачать Xcode 9.2 или 9.3, то есть последнюю версию по состоянию на май 2018 года, по ранее указанной ссылке. Все приложения для iOS в данной книге были протестированы в среде разработки Xcode версий 8.2.1, 9.2 и 9.3.

Для того чтобы установить среду Xcode, просто дважды щелкните на скачанном файле и следуйте инструкциям на экране. Все довольно просто. Теперь вы можете запускать приложения в симуляторе iOS, который поставляется вместе со средой Xcode, или на своем собственном устройстве iOS. Начиная с Xcode 7 вы можете запускать и отлаживать свои приложения iOS на устройстве iOS бесплатно, но если же вы желаете свои приложения распространять или публиковать, то вам придется зарегистрироваться в качестве физического лица в программе Apple Developer Program за 99 долларов США в год: <https://developer.apple.com/programs/enroll>.

Хотя вы можете выполнять тестовый прогон многих приложений книги с помощью симулятора Xcode, некоторые приложения данной книги задействуют фотокамеру вашего устройства iOS, делая снимок перед его обработкой глубоко обучающейся моделью, натренированной с помощью платформы TensorFlow. Кроме того, обычно предпочтительнее тестировать модель на реальном устройстве для определения точной производительности и потребле-

ния оперативной памяти: модель, которая отлично работает в симуляторе, может завершить работу аварийно или же работать слишком медленно на реальном устройстве. Поэтому настоятельно рекомендуется или требуется протестировать и запускать приводимые в книге приложения для iOS на своем фактическом устройстве iOS, хотя бы раз, если не всегда.

В этой книге мы исходим из того, что вы знакомы с программированием для iOS, но если вы в разработке приложений для iOS новичок, то можете познакомиться с ней из многочисленных и отличных онлайн-руководств, таких как руководства по iOS Рэя Вендерлика (Ray Wenderlich) (<https://www.raywenderlich.com>). Мы не будем касаться сложных вопросов программирования для iOS; мы в основном покажем вам, как использовать API C++ платформы TensorFlow в наших приложениях для iOS с целью запуска натренированных моделей TensorFlow и выполнения всевозможных интеллектуальных задач. При этом для взаимодействия с программным кодом C++ в наших мобильных приложениях искусственного интеллекта будет использоваться программный код на двух официальных языках программирования iOS, предлагаемых компанией Apple: Objective-C и Swift.

НАСТРОЙКА СРЕДЫ РАЗРАБОТКИ ANDROID STUDIO

Среда разработки Android Studio является самым лучшим инструментом разработки приложений для Android, и платформа TensorFlow имеет весомую поддержку ее использования. В отличие от среды разработки Xcode, вы можете установить и запустить Android Studio в Mac, Windows или Linux. Подробные требования к операционной системе вы можете найти на веб-сайте Android Studio (<https://developer.android.com/studio/index.html>). Здесь мы расскажем, как настроить среду разработки Android Studio 3.0 или 3.0.1 в Mac – все приложения в книге были протестированы в обеих этих версиях.

Сначала следует скачать среду разработки Android Studio 3.0.1 или ее последнюю версию, если она новее, чем 3.0.1, и, если вы не возражаете, исправить возможные незначительные проблемы, упомянутые в предыдущей ссылке. Вы также можете скачать версии 3.0.1 или 3.0 из ее архивов на веб-странице по адресу <https://developer.android.com/studio/archive.html>.

Затем нужно дважды щелкнуть на скачанном файле и перетащить значок Android studio.app в Applications (Приложения). Если у вас среда разработки Android Studio уже установлена, то вам будет предложено заменить ее на более новую версию. Вы можете просто выбрать **Replace** (Заменить).

Далее нужно открыть среду разработки Android Studio и указать путь к комплекту разработчика Android SDK, который по умолчанию расположен в папке `~/Library/Android/sdk`, в случае если у вас установлена предыдущая версия Android Studio, либо можно выбрать в меню **Open an existing Android Studio project** (Открыть существующий проект Android Studio), а затем перейти в каталог с исходным кодом платформы TensorFlow 1.4, созданный в разделе «Настройка плат-

формы TensorFlow в macOS», и открыть папку `tensorflow/examples/android`. После этого вы можете скачать Android SDK, щелкнув ссылку на сообщении **Install Build Tools** (Установить инструменты сборки) либо перейдя в **Android Studio Tools** (Инструменты Android Studio) | **Android** | **SDK Manager** (Менеджер SDK), как показано на следующем ниже скриншоте. На вкладке **SDK Tools** (Инструменты SDK) можно установить флажок рядом с определенной версией инструментов Android SDK Tools и нажать кнопку **OK**, чтобы установить эту версию:

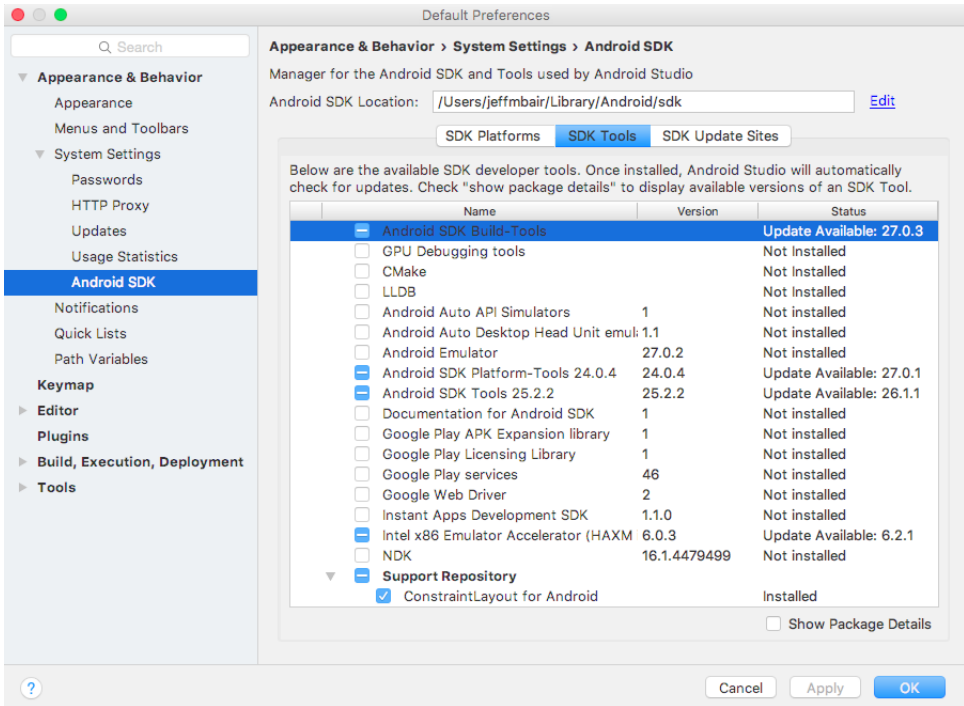


Рисунок 1.4. Менеджер Android SDK для установки инструментов SDK и NDK

Наконец, поскольку для загрузки и выполнения моделей TensorFlow в приложениях TensorFlow для Android используется нативная библиотека TensorFlow C++, вам нужно установить нативный комплект разработчика Android **Native Development Kit** (NDK). Это можно сделать из менеджера Android SDK, показанного на предыдущем скриншоте, либо скачать NDK непосредственно с <https://developer.android.com/ndk/downloads/index.html>. Обе версии NDK, r16b и r15c, были протестированы на предмет запуска приводимых в данной книге приложений для Android. Если вы скачаете NDK непосредственно, то вам также может потребоваться задать месторасположение Android NDK после открытия проекта Android Studio и выбора **File** (Файл) | **Project Structure** (Структура проекта), как показано на следующем ниже скриншоте:

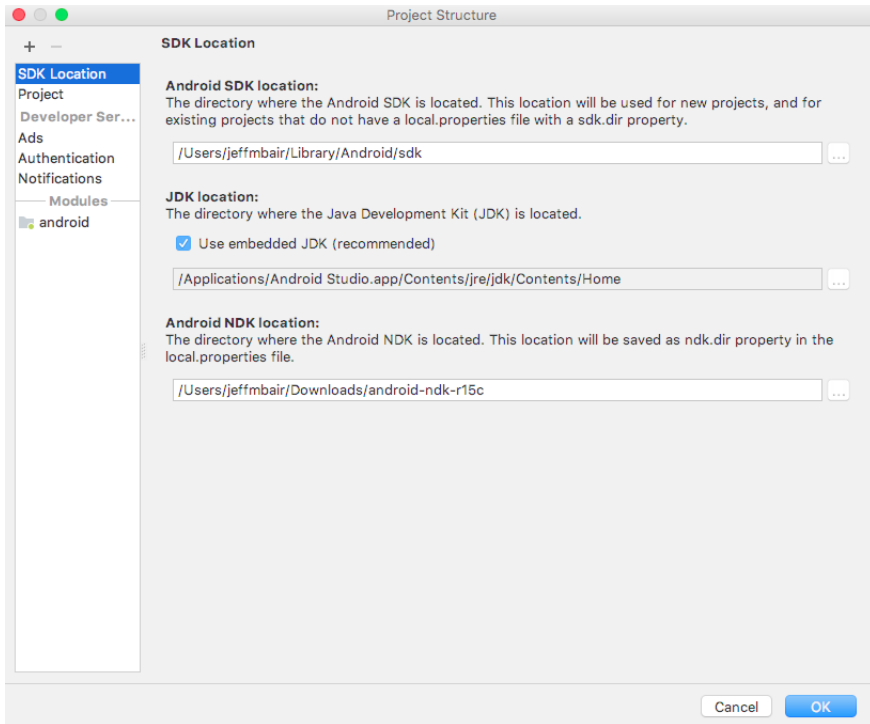


Рисунок 1.5. Установка расположения Android NDK уровня проекта

После установки и настройки Android SDK и NDK вы готовы к тестовому прогону примеров приложений TensorFlow для Android.

TENSORFLOW MOBILE ПРОТИВ TENSORFLOW LITE

Прежде чем начать работать с примерами приложений TensorFlow для iOS и Android, давайте проясним одну общую картину. Платформа TensorFlow в настоящее время имеет два подхода к разработке и развертыванию приложений с применением глубокого обучения на мобильных устройствах: TensorFlow Mobile и TensorFlow Lite. Решение TensorFlow Mobile было частью платформы TensorFlow с самого начала, тогда как TensorFlow Lite представляет собой новый способ разработки и развертывания приложений TensorFlow, поскольку эта платформа обеспечивает лучшую производительность и меньший размер приложения. Однако существует один ключевой фактор, который позволит нам в этой книге сосредоточиться исключительно на платформе TensorFlow Mobile и посвятить рассмотрению платформы TensorFlow Lite всего одну главу: на момент выхода TensorFlow 1.8 и по данным конференции разработчиков Google I/O, проходившей в мае 2018 года, платформа TensorFlow Lite все

еще находится в предварительной версии. Поэтому, как рекомендует сама компания Google, в настоящее время для разработки готовых к производству мобильных приложений с поддержкой платформы TensorFlow вам нужно использовать платформу TensorFlow Mobile.

Еще одна причина, по которой мы сейчас решили сосредоточиться на платформе TensorFlow Mobile, состоит в том, что, в отличие от платформы TensorFlow Lite, которая предлагает лишь ограниченную поддержку модельных операторов, платформа TensorFlow Mobile поддерживает кастомизацию, обеспечивающую добавление новых операторов, не поддерживаемых платформой TensorFlow Mobile по умолчанию, что, как вы увидите, происходит довольно часто в наших самых разных моделях из разных приложений ИИ.

Однако в будущем, когда платформа TensorFlow Lite выйдет из предварительной версии, она, вероятно, заменит собой платформу TensorFlow Mobile или, по крайней мере, преодолеет свои текущие ограничения. Для того чтобы подготовиться к этому моменту, мы подробно рассмотрим платформу TensorFlow Lite в одной из последних глав.

Выполнение примеров приложений TensorFlow для iOS

В последних двух разделах этой главы мы протестируем три примера приложений для iOS и четыре примера приложений для Android, которые поставляются вместе с TensorFlow 1.4. Это позволит вам убедиться, что ваши мобильные среды разработки TensorFlow настроены правильно, а также получить предварительное впечатление о том, что способны выполнять мобильные приложения с поддержкой TensorFlow.

Исходный код трех примеров приложений TensorFlow для iOS находится в папке `tensorflow/examples/ios`. Эти приложения называются `simple` (простое), `camera` (фотокамера) и `benchmark` (оценка производительности). Для того чтобы успешно выполнить эти примеры, сначала необходимо скачать одну глубоко обучающуюся модель, предварительно натренированную компанией Google, под названием Inception (Исходная) (<https://github.com/tensorflow/models/tree/master/research/inception>), предназначенную для распознавания изображений. Существует несколько версий модели Inception: от v1 до v4, с более высокой точностью в каждой новой версии. Здесь мы будем использовать Inception v1, так как примеры приложений были разработаны для нее. Скачав файл модели, скопируйте связанные с моделью файлы в папку `data` каждого примера:

```
curl -o ~/graphs/inception5h.zip https://storage.googleapis.com/download.tensorflow.org/models/inception5h.zip
unzip ~/graphs/inception5h.zip -d ~/graphs/inception5h
cd tensorflow/examples/ios
cp ~/graphs/inception5h/* simple/data/
cp ~/graphs/inception5h/* camera/data/
cp ~/graphs/inception5h/* benchmark/data/
```

Теперь перейдите в каждую папку app и выполните следующие ниже команды, чтобы перед открытием и запуском приложений скачать необходимый для каждого приложения модуль:

```
cd simple
pod install
open tf_simple_example.xcworkspace
cd ../camera
pod install
open tf_camera_example.xcworkspace
cd ../benchmark
pod install
open tf_benchmark_example.xcworkspace
```

Затем вы можете запустить все три приложения на устройстве iOS или же приложения simple и benchmark в симуляторе iOS. Если после запуска приложения simple коснуться кнопки **Run Model** (Запустить модель), то вы увидите текстовое сообщение о том, что модель Inception платформы TensorFlow загружена, а затем будет выведено несколько самых лучших результатов распознавания вместе с показателями достоверности.

Если после запуска приложения benchmark коснуться кнопки **Benchmark Model** (Оценка производительности), то можно увидеть среднее время, необходимое для выполнения модели более 20 раз. Например, на моем iPhone 6 это занимает в среднем около 0,2089 секунды и 0,0359 секунды в симуляторе iPhone 6.

Наконец, после запуска приложения camera на устройстве iOS и наведения фотокамеры устройства на окружающие предметы будут показаны предметы, которые приложение видит и распознает в режиме реального времени.

ВЫПОЛНЕНИЕ ПРИМЕРОВ ПРИЛОЖЕНИЙ TENSORFLOW ДЛЯ ANDROID

В случае Android имеется четыре примера приложений TensorFlow. Это **TF Classify** (Классификация), **TF Detect** (Обнаружение), **TF Speech** (Речь) и **TF Stylize** (Стилизация), расположенные в папке tensorflow/examples/android. Самый простой способ запустить эти примеры – просто открыть проект в указанной выше папке с помощью Android Studio, как показано в разделе «*Настройка среды разработки Android Studio*», а затем внести одно изменение, отредактировав файл build.gradle проекта и заменив инструкцию `def nativeBuildSystem = 'bazel'` на `def nativeBuildSystem = 'none'`.

Теперь подключите устройство Android к компьютеру и выполните сборку, установите и запустите приложение, выбрав **Run** (Выполнить) | **Run android** (Запустить android) в среде Android Studio, в результате чего на вашем устройстве будут установлены четыре приложения для Android с именами TF Classify, TF Detect, TF Speech и TF Stylize. Приложение TF Classify так же, как и прило-

жение camera для iOS, использует модель TensorFlow Inception v1 для классифицирования объектов в режиме реального времени с помощью фотокамеры устройства. Приложение TF Detect использует другую модель под названием однократного мультирамочного детектора (Single Shot Multibox Detector, SSD)¹ вместе с MobileNet, новым набором выпущенных компанией Google глубоко обучающихся моделей, которые ориентированы, в частности, на мобильные и встроенные устройства, для обнаружения объектов и отрисовки рамок на обнаруженных объектах. Приложение TF Speech использует еще одну глубоко обучающуюся модель (распознавания речи) для прослушивания и распознавания небольшого набора слов, таких как yes, no, left, right, stop и go. Приложение TF Stylize использует модель изменения стиля изображений, которые видит фотокамера. Для получения более подробной информации об этих приложениях вы можете обратиться к документации по указанным примерам приложений платформы TensorFlow для Android на веб-странице <https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/android>.

РЕЗЮМЕ

В этой главе мы рассмотрели способы установки платформы TensorFlow 1.4 в Mac и Ubuntu, настройки затратно-эффективного графического процессора NVIDIA GPU в Ubuntu для ускоренной тренировки моделей и настройки сред разработки Xcode и Android Studio с целью разработки мобильных приложений ИИ. Мы также показали, как запускать несколько интересных примеров приложений с поддержкой TensorFlow для iOS и Android. В оставшейся части книги мы более подробно займемся вопросами выполнения сборки и тренировки или вторичной тренировки каждой из этих моделей, используемых в приложениях, и многих других, в операционной системе Ubuntu с поддержкой GPU, и покажем, как развертывать модели в приложениях для iOS и Android и писать программный код для использования моделей в своих мобильных приложениях ИИ. Теперь, когда все готово, нам очень не терпится отправиться в путь. Это будет захватывающее путешествие, путешествие, которым мы, безусловно, будем рады поделиться с нашими друзьями. Так почему бы не начать с наших *самых* лучших друзей? Давайте посмотрим, что требуется для того, чтобы создать приложение для распознавания пород собак.

¹ Термин SSD (однократный мультирамочный детектор объектов) используется для описания архитектур, в которых применяется одна сверточная нейронная сеть (feedforward convolutional network) для непосредственного предсказания расположения областей и их классов, без применения второго этапа классификации. – Прим. перев.