МАШИННОЕ ОБУЧЕНИЕ & TensorFlow

Нишант Шакла

MANNING

ББК 21.818
УДК 004.85
Ш17

Шакла Нишант

Ш17 Машинное обучение и TensorFlow. — СПб.: Питер, 2019. — 336 с.: ил. — (Серия «Библиотека программиста»).

ISBN 978-5-4461-0826-8

Знакомство с машинным обучением и библиотекой TensorFlow похоже на первые уроки в автошколе, когда вы мучаетесь с параллельной парковкой, пытаетесь переключить передачу в нужный момент и не перепутать зеркала, лихорадочно вспоминая последовательность действий, в то время как ваша нога нервно подрагивает на педали газа. Это сложное, но необходимое упражнение. Так и в машинном обучении: прежде чем использовать современные системы распознавания лиц или алгоритмы прогнозирования на фондовом рынке, вам придется разобраться с соответствующим инструментарием и набором инструкций, чтобы затем без проблем создавать собственные системы.

Новички в машинном обучении оценят прикладную направленность этой книги, ведь ее цель — познакомить с основами, чтобы затем быстро приступить к решению реальных задач. От обзора концепций машинного обучения и принципов работы с TensorFlow вы перейдете к базовым алгоритмам, изучите нейронные сети и сможете самостоятельно решать задачи классификации, кластеризации, регрессии и прогнозирования.

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 21.818
УДК 004.85

ISBN 978-1617293870 англ.
ISBN 978-5-4461-0826-8

Оглавление

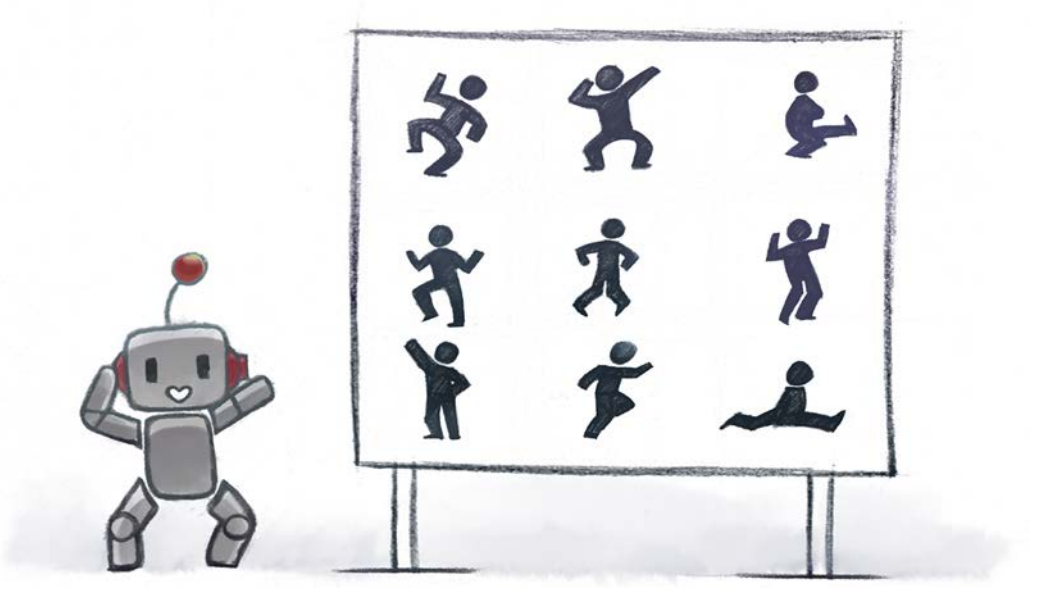
Часть I

Ваше снаряжение для машинного обучения

Обучение параллельной парковке поначалу представляется жутким испытанием. Первые несколько дней уходят на ознакомление с кнопками, вспомогательными камерами и чувствительностью двигателя. Знакомство с машинным обучением и библиотекой TensorFlow происходит аналогичным образом. Прежде чем использовать современные системы распознавания лиц или прогнозов фондовой биржи, необходимо в первую очередь освоиться с соответствующим инструментарием.

Чтобы подготовить надежную основу для машинного обучения, требуется учесть два аспекта. Первый аспект раскрывается в главе 1 и состоит в том, что необходимо освоить язык и теорию машинного обучения. Чтобы говорить на эту тему на одном языке, исследователи в своих публикациях дали точную терминологию и формулировки. Поэтому нам тоже лучше придерживаться этих правил, чтобы избежать путаницы. Второй аспект раскрывается в главе 2 и касается всего, что необходимо знать, чтобы начать использовать библиотеку TensorFlow. У самураев есть самурайский меч, у музыкантов — музыкальные инструменты, а у специалистов-практиков в машинном обучении есть TensorFlow.

1 Одиссея машинного обучения



Эта глава охватывает следующие темы:

- ✓ Основы машинного обучения
- ✓ Представление данных, признаки и нормы векторов
- ✓ Причины выбора TensorFlow

Вы когда-нибудь задумывались, есть ли предел того, что можно вычислить при помощи компьютерной программы? В наши дни компьютеры способны делать гораздо больше, чем просто решать математические уравнения. Во второй половине столетия программирование стало основным инструментом для автоматизации операций и средством экономии времени, но каков объем того, что мы можем автоматизировать и как вообще можно себе такое представить?

Может ли компьютер изучить фотографию и сказать: «Ага, вижу влюбленную парочку, прогуливающуюся по мосту под зонтом во время дождя»? Может ли программное обеспечение ставить настолько же точные медицинские диагнозы, как и опытный специалист? Могут ли прогнозы программного обеспечения относительно ситуации на фондовых рынках быть лучше, чем умозаключения человека? Достижения последнего десятилетия позволяют предположить, что ответом на все эти вопросы является многократное «да», а в основе реализации этих задач лежат общие методы.

Недавние достижения в теоретических исследованиях вкупе с новейшими технологиями дают возможность каждому при наличии компьютера попытаться найти свой подход к решению этих чрезвычайно сложных задач. Ну хорошо, не совсем каждому, но ведь поэтому-то вы и читаете эту книгу, верно?

Программисту больше не требуется знать запутанные подробности задачи, чтобы приступить к ее решению. Возьмем преобразование речи в текст: при традиционном подходе вам, вероятно, понадобилось бы разобраться с биологической структурой голосовых связок человека, чтобы иметь возможность декодировать выражения, используя для этого многочисленные, спроектированные вручную,

зависящие от предметной области и необобщаемые фрагменты программного кода. В наши дни можно написать программу, которая просмотрит множество примеров и выявит пути решения этой задачи при условии наличия достаточного количества времени и базы таких примеров.

Алгоритмы обучаются по данным аналогично тому, как люди учатся на собственном опыте. Люди учатся, читая книги, анализируя ситуации, обучаясь в школе, обмениваясь информацией во время разговоров и просматривая веб-сайты, и это помимо множества других методов. Как может машина развить способность обучаться? Окончательного ответа на этот вопрос нет, но исследователи мирового уровня разработали разумные программы для разных областей. В различных реализациях этих программ ученые заметили повторяющиеся образы в решении задач этого типа, что привело к возникновению специализированной области, которую сегодня называют *машинным обучением* (МО, machine learning).

По мере развития машинного обучения используемые инструменты становились все более стандартизованными, надежными, высокопроизводительными и масштабируемыми. Именно на этом этапе появилась TensorFlow. Эта библиотека программного обеспечения имеет интуитивный интерфейс, который позволяет программистам погрузиться в сложные идеи машинного обучения и сразу же применять их на практике. В следующей главе приводятся азы этой библиотеки, а все последующие главы содержат описание того, как использовать TensorFlow для каждой из различных областей применения машинного обучения.

НАДЕЖНЫЙ РЕЗУЛЬТАТ МАШИННОГО ОБУЧЕНИЯ

Распознавание образов теперь не является чертой, присущей исключительно человеческим существам. Взрывной рост тактовой частоты компьютера и объема используемой памяти привел нас к необычной ситуации: компьютеры теперь можно использовать для составления прогнозов, выявления аномалий, упорядочивания элементов и автоматической классификации изображений. Этот новый набор инструментов дает разумные ответы к задачам, которые не имеют четкого решения, но тут перед нами встает вопрос о доверии. Доверите ли вы компьютерному алгоритму выдачу жизненно важных рекомендаций относительно лечения: например, выполнять операцию на сердце или нет?

В таких вопросах нет места недостоверным методам машинного обучения. Доверие человека — слишком хрупкая субстанция, и наши алгоритмы, несомненно, должны быть надежными. Помните об этом и внимательно и с осторожностью изучайте информацию в этой главе.

1.1. Основы машинного обучения

Пытались ли вы когда-нибудь объяснить кому-либо, как правильно плавать? Объяснение ритма движений и форм обтекаемости ошеломляет своей сложностью. Аналогичным образом некоторые задачи программного обеспечения слишком сложны для нас, чтобы их можно было без труда охватить нашим сознанием. Именно для таких задач машинное обучение может оказаться самым подходящим инструментом.

Когда-то алгоритмы для выполнения этой работы собирались вручную и тщательно отлаживались, и это был единственный способ создания программного обеспечения. Проще говоря, традиционное программирование предполагает детерминированный выход для каждого набора входных данных. Машинное обучение, напротив, может решать класс задач, для которых соответствие входа и выхода недостаточно хорошо определено.

ПОЛНЫЙ ВПЕРЕД!

Машинное обучение является относительно молодым методом, а поэтому представьте, что вы — геометр в эпоху Евклида, прокладывающий путь в новой, только что открытой области. Или физик во времена Ньютона, обдумывающий нечто, подобное общей теории относительности, но для машинного обучения.

В машинном обучении используется программное обеспечение, которое обучается на основе ранее полученного опыта. Такая компьютерная программа улучшает свои результаты по мере того, как получает все новые и новые примеры. Тогда можно надеяться, что, если вы закинете в этот «механизм» достаточное количество данных, он научится распознавать образы и выдавать разумные результаты уже для новых входных данных.

Машинное обучение называют также *индуктивным обучением* (inductive learning), потому что код старается выявить структуру только лишь на основе данных. Это все равно что отправиться на каникулы за границу и читать местный журнал мод, пытаясь понять, как одеться, чтобы сойти «за своего». Изучая изображения людей в местной одежде, вы сможете сформировать в своем представлении некий образ локальной культуры. Такой способ обучения называют *индуктивным*.

Возможно, ранее вам никогда не доводилось применять такой подход к программированию, так как необходимость в индуктивном обучении есть не всегда. Предположим, вам нужно определить, четным или нечетным числом является сумма двух произвольных чисел . Уверен, вы можете представить себе тренировку алгоритма машинного обучения на миллионе обучающих примеров (рис. 1.1), но вы, безусловно, понимаете, что это крайность. Более прямой подход может без труда решить эту задачу.

Вход		Выход
$x_1 = (2, 2)$	→	y_1 = Четное
$x_2 = (3, 2)$	→	y_2 = Нечетное
$x_3 = (2, 3)$	→	y_3 = Нечетное
$x_4 = (3, 3)$	→	y_4 = Четное
...		...

Рис. 1.1. Каждая пара целых чисел при их суммировании дает четное или нечетное число. Перечисленные соответствия входа и выхода носят название контрольного набора данных (ground-truth dataset)

Например, сумма двух нечетных чисел всегда является четным числом. Убедитесь сами: возьмите два любых нечетных числа, сложите их между собой и проверьте, является ли их сумма четным числом. Вот как можно доказать этот факт:

- Для любого целого числа n выражение $2n + 1$ дает нечетное число. Более того, любое нечетное число можно записать как $2n + 1$ для некоторого целого числа n. Число 3 можно записать как $2(1) + 1$. А число 5 можно записать как $2(2) + 1$.

- Пусть у нас два нечетных числа, $2n + 1$ и $2m + 1$, где n и m — целые числа. Сложение двух нечетных чисел дает $(2n + 1) + (2m + 1) = 2n + 2m + 2 = 2(n + m + 1)$. Это — четное число, потому что умножение любого целого числа на 2 дает четное число.

Аналогичным образом мы видим, что сумма двух четных чисел тоже является четным числом: $2m + 2n = 2(m + n)$. И наконец, мы также приходим к выводу, что сумма четного и нечетного чисел является нечетным числом: $2m + (2n + 1) = 2(m + n) + 1$. На рис. 1.2 эта логика представлена более понятно.

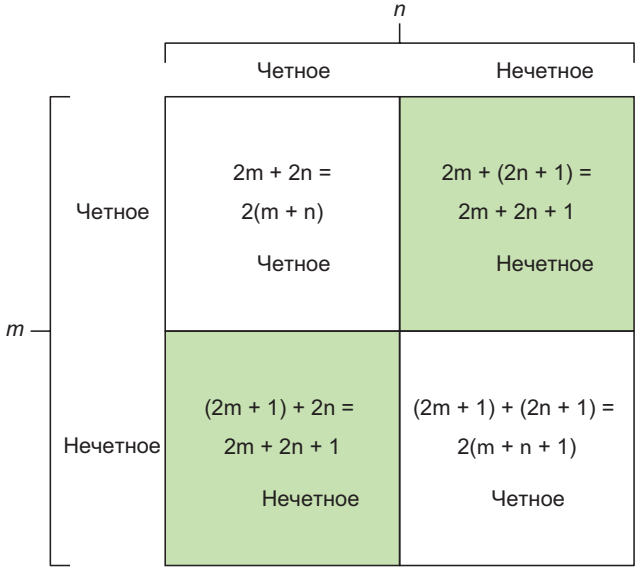


Рис. 1.2. Таблица раскрывает внутреннюю логику соответствия выходных данных входным парам целых чисел

Вот и все! Безо всякого машинного обучения вы можете решить эту задачу для любой пары целых чисел, которую вам кто-нибудь подкинет. Эту задачу можно решить прямым применением математических правил. Однако в алгоритмах машинного обучения внутреннюю логику принято рассматривать как *черный ящик*; это означает, что логика происходящего внутри может быть не очевидна для интерпретации, как это показано на рис. 1.3.

Рис. 1.3. Подход к решению задач в машинном обучении можно представить как настройку параметров черного ящика до тех пор, пока он не начнет выдавать удовлетворительные результаты

1.1.1. Параметры

Иногда тот способ, который позволяет наилучшим образом реализовать алгоритм, преобразующий входной сигнал в соответствующий выходной, является слишком сложным. Например, если на вход подать серию чисел, кодирующих изображение в градациях серого, можно представить, насколько сложно написать алгоритм для маркировки каждого элемента этого изображения. Машинное обучение оказывается полезным, когда не слишком понятно, какая именно работа происходит внутри объекта. Оно предоставляет нам набор инструментов для написания программы без необходимости вдаваться в каждую деталь алгоритма. Программист может оставить некоторые значения неопределенными, тем самым давая возможность системе машинного обучения самой определить их наилучшие значения.

МАШИННОЕ ОБУЧЕНИЕ МОЖЕТ РЕШАТЬ ЗАДАЧИ, НЕ ПРОНИКАЯ В СУТЬ ПРЕДМЕТА

Искусство индуктивного решения задач — это палка о двух концах. Алгоритмы машинного обучения дают неплохой результат при решении определенных задач, однако, несмотря на это, попытки пошагово, с применением дедуктивных методов, проследить, как получился такой результат, могут не сразу увенчаться успехом. Тщательно продуманная система машинного обучения изучает тысячи параметров, но выяснение значения каждого параметра не всегда является ее основной задачей. Уверен, что, располагая этой информацией, вы откроете перед собой поистине волшебный мир.

Неопределенные значения называют *параметрами*, а их описание называют *моделью*. Ваша работа состоит в том, чтобы написать алгоритм, который проанализирует имеющиеся примеры и выявит, как наилучшим образом настроить параметры для получения оптимальной модели. Это невероятно важная мысль! Не волнуйтесь, эта концепция станет лейтмотивом книги, и вы столкнетесь с ней еще много раз.

УПРАЖНЕНИЕ 1.1

Предположим, вы три месяца собирали данные о ситуации на фондовом рынке. Вам хотелось бы прогнозировать дальнейшие тенденции этого рынка в целях переиграть систему и получать денежную выгоду. Как бы вы решили эту задачу, не используя методы машинного обучения? (Как вы узнаете в главе 8, эту задачу можно решить именно методами машинного обучения.)

ОТВЕТ

Верите вы или нет, но твердо установленные правила являются стандартным способом определить торговые стратегии фондового рынка. Например, часто используют простой алгоритм: «если цена падает на 5 %, акции можно покупать». Обратите внимание на то, что в этом случае машинное обучение не используется, а применяется только традиционная логика.

1.1.2. Обучение и вывод

Представьте, что вы пытаетесь приготовить десерт в духовке. Если вы новичок в готовке, то, чтобы получить что-то по-настоящему вкусное, вам потребуется несколько дней для выяснения правильного состава и точного соотношения ингредиентов. Если вы запишете этот рецепт, то впоследствии сможете быстро воспроизвести его и получить в точности то же самое вкусное блюдо.

Машинное обучение аналогичным образом использует идею с рецептами. Обычно мы проверяем алгоритм на двух стадиях: *стадии обучения* и *стадии логического вывода*. Цель этапа обучения — описание данных, которые называются *вектором признаков*, и сведение их в модели. Модель как раз и является

нашим рецептом. В сущности, модель — это программа с парой открытых интерпретаций, а данные позволяют устранять присущую ей двусмысленность.

> **ПРИМЕЧАНИЕ** *Вектор признаков объекта, признаковое описание* (feature vector), представляет собой упрощенное представление исходных данных. Вектор признаков можно рассматривать как сводку характеристик реальных объектов. Стадии обучения и вывода используют вектор признаков, а не сами исходные данные.

Аналогично тому как рецепты могут использоваться другими людьми, модель, полученная в результате обучения, может использоваться повторно в других программах. Больше всего времени занимает стадия обучения. Для получения полезной модели может потребоваться ждать выполнения алгоритма в течение нескольких часов, если не дней и недель. На рис. 1.4 показан процесс обучения.

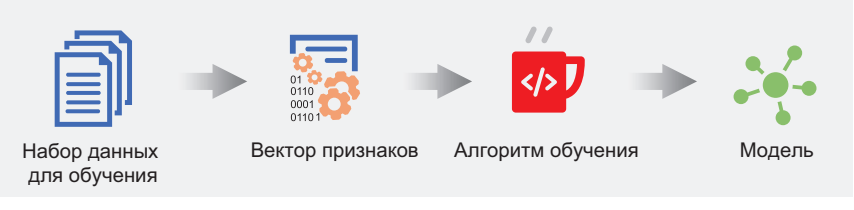


Рис. 1.4. Метод обучения обычно следует структурированному рецепту. Сначала набор данных необходимо преобразовать в представление, чаще всего — в список признаков, который затем можно использовать в обучающем алгоритме. Обучающий алгоритм выбирает модель и подбирает ее параметры наилучшим образом

Стадия вывода использует полученную модель, чтобы сделать умозаключения в отношении данных, прежде ему неизвестных. Это похоже на использование рецепта, найденного в интернете. Процесс логического вывода обычно занимает на порядок меньше времени, чем обучение; вывод можно сделать достаточно быстро в режиме реального времени. К этапу вывода относят все, что касается тестирования модели на новых данных и анализа результатов в процессе испытания, как показано на рис. 1.5.

Рис. 1.5. Для логического вывода обычно используется модель, которая уже прошла стадию обучения или проходит ее. После преобразования данных в удобное представление, такое как вектор признаков, они используются моделью для получения выходных данных

1.2. Представление данных и признаки

Данные имеют первостепенное значение для машинного обучения. Компьютеры — это не более чем сложные калькуляторы, и поэтому данные, которые мы подаем в систему машинного обучения, должны быть вполне определенными математическими объектами, такими как векторы, матрицы или графы.

Главными элементами всех форм представления данных выступают *признаки*, которые являются наблюдаемыми свойствами объекта:

- *Векторы* имеют плоскую и простую структуру и обычно, в большинстве приложений для машинного обучения, представляют собой объединение данных. У них есть два атрибута: *размерность* вектора (это натуральное число) и *тип* его элементов (например, действительные числа, целые числа и т. п.). В качестве напоминания вот некоторые примеры двумерных целочисленных векторов: (1, 2) и (–6, 0) — некоторые примеры трехмерных векторов действительных чисел, такие как (1,1; 2,0; 3,9) и (π, $\pi/2$, $\pi/3$). Вы, вероятно, уже догадались: все это собрание чисел одного типа. В программе, использующей машинное обучение, вектор применяют для измерения свойств данных, таких как цвет, плотность, сила звука, близость к чему-либо, то есть всего того, что можно описать с помощью последовательности чисел — по одному числу для каждого оцениваемого свойства.
- А вектором векторов является *матрица*. Если каждый вектор описывает признаки одного объекта в наборе данных, то матрица описывает признаки всех объектов, при этом каждый элемент вектора является узлом, представляющим собой список признаков одного объекта.