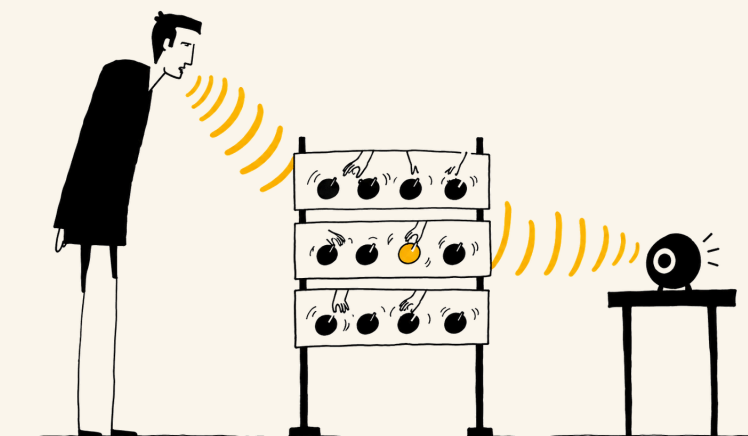


Погружение в глубокое обучение



ton Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola · 20

Иллюстрированные аудиолекции

Lecturized

Погружение в глубокое обучение

Иллюстрированные аудиолекции

Aston Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola · 2023

Пересказано в иллюстрированных лекциях Lecturized
lecturized.com

По оригиналу: Dive into Deep Learning (Cambridge University Press, 2023), d2l.ai

Содержание

Введение

Пример для начала

Ключевые составляющие

Модели и целевые функции

Алгоритмы оптимизации и обучение с учителем

Регрессия

Классификация

Разметка тегами, поиск и рекомендательные системы

Обучение на последовательностях

Обучение без учителя и самообучение

Взаимодействие со средой и обучение с подкреплением

Истоки

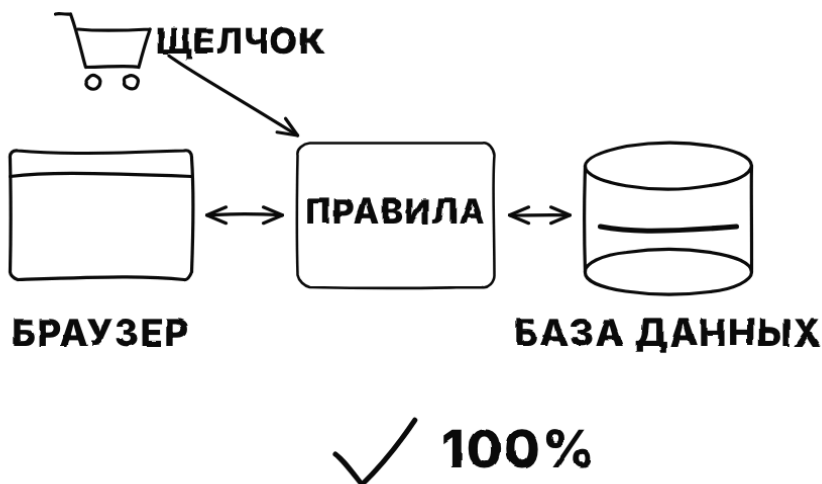
Путь к глубокому обучению

Истории успеха

Суть глубокого обучения

Введение

До недавнего времени почти любая компьютерная программа, с которой вы могли иметь дело в обычный день, была написана как жёсткий набор правил, точно предписывающих, как ей себя вести. Допустим, мы хотим написать приложение для управления интернет-магазином. Посоветовавшись несколько часов у доски, мы, вероятно, придём к общим чертам рабочего решения: во-первых, пользователи взаимодействуют с приложением через интерфейс в веб-браузере или мобильном приложении; во-вторых, наше приложение работает с промышленной базой данных, чтобы отслеживать состояние каждого пользователя и хранить историю всех покупок; и в-третьих, в самом сердце приложения бизнес-логика (можно сказать, его мозг) задаёт набор правил, сопоставляющих каждой мыслимой ситуации то действие, которое должна совершить программа.



Магазин, построенный из правил

Чтобы построить мозг нашего приложения, мы могли бы перечислить все обычные события, которые программа должна обрабатывать. Например, всякий раз, когда покупатель нажимает кнопку, чтобы добавить товар в корзину, программа должна добавить запись в таблицу корзины базы данных, связав ID этого пользователя с ID выбранного товара. Затем мы могли бы попытаться пройти по всем возможным пограничным случаям, проверяя, подходят ли наши правила, и внося нужные изменения. Что произойдёт, если пользователь начнёт оформлять покупку с пустой корзиной? Хотя мало кому из разработчиков удаётся всё сделать правильно с первого раза (может понадобиться несколько пробных запусков, чтобы устранить недочёты), в большинстве случаев мы способны

написать такие программы и уверенно запустить их ещё до того, как увидим первого настоящего покупателя. Наша способность вручную проектировать автоматические системы, на которых работают действующие продукты, причём часто в совершенно новых ситуациях, — замечательное достижение человеческого ума. А если вы способны придумать решение, которое работает в 100% случаев, то о машинном обучении вам, как правило, беспокоиться не нужно.

К счастью для растущего сообщества исследователей машинного обучения, многие задачи, которые мы хотели бы автоматизировать, не так легко поддаются человеческой изобретательности. Представьте, что вы стоите у доски вместе с самыми умными людьми, каких знаете, но на этот раз решаете одну из следующих задач.



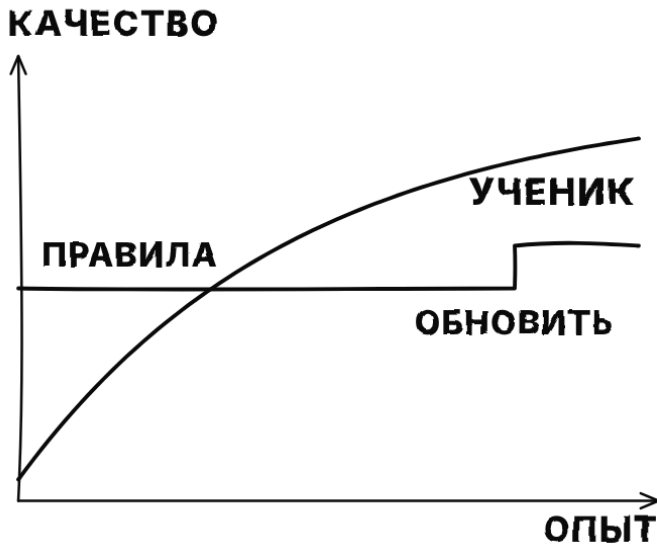
Задачи, для которых правил не написать

Написать программу, которая предсказывает завтрашнюю погоду по географическим данным, спутниковым снимкам и погоде за последние дни. Написать программу, которая получает фактический вопрос, заданный свободным текстом, и правильно на него отвечает. Написать программу, которая по изображению находит каждого изображённого на нём человека и обводит его контуром. Написать программу, которая показывает пользователям товары, которые им, скорее всего, понравятся, но на которые они вряд ли наткнулись бы, просто просматривая каталог.

Над такими задачами даже лучшие программисты долго бились бы, пытаясь написать решение с нуля. Причины бывают

разными. Иногда нужна нам программа следует закономерности, которая меняется со временем, так что единственно верного ответа просто нет! В таких случаях любое удачное решение должно плавно подстраиваться под меняющийся мир. В других случаях связь (скажем, между пикселями и абстрактными категориями) может оказаться слишком сложной: она требует тысяч или миллионов вычислений и подчиняется неизвестным нам принципам. В распознавании изображений точная последовательность шагов, нужная для решения задачи, лежит за пределами нашего сознательного понимания, хотя наше подсознание выполняет эту задачу без малейших усилий.

Машинное обучение — это наука об алгоритмах, способных учиться на опыте. Чем больше опыта накапливает алгоритм машинного обучения, обычно в виде данных наблюдений или взаимодействий со средой, тем лучше он работает. Сравните это с нашим детерминированным интернет-магазином: он следует одной и той же бизнес-логике, сколько бы опыта ни накопилось, пока сами разработчики не узнают что-то новое и не решат, что пора обновить программу. В этой книге мы научим вас основам машинного обучения, уделяя особое внимание глубокому обучению — мощному набору методов, который движет прогрессом в самых разных областях: в компьютерном зрении, обработке естественного языка, здравоохранении и геномике.



Обучение улучшается с опытом

Вопросы

1. Когда, согласно тексту, вам, как правилу, не нужно беспокоиться о машинном обучении?
 - a. Когда программа будет работать в веб-браузере
 - b. Когда вы можете придумать правила, которые работают в 100% случаев
 - c. Когда программа должна хранить записи о покупках
 - d. Когда разработчики ещё не видели ни одного настоящего покупателя

2. Почему, по первой из причин, названных в тексте, неизменный набор правил не справится с подбором товаров, которые понравятся покупателям?
 - a. Потому что база данных не может хранить достаточно товаров
 - b. Потому что программа должна работать и на телефонах, и на компьютерах
 - c. Потому что закономерность может меняться со временем, и единственно верного ответа нет
 - d. Потому что покупатели никогда не нажимают на рекомендованные товары
3. Почему распознавание изображений трудно запрограммировать вручную, хотя люди делают это без усилий?
 - a. Точная последовательность шагов лежит за пределами нашего сознательного понимания
 - b. Картинки занимают слишком много памяти
 - c. Лишь немногие люди надёжно узнают лица
 - d. Изображения меняются слишком медленно, чтобы на них учиться
4. Чем алгоритм машинного обучения отличается от интернет-магазина, описанного в тексте?
 - a. Внутри него вообще нет правил
 - b. Он работает быстрее на том же компьютере
 - c. Для его создания вовсе не нужны разработчики
 - d. Качество его работы растёт по мере накопления опыта

Ответы

1. b) Если вы способны придумать решение, которое работает в 100% случаев, о машинном обучении вам, как правило, беспокоиться не нужно.

2. c) Иногда нужная нам закономерность меняется со временем, поэтому любое удачное решение должно плавно подстраиваться под меняющийся мир.

3. a) Точные шаги, нужные для решения задачи, лежат за пределами нашего сознательного понимания, хотя подсознание выполняет её без усилий.

4. d) Алгоритм машинного обучения улучшается по мере накопления опыта, а магазин следует одной и той же бизнес-логике, пока разработчики его не обновят.

Задачи

1. Возьмите задачу предсказания завтрашней погоды по географическим данным, спутниковым снимкам и погоде прошлых дней. Опираясь на две причины, названные в тексте, объясните, почему программистам было бы трудно записать её решение в виде неизменного набора правил.

Подсказка: Подумайте, остаётся ли закономерность одной и той же со временем и насколько сложна связь между снимками и прогнозом.

Ответ: Связь между спутниковыми снимками, прошлой погодой и завтрашней погодой слишком сложна, чтобы её расписать: она требует огромного числа вычислений по принципам, которые никто не может записать в виде правил. К тому же закономерности могут меняться со временем, так что однажды записанный ответ быстро

отстанет; удачное решение должно подстраиваться под меняющийся мир.

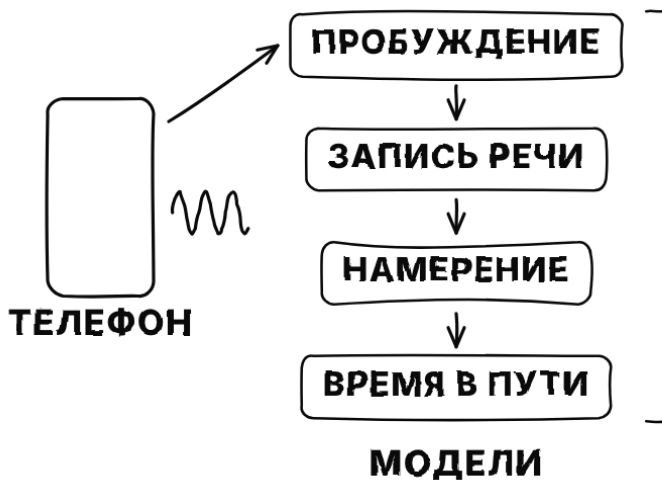
2. Своими словами объясните разницу между тем, как улучшается интернет-магазин, и тем, как улучшается алгоритм машинного обучения.

Подсказка: Спросите себя, кто учится в каждом из случаев.

Ответ: Магазин следует одной и той же бизнес-логике, сколько бы опыта ни накопилось; он меняется, только когда сами разработчики что-то узнают и обновляют программу. Алгоритм машинного обучения улучшается сам по мере того, как набирает опыт, например данные наблюдений или взаимодействия со средой.

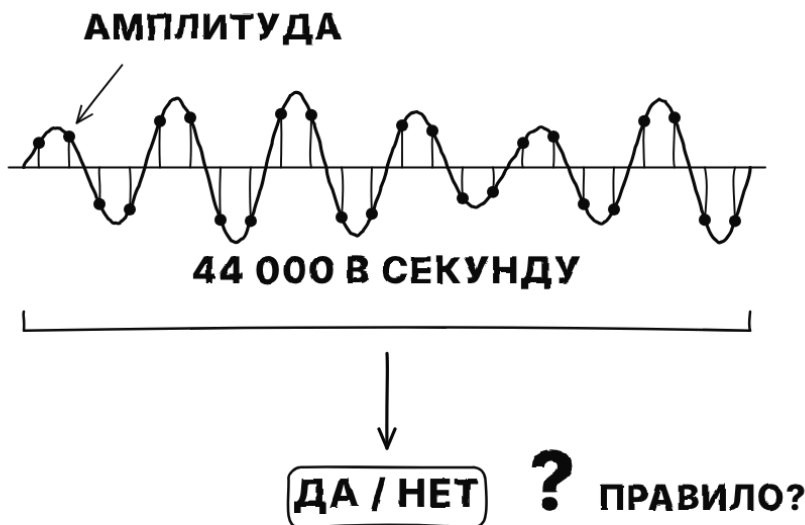
Пример для начала

Прежде чем сесть за работу, авторам этой книги, как и большей части работающих людей, нужно было выпить кофе. Мы сели в машину и поехали. Взяв iPhone, Алекс сказал «Hey Siri», разбудив систему распознавания голоса. Затем Му скомандовал: «маршрут до кофейни Blue Bottle». Телефон быстро вывел на экран запись его команды. Он также понял, что мы просим проложить маршрут, и запустил приложение «Карты», чтобы выполнить нашу просьбу. Запустившись, «Карты» нашли несколько маршрутов. Рядом с каждым маршрутом телефон показал предсказанное время в пути. Хотя эта история придумана ради наглядности, она показывает, что всего за несколько секунд наше обычное общение со смартфоном может задействовать несколько моделей машинного обучения.



Несколько моделей в одном телефоне

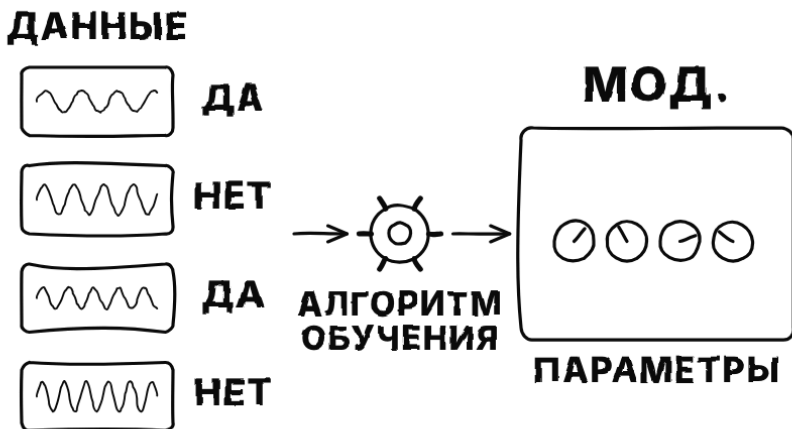
Представьте, что вам нужно всего лишь написать программу, которая откликается на слово пробуждения вроде «Alexa», «OK Google» или «Hey Siri». Попробуйте написать её, сидя одни в комнате, где нет ничего, кроме компьютера и редактора кода, как показано на рисунке. Как бы вы написали такую программу, исходя из первых принципов? Подумайте... задача трудная. Каждую секунду микрофон собирает примерно 44 000 отсчётов. Каждый отсчёт — это измерение амплитуды звуковой волны. Какое правило могло бы надёжно превращать фрагмент необработанного звука в уверенное предсказание, да или нет, есть ли в этом фрагменте слово пробуждения? Если вы застряли, не волнуйтесь. Мы тоже не знаем, как написать такую программу с нуля. Именно поэтому мы и используем машинное обучение.



От необработанного звука к «да» или «нет»

Вот в чём хитрость. Часто, даже когда мы не знаем, как явно объяснить компьютеру, как получать из входа выход, мы сами всё же способны выполнить эту умственную работу. Иными словами, даже если вы не знаете, как запрограммировать компьютер на распознавание слова «Alexa», сами вы его узнать способны. Пользуясь этой способностью, мы можем собрать огромный набор данных из примеров звуковых фрагментов с метками, указывающими, в каких фрагментах есть слово пробуждения. В нынешнем господствующем подходе к машинному обучению мы не пытаемся явно спроектировать систему, распознающую слова пробуждения. Вместо этого мы задаём гибкую программу, поведение которой определяется набором параметров. Затем мы с помощью набора данных находим наилучшие значения

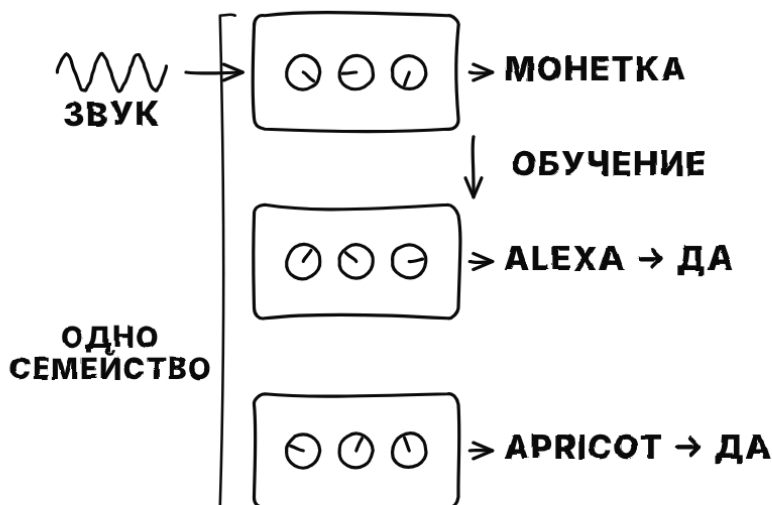
параметров, то есть те, что улучшают работу программы по выбранной мере качества.



Данные, ручки, алгоритм обучения

Параметры можно представлять себе как ручки, которые мы крутим, меняя поведение программы. Когда параметры зафиксированы, мы называем программу моделью. Множество всех различных программ (отображений входа в выход), которые мы можем получить, лишь меняя параметры, называется семейством моделей. А «метапрограмма», которая по нашему набору данных выбирает параметры, называется алгоритмом обучения.

Прежде чем запустить алгоритм обучения, нужно точно поставить задачу: определить, какими именно будут вход и выход, и выбрать подходящее семейство моделей. В нашем случае модель получает на вход фрагмент звука, а на выходе выбирает между «да» и «нет». Если всё пойдёт по плану, догадки модели о том, есть ли во фрагменте слово пробуждения, обычно будут верными.

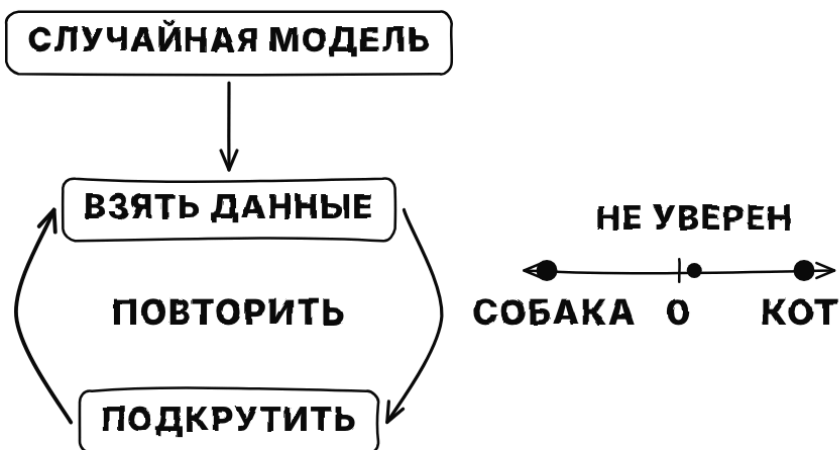


Одно семейство, много настроек

Если мы выбрали правильное семейство моделей, должно существовать такое положение ручек, при котором модель выдаёт «да» каждый раз, когда слышит слово «Аlexa». Поскольку выбор слова пробуждения произволен, нам, вероятно, понадобится достаточно богатое семейство моделей, чтобы при

другом положении ручек оно выдавало «да» только на слово «Apricot». Мы ожидаем, что одно и то же семейство моделей подойдёт и для распознавания «Alexa», и для распознавания «Apricot», потому что интуитивно это похожие задачи. Однако может понадобиться совсем другое семейство моделей, если мы захотим работать с принципиально иными входами или выходами, скажем, переводить изображения в подписи или английские предложения в китайские.

Как вы, наверное, догадываетесь, если просто выставить все ручки наугад, наша модель вряд ли узнает «Alexa», «Apricot» или любое другое английское слово. В машинном обучении обучение — это процесс, в котором мы находим правильное положение ручек, добиваясь от модели нужного поведения. Иными словами, мы обучаем модель на данных. Как показано на рисунке, процесс обучения обычно выглядит так.



Цикл обучения, затем оценка «кошка»

Во-первых, начинаем со случайно инициализированной модели, которая не умеет ничего полезного. Во-вторых, берём часть данных (например, звуковые фрагменты с соответствующими метками «да» или «нет»). В-третьих, подкручиваем ручки так, чтобы модель лучше справлялась с этими примерами. В-четвёртых, повторяем шаги два и три, пока модель не станет великолепной.

Итак, вместо того чтобы писать распознаватель слова пробуждения, мы пишем программу, которая может научиться распознавать слова пробуждения, если дать ей большой размеченный набор данных. Такое задание поведения программы через предъявление ей набора данных можно

считать программированием с помощью данных. То есть мы можем «запрограммировать» детектор кошек, дав нашей системе машинного обучения много примеров кошек и собак. Так детектор со временем научится выдавать очень большое положительное число, если это кошка, очень большое отрицательное, если это собака, и что-то ближе к нулю, если он не уверен. Это лишь самая верхушка того, на что способно машинное обучение. Глубокое обучение, которое мы подробнее объясним позже, — лишь один из многих популярных методов решения задач машинного обучения.

Вопросы

1. Почему, согласно тексту, мы можем собрать набор данных для распознавания слова пробуждения, хотя не можем написать саму программу?
 - a. Потому что микрофоны размечают звук автоматически
 - b. Потому что мы сами способны узнать слово пробуждения и пометить фрагменты
 - c. Потому что у слов пробуждения постоянная форма звуковой волны
 - d. Потому что в телефоне уже есть список всех слов
2. Что превращает гибкую программу в то, что в тексте называется моделью?
 - a. Сбор набора данных для неё
 - b. Выбор меры качества
 - c. Фиксация значений её параметров

- d. Запуск на смартфоне
3. Почему одно и то же семейство моделей может подойти и для «Alexa», и для «Apricot», но не для превращения изображений в подписи?
- a. Для подписей нужно гораздо больше данных, чем для слов
 - b. Изображения нельзя хранить в виде чисел
 - c. «Apricot» произнести легче, чем «Alexa»
 - d. У подписывания изображений принципиально другие входы и выходы
4. Что происходит в описанном в тексте процессе обучения после того, как ручки подкрутили на некоторых примерах?
- a. Берут ещё данных и снова подкручивают, пока модель не станет хорошей
 - b. Снова выставляют все ручки наугад
 - c. Окончательно фиксируют параметры и останавливаются
 - d. Заменяют семейство моделей на более богатое

Ответы

1. **b)** Даже если вы не знаете, как запрограммировать компьютер на распознавание слова, сами вы способны его узнать, поэтому можете собрать звуковые фрагменты с метками.
2. **c)** Параметры — это ручки; когда параметры зафиксированы, мы называем программу моделью.
3. **d)** Два слова пробуждения — похожие задачи, но для принципиально иных входов или выходов, например изображений и подписей, может понадобиться другое семейство.

4. а) Шаги такие: начать со случайной модели, взять данные, подкрутить ручки, чтобы лучше справиться с ними, и повторять шаги два и три, пока модель не станет великолепной.

Разобранные примеры

Сколько слышит микрофон Микрофон собирает примерно 44 000 отсчётов в секунду. Модели слова пробуждения дают фрагмент необработанного звука длиной 2 секунды. Сколько измерений амплитуды она получает на вход?

Каждый отсчёт — это одно измерение амплитуды звуковой волны, поэтому считаем отсчёты.

За одну секунду микрофон делает около 44 000 отсчётов.

За 2 секунды он делает $2 \times 44\,000 = 88\,000$ отсчётов.

Модель должна превратить все эти числа в один ответ, «да» или «нет», поэтому простым правилом здесь не обойтись.

Ответ: Около 88 000 измерений амплитуды

Задачи

1. Детектор кошек выдаёт очень большое положительное число для кошки и очень большое отрицательное для собаки. Что он должен выдать для размытой фотографии, в которой не уверен, и почему?

Подсказка: Подумайте о двух концах шкалы и о том, где между ними находится сомнение.

Ответ: Что-то близкое к нулю: большое положительное число означает кошку, большое отрицательное — собаку, а значение около нуля — середина шкалы, где детектор не уверен.

2. Своими словами объясните, что в тексте имеется в виду под «программированием с помощью данных».

Подсказка: Сравните, как вы сами пишете правила и как показываете программе примеры.

Ответ: Вместо того чтобы самим писать правила распознавателя слова пробуждения или детектора кошек, мы пишем гибкую программу и показываем ей большой размеченный набор данных; затем алгоритм обучения выставляет её параметры. Поведение программы определяется примерами, которые мы ей предъявляем, так что программирует её сами данные.

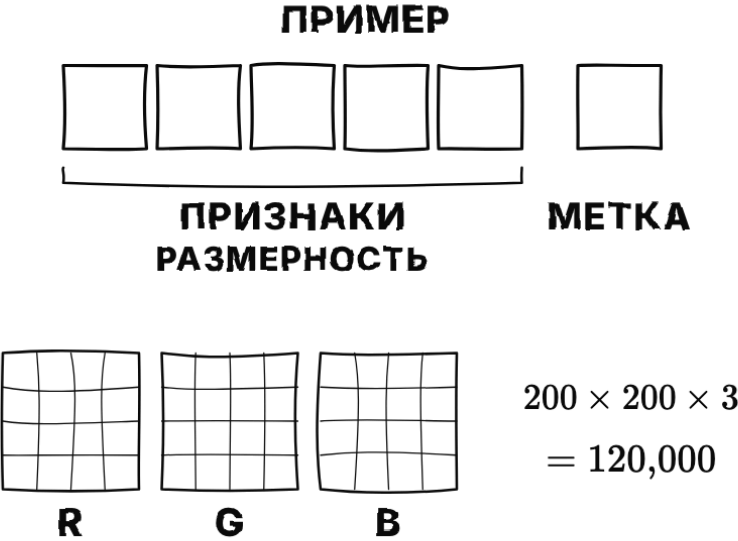
Ключевые составляющие

В примере со словом пробуждения мы описали набор данных из звуковых фрагментов и двоичных меток и в общих чертах показали, как можно обучить модель приближать отображение фрагментов в классы. Задача такого рода, где мы по известным входам пытаемся предсказать определённую неизвестную метку, располагая набором данных из примеров с известными метками, называется обучением с учителем. Это лишь один из многих видов задач машинного обучения. Прежде чем перейти к другим видам, мы хотим подробнее осветить несколько основных составляющих, которые будут сопровождать нас, за какую бы задачу машинного обучения мы ни взялись.

Во-первых, данные, на которых мы можем учиться. Во-вторых, модель того, как преобразовывать данные. В-третьих, целевая функция, которая численно оценивает, насколько хорошо (или плохо) работает модель. В-четвёртых, алгоритм, который подстраивает параметры модели, чтобы оптимизировать целевую функцию.

Само собой разумеется, что наукой о данных невозможно заниматься без данных. Можно было бы потратить сотни страниц на размышления о том, что такое данные, но пока мы сосредоточимся на ключевых свойствах наборов данных, с которыми будем иметь дело. Как правило, нас интересует собрание примеров. Чтобы с данными можно было с пользой работать, обычно нужно придумать для них подходящее числовое представление. Каждый пример (его называют также

точкой данных, экземпляром данных или образцом) обычно состоит из набора атрибутов, называемых признаками (иногда — ковариатами или входами), на основе которых модель должна делать свои предсказания. В задачах обучения с учителем наша цель — предсказать значение особого атрибута, называемого меткой (или целевой переменной), который не входит во вход модели.



Признаки и метка

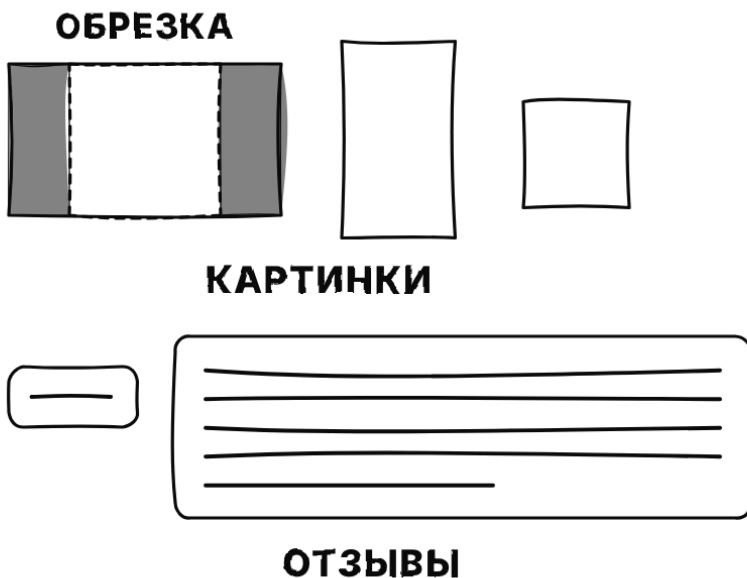
Если бы мы работали с изображениями, каждый пример мог бы состоять из отдельной фотографии (признаки) и числа, указывающего, к какой категории относится фотография (метка). Фотография представлялась бы численно в виде трёх сеток чисел, задающих яркость красного, зелёного и синего света в каждом

пикселе. Например, цветная фотография размером 200×200 пикселей состояла бы из $200 \times 200 \times 3 = 120\,000$ чисел.

Или мы могли бы работать с электронными медицинскими картами и решать задачу предсказания вероятности того, что пациент проживёт следующие 30 дней. Тогда нашими признаками мог бы быть набор легкодоступных атрибутов и часто снимаемых показателей, включая возраст, жизненные показатели, сопутствующие заболевания, текущие лекарства и недавние процедуры. Метка, доступная для обучения, была бы двоичным значением, указывающим, прожил ли каждый пациент из исторических данных 30 дней.

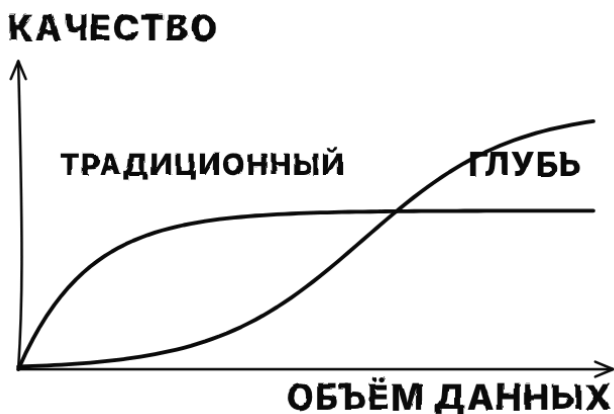
В таких случаях, когда каждый пример описывается одним и тем же числом числовых признаков, мы говорим, что входы — это векторы фиксированной длины, а (постоянную) длину векторов называем размерностью данных. Как вы понимаете, входы фиксированной длины удобны: одной сложностью меньше. Однако не все данные легко представить векторами фиксированной длины. Можно ожидать, что снимки с микроскопа получены на стандартном оборудовании, но нельзя ожидать, что все изображения, собранные в интернете, будут одного разрешения и формы. Изображения можно было бы обрезать до стандартного размера, но такая стратегия помогает лишь до определённого предела. Мы рискуем потерять информацию в обрезанных частях. Более того, текстовые данные ещё упорнее сопротивляются представлению фиксированной длины. Возьмите отзывы покупателей на сайтах электронной торговли вроде Amazon, IMDb и TripAdvisor. Одни короткие: «гадость!». Другие растягиваются на страницы. Одно из главных преимуществ глубокого обучения перед традиционными

методами — сравнительная лёгкость, с которой современные модели справляются с данными переменной длины.



Данные, которые не выстроить в ряд

Как правило, чем больше у нас данных, тем легче наша работа. Когда данных больше, мы можем обучать более мощные модели и меньше опираться на заранее сделанные предположения. Переход от (сравнительно) малых данных к большим во многом обеспечил успех современного глубокого обучения. Скажем прямо: многие из самых впечатляющих моделей глубокого обучения без больших наборов данных не работают. Некоторые другие могут работать и на малых данных, но не лучше традиционных подходов.



ПЛОХИЕ ДАННЫЕ

МОД.

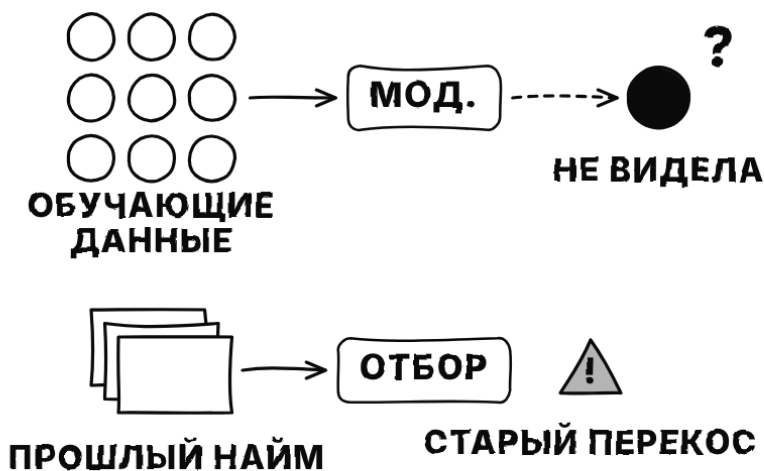
ПЛОХОЙ ОТВЕТ

МУСОР → МУСОР

Больше данных — больше модели

Наконец, недостаточно иметь много данных и умно их обрабатывать. Нужны правильные данные. Если данные полны ошибок или выбранные признаки не позволяют предсказать интересующую нас целевую величину, обучение провалится. Ситуацию хорошо описывает избитая фраза: мусор на входе — мусор на выходе. Более того, плохое качество предсказаний — не единственное возможное последствие. В чувствительных применениях машинного обучения, таких как предиктивная работа полиции, отбор резюме и модели риска при кредитовании, нужно быть особенно внимательными к последствиям мусорных данных. Одна из частых ошибок связана с наборами данных, в которых некоторые группы людей не представлены в обучающих данных. Представьте, что

применяете систему распознавания рака кожи, которая никогда раньше не видела тёмной кожи. Сбой может произойти и тогда, когда данные не только недостаточно представляют некоторые группы, но и отражают общественные предрассудки. Например, если на прошлых решениях о найме обучить модель, которая будет отбирать резюме, модели машинного обучения могут невольно усвоить и автоматизировать исторические несправедливости. Заметьте, что всё это может произойти без всякого злого умысла со стороны специалиста по данным и даже без его ведома.



Пропущенные группы, несправедливые результаты

Вопросы

1. Какая часть примера является меткой в задаче обучения с учителем на фотографиях?
 - a. Значения яркости красного, зелёного и синего света
 - b. Число, указывающее категорию фотографии
 - c. Ширина и высота фотографии в пикселях
 - d. Сетка расположения пикселей
2. Почему обрезка изображений из интернета до стандартного размера помогает лишь до определённого предела?
 - a. Обрезанные изображения нельзя представить числами
 - b. Обрезка меняет цвета пикселей
 - c. Мы рискуем потерять информацию в обрезанных частях
 - d. Обрезанные изображения всегда слишком малы, чтобы на них учиться
3. Почему текст ещё труднее, чем изображения, представить векторами фиксированной длины?
 - a. Отзывы бывают от нескольких слов до многих страниц
 - b. В тексте нет числовой информации
 - c. Текст всегда короче изображения
 - d. Отзывы пишут разными алфавитами
4. Модель найма обучили на прошлых решениях о найме, в которых отражались предрассудки. О чём предупреждает текст?

- a. Модель откажется делать предсказания
- b. Модель исправит прошлые несправедливости
- c. Модель ошибётся, только если специалист по данным этого захочет
- d. Модель может усвоить и автоматизировать исторические несправедливости

Ответы

1. **b)** Каждый пример может состоять из фотографии — признаков — и числа, указывающего, к какой категории она относится, — метки, которая не входит во вход модели.
2. **c)** Обрезка даёт входы фиксированной длины, но мы рискуем потерять информацию в обрезанных частях.
3. **a)** Отзывы покупателей сильно различаются: одни короткие, как «гадость!», другие растягиваются на страницы.
4. **d)** Если на прошлых решениях о найме обучить модель для отбора резюме, она может невольно усвоить и автоматизировать исторические несправедливости, даже если никто об этом не знает.

Разобранные примеры

Сколько чисел в фотографии Цветная фотография имеет размер 200×200 пикселей. Из скольких чисел она состоит, если её представить тремя сетками значений яркости красного, зелёного и синего света?

В каждой сетке по одному значению на пиксель: $200 \times 200 = 40\,000$ значений.

Сеток три: для красного, зелёного и синего света.

Значит, фотография — это $40\,000 \times 3 = 120\,000$ чисел.

Ответ: 120 000 чисел

Задачи

1. Каждый пример в наборе цветных фотографий размером 100×100 пикселей — вектор фиксированной длины. Какова размерность данных?

Подсказка: Посчитайте значения на пиксель и число пикселей, как для фотографии 200×200 .

Ответ: 30 000: пикселей $100 \times 100 = 10\,000$, у каждого три значения для красного, зелёного и синего, а постоянная длина векторов и есть размерность.

2. Для задачи предсказания, проживёт ли пациент следующие 30 дней, назовите несколько признаков и метку и скажите, какого рода значение у метки.

Подсказка: Признаки — то, что видит модель; метка — то, что она должна предсказать.

Ответ: Признаками могут быть возраст, жизненные показатели, сопутствующие заболевания, текущие лекарства и недавние процедуры; метка — прожил ли каждый пациент из исторических данных 30 дней, это двоичное значение.

3. Своими словами объясните, что означает для машинного обучения выражение «мусор на входе — мусор на выходе», и назовите одно последствие помимо плохих предсказаний.

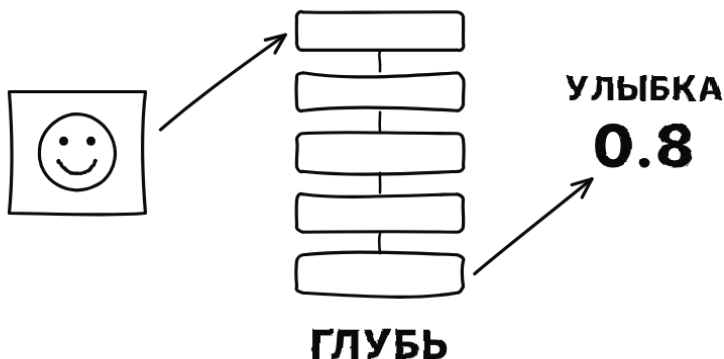
Подсказка: Подумайте об ошибочных данных, бесполезных признаках и группах, которых нет в данных.

Ответ: Если данные полны ошибок или признаки не предсказывают целевую величину, обучение провалится, как бы умно ни обрабатывались данные. Помимо плохих предсказаний, система

может несправедливо подводить людей, например распознаватель рака кожи, который никогда не видел тёмной кожи, или система отбора резюме, автоматизирующая прошлые предрассудки.

Модели и целевые функции

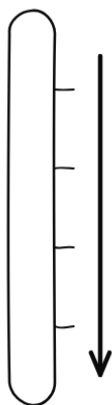
Большая часть машинного обучения так или иначе связана с преобразованием данных. Мы можем захотеть построить систему, которая принимает фотографии и предсказывает, насколько человек на них улыбается. Или мы можем захотеть принимать набор показаний датчиков и предсказывать, насколько эти показания нормальны или аномальны. Моделью мы называем вычислительный механизм, который принимает данные одного типа и выдаёт предсказания, возможно, другого типа. В частности, нас интересуют статистические модели, которые можно оценить по данным. Хотя простые модели прекрасно справляются с достаточно простыми задачами, задачи, на которых сосредоточена эта книга, выходят за пределы возможностей классических методов. Глубокое обучение отличается от классических подходов прежде всего набором мощных моделей, на которых оно сосредоточено. Эти модели состоят из многих последовательных преобразований данных, сцепленных друг с другом сверху вниз, отсюда и название — глубокое обучение. На пути к глубоким моделям мы обсудим и некоторые более традиционные методы.



Модель, мелкая и глубокая

Ранее мы представили машинное обучение как обучение на опыте. Под обучением здесь мы понимаем улучшение в какой-то задаче со временем. Но кто решает, что считать улучшением? Представьте, что мы предлагаем обновить модель, а кто-то не согласен, что наше предложение было улучшением.

ПОТЕРИ



ЛУЧШЕ

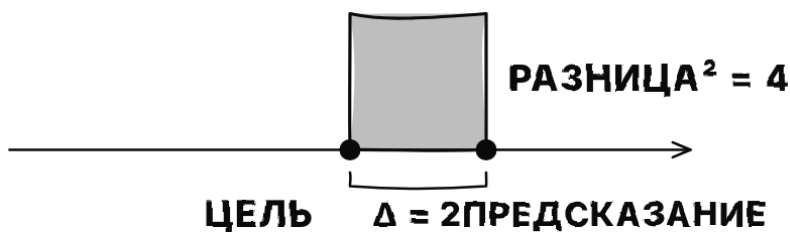
оценка $\uparrow$

потери = $-$ оценка $\downarrow$

Меньше — лучше

Чтобы построить формальную математическую систему обучающихся машин, нам нужны формальные меры того, насколько хороши (или плохи) наши модели. В машинном обучении и вообще в оптимизации их называют целевыми функциями. По соглашению мы обычно определяем целевые функции так, чтобы меньше означало лучше. Это всего лишь соглашение. Можно взять любую функцию, у которой больше — лучше, и, поменяв знак, превратить её в новую функцию, по сути такую же, но у которой меньше — лучше. Поскольку мы выбираем, что меньше — лучше, такие функции иногда называют функциями потерь.

При предсказании числовых значений самая распространённая функция потерь — квадрат ошибки, то есть квадрат разности между предсказанием и истинным целевым значением. Для классификации самая распространённая цель — минимизировать долю ошибок, то есть долю примеров, на которых наши предсказания расходятся с истиной. Некоторые цели (например, квадрат ошибки) легко оптимизировать, а другие (например, долю ошибок) трудно оптимизировать напрямую из-за недифференцируемости или других сложностей. В таких случаях обычно оптимизируют суррогатную цель.

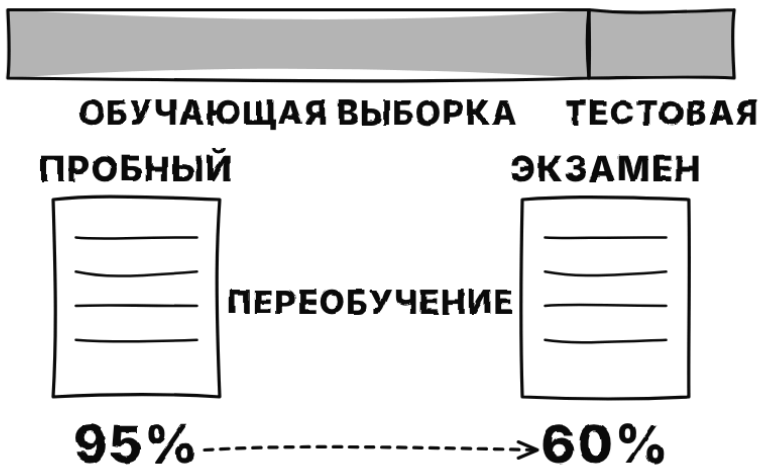


ОШИБКИ 3/10
БЕРЁМ СУРРОГАТ

Квадрат ошибки и доля ошибок

Во время оптимизации мы рассматриваем потери как функцию параметров модели, а обучающий набор данных считаем

постоянным. Мы находим наилучшие значения параметров модели, минимизируя потери на наборе из некоторого числа примеров, собранных для обучения. Однако хорошие результаты на обучающих данных не гарантируют хороших результатов на данных, которых модель не видела. Поэтому обычно имеющиеся данные делят на две части: обучающий набор данных (или обучающую выборку) для подбора параметров модели и тестовый набор данных (или тестовую выборку), который откладывают для оценки. В итоге мы обычно сообщаем, как модели работают на обеих частях. Качество на обучающих данных можно сравнить с баллами, которые студент получает на пробных экзаменах, готовясь к настоящему выпускному экзамену. Даже если результаты обнадеживают, это не гарантирует успеха на выпускном экзамене. Занимаясь, студент может начать заучивать пробные вопросы и казаться знатоком темы, но растеряться перед вопросами, которых он раньше не видел, на настоящем экзамене. Когда модель хорошо работает на обучающей выборке, но не обобщается на невиданные данные, мы говорим, что она переобучается на обучающих данных.



Обучающая выборка, тестовая выборка, переобучение

Вопросы

1. Что, согласно тексту, прежде всего отличает глубокое обучение от классических подходов?
 - a. Ему не нужна целевая функция
 - b. Его мощные модели сцепляют много последовательных преобразований данных
 - c. Оно работает только с фотографиями
 - d. Оно никогда не использует статистические модели

2. У вас есть цель, где больше — лучше, например точность. Как привести её к обычному соглашению?
- a. Возвести её в квадрат
 - b. Разделить её на число примеров
 - c. Поменять у неё знак, чтобы меньше стало лучше
 - d. Заменить её долей ошибок на тестовой выборке
3. Почему на практике вместо доли ошибок часто оптимизируют суррогатную цель?
- a. Долю ошибок трудно оптимизировать напрямую из-за недифференцируемости
 - b. Долю ошибок можно использовать только для числовых предсказаний
 - c. Доля ошибок всегда больше квадрата ошибки
 - d. Доля ошибок не учитывает обучающие данные
4. Модель показывает очень хорошие результаты на обучающей выборке, но плохие на отложенной тестовой. Как это называется?
- a. Использование суррогатной цели
 - b. Смена знака функции потерь
 - c. Рассмотрение набора данных как постоянного
 - d. Переобучение на обучающих данных

Ответы

1. b) Глубокое обучение отличается главным образом своими мощными моделями, которые состоят из многих последовательных сцепленных преобразований данных, отсюда и название.

2. c) Любую функцию, у которой больше — лучше, можно превратить в по сути такую же, у которой меньше — лучше, поменяв знак.

3. a) Некоторые цели, например долю ошибок, трудно оптимизировать напрямую из-за недифференцируемости или других сложностей, поэтому оптимизируют суррогат.

4. d) Когда модель хорошо работает на обучающей выборке, но не обобщается на невиданные данные, она переобучается, как студент, заучивший пробные вопросы.

Разобранные примеры

Квадрат ошибки и доля ошибок Модель предсказывает значение 7, а истинное целевое значение равно 4. Каков квадрат ошибки? Отдельно: классификатор ошибается на 3 из 20 тестовых примеров; какова доля ошибок?

Квадрат ошибки — квадрат разности между предсказанием и целью:

$$7 - 4 = 3.$$

Возведение в квадрат даёт $3 \times 3 = 9$.

Доля ошибок — это доля примеров, на которых предсказания расходятся с истиной.

Это 3 из 20, то есть $3 \div 20 = 0,15$.

Ответ: Квадрат ошибки 9; доля ошибок 0,15 (15%)

Задачи

1. Предсказание 10 сравнивают с истинным значением 12. Каков квадрат ошибки, и будет ли у предсказания 14 квадрат ошибки больше или такой же?

Подсказка: Возведите разность в квадрат; знак разности при этом исчезает.

Ответ: $10 - 12 = -2$, в квадрате 4. Предсказание 14 отличается на +2, что тоже даёт 4, так что квадрат ошибки такой же.

2. Своими словами объясните, зачем данные делят на обучающую и тестовую выборки, используя сравнение со студентом и экзаменом.

Подсказка: Что не доказывает хороший балл на пробных экзаменах?

Ответ: Хорошие результаты на обучающих данных не гарантируют хороших результатов на невиданных. Студент может заучить пробные вопросы и растеряться на новых вопросах экзамена; так же и модель обучают на обучающей выборке и проверяют на отложенной тестовой, чтобы увидеть, обобщается ли она, а не переобучается.

Алгоритмы оптимизации и обучение с учителем

Когда у нас есть источник данных и их представление, модель и чётко определённая целевая функция, нам нужен алгоритм, способный искать наилучшие параметры для минимизации функции потерь. Популярные алгоритмы оптимизации для глубокого обучения основаны на подходе, который называется градиентным спуском. Вкратце, на каждом шаге этот метод проверяет для каждого параметра, как изменились бы потери на обучающей выборке, если бы этот параметр чуть-чуть изменить. Затем он сдвигает параметр в ту сторону, в которой потери уменьшаются.



Градиентный спуск, шаг за шагом

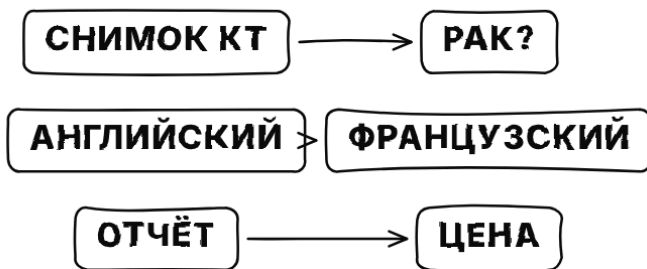
Задача со словом пробуждения из нашего вводного примера — лишь одна из многих, которые может решать машинное обучение. Чтобы ещё больше заинтересовать читателя и дать нам общий язык, который пригодится на протяжении всей книги, мы сейчас дадим широкий обзор задач машинного обучения.

Обучение с учителем описывает задачи, в которых нам дан набор данных, содержащий и признаки, и метки, и нужно построить модель, предсказывающую метки по входным признакам. Каждая пара «признаки — метка» называется примером. Иногда, когда это ясно из контекста, мы можем называть примерами и набор входов, даже если соответствующие метки неизвестны. Учитель здесь появляется потому, что для выбора параметров мы

(учителя) даём модели набор данных из размеченных примеров. В вероятностных терминах нас обычно интересует оценка условной вероятности метки при данных входных признаках. Хотя это лишь одна из нескольких парадигм, на обучение с учителем приходится большинство успешных применений машинного обучения в промышленности. Отчасти это потому, что многие важные задачи можно чётко описать как оценку вероятности чего-то неизвестного по определённому набору имеющихся данных.



$$P(\text{метка} \mid \text{признаки})$$



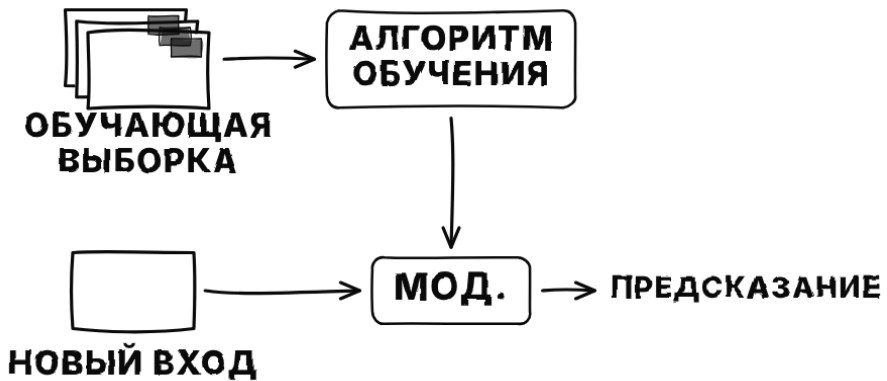
Предсказать метки по признакам

Предсказать по снимку компьютерной томографии, есть рак или нет. Предсказать правильный перевод на французский по

предложению на английском. Предсказать цену акции в следующем месяце по финансовой отчётности этого месяца.

Хотя все задачи обучения с учителем охватываются простым описанием «предсказать метки по входным признакам», само обучение с учителем может принимать разные формы и требовать множества решений о моделировании в зависимости (помимо прочего) от типа, размера и количества входов и выходов. Например, для обработки последовательностей произвольной длины и представлений в виде векторов фиксированной длины мы используем разные модели. Многие из этих задач мы подробно рассмотрим на протяжении книги.

Если говорить неформально, процесс обучения выглядит примерно так. Сначала берём большое собрание примеров с известными признаками и выбираем из него случайное подмножество, получая для каждого примера истинную метку. Иногда эти метки — уже собранные данные (например, умер ли пациент в течение следующего года?), а иногда приходится привлекать людей-разметчиков (например, чтобы распределить изображения по категориям). Вместе эти входы и соответствующие им метки составляют обучающую выборку. Мы подаём обучающий набор данных в алгоритм обучения с учителем — функцию, которая принимает на вход набор данных и выдаёт другую функцию: обученную модель. Наконец, мы можем подавать обученной модели входы, которых она раньше не видела, и использовать её выходы как предсказания соответствующих меток. Весь процесс изображён на рисунке.



От размеченных данных к модели

Вопросы

1. Что метод градиентного спуска проверяет для каждого параметра на каждом шаге?
 - a. Больше ли этот параметр всех остальных
 - b. Как изменились бы потери на обучающей выборке, если бы параметр чуть изменить
 - c. Встречается ли параметр в тестовой выборке
 - d. На скольких примерах обучался параметр

2. Откуда берётся «учитель» в обучении с учителем?
- a. От агента, действующего в среде
 - b. Из тестовой выборки, отложенной для оценки
 - c. Из размеченных примеров, которые мы даём для выбора параметров
 - d. От модели, которая сама проверяет свои ответы
3. Что, согласно тексту, алгоритм обучения с учителем принимает на вход и что выдаёт?
- a. Он принимает набор данных и выдаёт другую функцию — обученную модель
 - b. Он принимает один вход и выдаёт метку
 - c. Он принимает модель и выдаёт набор данных
 - d. Он принимает метки и выдаёт признаки

Ответы

1. **b)** На каждом шаге метод проверяет для каждого параметра, как изменились бы потери на обучающей выборке при малом изменении, и затем сдвигает его в ту сторону, где потери уменьшаются.
2. **c)** Мы, учителя, даём модели набор данных из размеченных примеров для выбора её параметров.
3. **a)** Алгоритм обучения с учителем — это функция, которая принимает на вход набор данных и выдаёт другую функцию: обученную модель.

Задачи

1. Для каждой задачи из текста скажите, что является входными признаками, а что меткой: обнаружение рака, перевод и

предсказание цены акции.

Подсказка: Каждая задача сформулирована как «предсказать ... по ...».

Ответ: Рак: признаки — снимок компьютерной томографии, метка — рак или не рак. Перевод: признаки — предложение на английском, метка — его правильный перевод на французский.

Акция: признаки — финансовая отчётность этого месяца, метка — цена акции в следующем месяце.

2. Своими словами опишите шаги процесса обучения с учителем, как их излагает текст, включая два способа получить метки.

Подсказка: Начните с большого собрания примеров с известными признаками.

Ответ: Берём большое собрание примеров с известными признаками, выбираем случайное подмножество и получаем для него истинные метки — либо из уже собранных данных, например умер ли пациент в течение года, либо от людей-разметчиков, которые, например, распределяют изображения по категориям. Эти входы и метки образуют обучающую выборку, которая поступает в алгоритм обучения с учителем, выдающий обученную модель. Затем модели подают новые, невиданные входы, а её выходы используют как предсказания их меток.

Регрессия

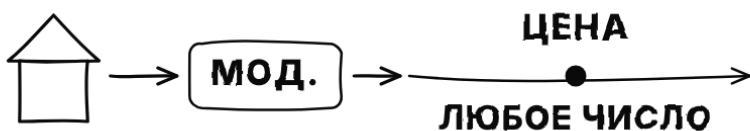
Пожалуй, самая простая для понимания задача обучения с учителем — регрессия. Возьмём, например, данные из базы продаж домов. Можно составить таблицу, в которой каждая строка соответствует отдельному дому, а каждый столбец — какому-то существенному атрибуту: площади дома в квадратных футах, числу спален, числу ванных комнат и числу минут ходьбы до центра города. В этом наборе данных каждым примером был бы конкретный дом, а соответствующим вектором признаков — одна строка таблицы. Если вы живёте в Нью-Йорке или Сан-Франциско и не возглавляете Amazon, Google, Microsoft или Facebook, вектор признаков вашего жилья (площадь, число спален, число ванных, расстояние пешком) может выглядеть примерно так: [600, 1, 1, 60]. Но если вы живёте в Питтсбурге, он может выглядеть скорее так: [3000, 4, 3, 10]. Подобные векторы признаков фиксированной длины необходимы большинству классических алгоритмов машинного обучения.

ОДИН ДОМ

	КВ. ФТ	СПАЛЬНИВАННЫЕ	МИН	
НЬЮ-ЙОРК	600	1	1	60
ПИТТСБУРГ	3000	4	3	10
ФИКС. ДЛИНА				

Дома как строки чисел

Регрессией задачу делает, по сути, вид целевой величины. Допустим, вы подыскиваете новый дом. Вы можете захотеть оценить справедливую рыночную стоимость дома по признакам вроде перечисленных выше. Данными здесь могут быть исторические объявления о продаже домов, а метками — наблюдавшиеся цены продажи. Когда метки принимают произвольные числовые значения (пусть даже в некотором интервале), мы называем это задачей регрессии. Цель — построить модель, предсказания которой близко приближают истинные значения меток.



СКОЛЬКО?

ОЦЕНКА ФИЛЬМА

ДНИ В БОЛЬНИЦЕ

ЧАСЫ ОПЕРАЦИИ

ОСАДКИ

Регрессия: метка — число

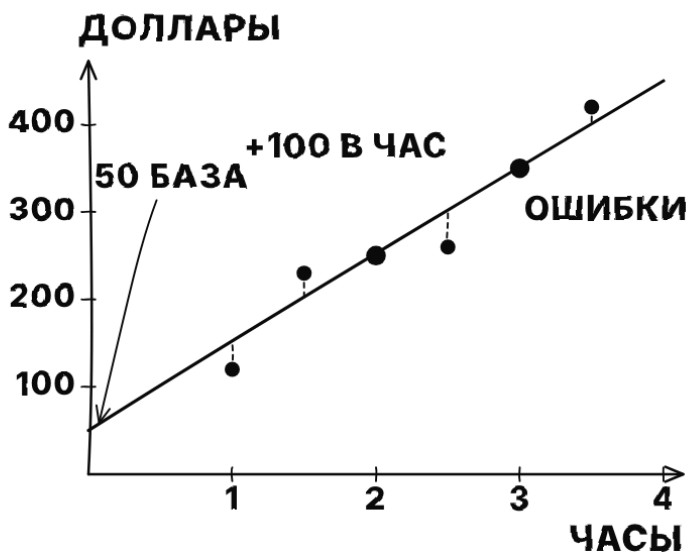
Многие практические задачи легко описать как задачи регрессии. Предсказание оценки, которую пользователь поставит фильму, можно рассматривать как задачу регрессии, и если бы вы в 2009 году придумали для этого отличный алгоритм, то могли бы выиграть приз Netflix в миллион долларов.

Предсказание того, сколько пациенты пробудут в больнице, — тоже задача регрессии. Хорошее эмпирическое правило: любая задача вида «сколько?» скорее всего окажется регрессией.

Например: сколько часов займёт эта операция? Сколько осадков выпадет в этом городе за следующие шесть часов?

Даже если вы никогда не имели дела с машинным обучением, вы, вероятно, уже решали задачу регрессии неформально.

Представьте, например, что вам чинили канализацию и мастер потратил 3 часа, вычищая грязь из труб. Потом он прислал вам счёт на 350 долларов. А теперь представьте, что ваш друг нанял того же мастера на 2 часа и получил счёт на 250 долларов. Если кто-то спросит вас, какой счёт ждать ему за предстоящую чистку, вы можете сделать разумные предположения, например, что больше часов работы стоит больше долларов. Вы можете также предположить, что есть некая базовая плата, а затем мастер берёт почасово. Если эти предположения верны, то по этим двум примерам вы уже можете определить, как мастер считает цену: 100 долларов в час плюс 50 долларов за выезд к вам домой. Если вы проследили за этим рассуждением, вы уже понимаете общую идею линейной регрессии.



В этом случае мы могли подобрать параметры, точно совпадающие с ценами мастера. Иногда это невозможно, например если часть разброса вызвана факторами, которые не входят в два ваших признака. В таких случаях мы будем стараться обучать модели, минимизирующие расстояние между нашими предсказаниями и наблюдаемыми значениями. В большинстве глав мы будем сосредоточены на минимизации функции потерь в виде квадрата ошибки. Как мы увидим позже, эта функция потерь соответствует предположению, что наши данные искажены гауссовым шумом.

Вопросы

1. Что, согласно тексту, делает задачу задачей регрессии?
 - a. Признаки образуют таблицу с одной строкой на пример
 - b. Метки принимают произвольные числовые значения
 - c. Модель должна быть линейной
 - d. Данные взяты из базы продаж
2. Какой из этих вопросов, по эмпирическому правилу из текста, скорее всего задача регрессии?
 - a. Спам это письмо или нет?
 - b. Какое из десяти животных на этой фотографии?
 - c. Сколько осадков выпадет в этом городе за следующие шесть часов?
 - d. Есть ли в этом фрагменте слово пробуждения?

3. Когда мы не можем найти параметры, точно совпадающие с наблюдаемыми значениями?
- а. Когда часть разброса вызвана факторами, не входящими в наши признаки
 - б. Когда у нас ровно два примера
 - с. Когда метки — суммы в долларах
 - д. Когда мастер берёт базовую плату

Ответы

1. **б)** Регрессией задачу делает вид целевой величины: когда метки принимают произвольные числовые значения, мы называем это регрессией.
2. **с)** Хорошее эмпирическое правило: любая задача вида «сколько?» скорее всего окажется регрессией.
3. **а)** Точное совпадение иногда невозможно, например если часть разброса вызвана факторами за пределами признаков; тогда мы минимизируем расстояние между предсказаниями и наблюдениями.

Разобранные примеры

Как мастер считает цену Мастер берёт 350 долларов за 3 часа работы и 250 долларов за 2 часа. Если считать, что есть базовая плата плюс фиксированная цена за час, как устроена цена и сколько будут стоить 5 часов?

Один лишний час (с 2 до 3 часов) добавляет $350 - 250 = 100$ долларов, значит, почасовая ставка — 100 долларов.

За 2 часа почасовая часть составляет $2 \times 100 = 200$ долларов, значит, базовая плата — $250 - 200 = 50$ долларов.

Проверка на 3 часах: $50 + 3 \times 100 = 350$ долларов, как в счёте.

За 5 часов: $50 + 5 \times 100 = 550$ долларов.

Ответ: 100 долларов в час плюс 50 долларов за выезд; 5 часов стоили бы 550 долларов

Задачи

1. Сантехник выставляет счёт на 180 долларов за 2 часа и на 300 долларов за 4 часа. Считая, что есть базовая плата плюс цена за час, найдите обе величины.

Подсказка: Сколько добавляют два лишних часа?

Ответ: Два лишних часа добавляют $300 - 180 = 120$ долларов, то есть 60 долларов в час; базовая плата — $180 - 2 \times 60 = 60$ долларов.

2. Дом описывается площадью в квадратных футах, числом спален, числом ванных и минутами ходьбы до центра. Запишите вектор признаков дома площадью 1200 квадратных футов с 2 спальнями, 1 ванной, в 25 минутах от центра и скажите, что было бы меткой в регрессии по продажам домов.

Подсказка: Сохраните порядок признаков из текста.

Ответ: Вектор признаков — $[1200, 2, 1, 25]$; меткой была бы наблюдавшаяся цена продажи, числовое значение.

Классификация

Хотя регрессионные модели прекрасно отвечают на вопросы «сколько?», многие задачи в этот шаблон не укладываются. Возьмём, например, банк, который хочет добавить в своё мобильное приложение функцию сканирования чеков. В идеале клиент просто фотографирует чек, а приложение автоматически распознаёт текст на изображении. Если предположить, что мы как-то умеем вырезать фрагменты изображения, соответствующие каждому рукописному символу, то главной оставшейся задачей будет определить, какой символ из некоторого известного набора изображён на каждом фрагменте. Задачи такого рода, «который из?», называются классификацией и требуют иного набора инструментов, чем регрессия, хотя многие приёмы переносятся.



ПОТЕРИ: ПЕРЕКРЁСТНАЯ ЭНТРОПИЯ

Который из? Вероятности по классам

В классификации мы хотим, чтобы модель смотрела на признаки, например на значения пикселей изображения, и предсказывала, к какой категории (иногда её называют классом) из некоторого дискретного набора вариантов относится пример. Для рукописных цифр у нас может быть десять классов, соответствующих цифрам от 0 до 9. Простейшая форма классификации — когда классов всего два; такую задачу мы называем бинарной классификацией. Например, наш набор данных может состоять из изображений животных, а метками могут быть классы «кошка» и «собака». Если в регрессии мы искали регрессор, выдающий числовое значение, то в классификации ищем классификатор, на выходе которого — предсказанный класс.

По причинам, к которым мы перейдём, когда книга станет более технической, бывает трудно оптимизировать модель, которая может выдавать только твёрдое отнесение к категории, например либо «кошка», либо «собака». В таких случаях обычно гораздо проще выразить модель на языке вероятностей. По признакам примера модель приписывает вероятность каждому возможному классу. Вернёмся к примеру с классификацией животных, где классы — кошка и собака: классификатор может посмотреть на изображение и выдать вероятность того, что на нём кошка, равную 0,9. Это число можно истолковать так: классификатор на 90% уверен, что на изображении кошка. Величина вероятности предсказанного класса передаёт представление о неопределённости. Это не единственное такое представление, и другие мы обсудим в главах о более сложных темах.

Когда возможных классов больше двух, мы называем задачу многоклассовой классификацией. Типичный пример — распознавание рукописных символов с классами 0, 1, 2 и так далее до 9, затем a, b, c и так далее. Если задачи регрессии мы решали, стараясь минимизировать квадрат ошибки, то обычная функция потерь для задач классификации называется перекрёстной энтропией; её название станет понятным, когда в следующих главах мы познакомимся с теорией информации.

Заметьте, что самый вероятный класс — не обязательно тот, на который вы будете опираться при принятии решения. Предположим, вы нашли у себя на заднем дворе красивый гриб, как на рисунке.



ПОГАНКА 0,2
БЕЗОПАСЕН 0,8

СЪЕСТЬ

$$0.2 \times \infty + 0.8 \times 0 = \infty$$

ВЫБРОСИТЬ ✓

$$0.2 \times 0 + 0.8 \times 1 = 0.8$$

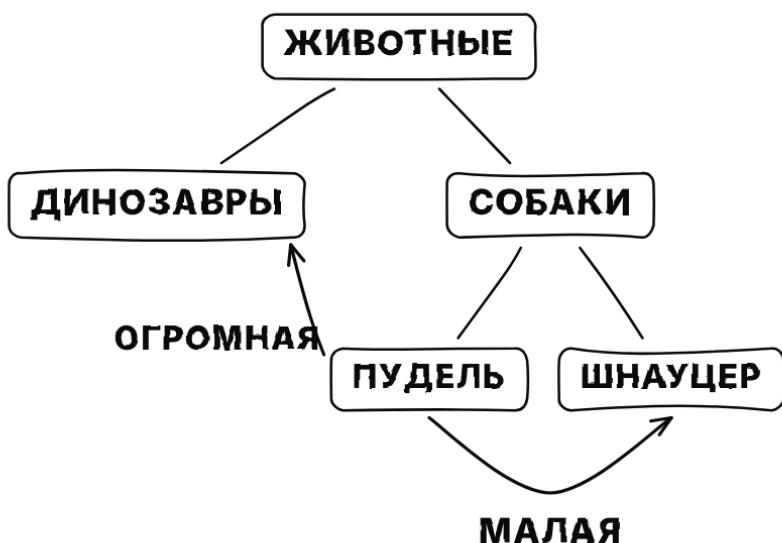
Ожидаемые потери от гриба

Теперь предположим, что вы построили классификатор и обучили его предсказывать по фотографии, ядовит ли гриб. Скажем, наш классификатор ядовитости выдаёт, что вероятность того, что на рисунке бледная поганка, равна 0,2. Иными словами, классификатор на 80% уверен, что наш гриб не бледная поганка. И всё же есть его мог бы только безумец. Потому что верная польза вкусного ужина не стоит 20-процентного риска от него умереть. Иными словами, последствия неопределённого риска намного перевешивают пользу. Поэтому, чтобы решить, есть ли гриб, нужно вычислить ожидаемый ущерб от каждого действия, который зависит и от вероятных исходов, и от пользы или вреда, связанных с каждым из них. В этом случае ущерб от того, чтобы съесть гриб, может составить $0,2 \times \infty + 0,8 \times 0 = \infty$, тогда как

потери от того, чтобы его выбросить, — $0,2 \times 0 + 0,8 \times 1 = 0,8$.

Наша осторожность оправдалась: как сказал бы любой миколог, гриб на рисунке — действительно бледная поганка.

Классификация может быть гораздо сложнее, чем просто бинарная или многоклассовая. Например, есть разновидности классификации, имеющие дело с иерархически устроенными классами. В таких случаях не все ошибки равноценны: если уж ошибаться, мы предпочли бы спутать объект с родственным классом, а не с далёким. Обычно это называют иерархической классификацией. Для вдохновения можно вспомнить Линнея, который выстроил животный мир в иерархию.



Близкие и далёкие ошибки

В классификации животных, может быть, не так страшно принять пуделя за шнауцера, но наша модель заплатила бы огромный штраф, если бы спутала пуделя с динозавром. Какая иерархия важна, может зависеть от того, как вы собираетесь использовать модель. Например, гремучие змеи и подвязочные змеи могут быть близки на филогенетическом древе, но принять гремучую змею за подвязочную может обернуться смертью.

Вопросы

1. Чтение рукописных символов на чеке — это классификация, а не регрессия. Почему?
 - a. Потому что чек — это изображение
 - b. Потому что сумма на чеке — число
 - c. Потому что каждый фрагмент нужно отнести к одному символу из известного набора
 - d. Потому что у банка много клиентов
2. Почему классификаторы обычно выдают вероятность для каждого класса, а не твёрдый ответ?
 - a. Модель, выдающую только твёрдые ответы, бывает трудно оптимизировать
 - b. Вероятности занимают меньше памяти, чем названия классов
 - c. Твёрдые ответы никогда не бывают верными
 - d. Вероятности убирают всякую неопределённость

3. Классификатор говорит, что гриб — бледная поганка с вероятностью 0,2. Почему его всё равно не стоит есть?
- a. Потому что классификатор ошибается в 80% случаев
 - b. Потому что 0,2 — самый вероятный класс
 - c. Потому что классификаторы не умеют судить о грибах
 - d. Потому что ожидаемый вред от того, чтобы съесть гриб, перевешивает пользу ужина
4. Какая ошибка в иерархической классификации обходится дороже для пуделя?
- a. Назвать его шнауцером
 - b. Назвать его динозавром
 - c. Обе стоят одинаково
 - d. Ни одна, потому что считаются только бинарные ошибки

Ответы

1. **с)** Оставшаяся задача — определить, какой символ из известного набора изображён на каждом фрагменте: это задача «который из?», называемая классификацией.
2. **а)** Модель, которая может выдавать только твёрдое отнесение к категории, бывает трудно оптимизировать, поэтому модель проще выразить на языке вероятностей.
3. **d)** Верная польза ужина не стоит 20-процентного риска умереть; ожидаемый ущерб от того, чтобы съесть гриб, бесконечен, а выбросить его стоит лишь 0,8.
4. **b)** Не все ошибки равноценны: принять пуделя за шнауцера, может быть, не так страшно, но спутать его с динозавром стоит огромного штрафа.

Разобранные примеры

Ожидаемый ущерб от гриба Классификатор даёт вероятность 0,2, что гриб — бледная поганка. Съесть бледную поганку стоит ∞ (смерть), а съесть безопасный гриб — 0; выбросить бледную поганку стоит 0, а выбросить безопасный гриб — 1 (потерянный ужин). Вычислите ожидаемый ущерб от каждого действия.

Вероятность того, что гриб безопасен, равна $1 - 0,2 = 0,8$.

Съесть: $0,2 \times \infty + 0,8 \times 0 = \infty$, поскольку любой шанс бесконечного вреда делает сумму бесконечной.

Выбросить: $0,2 \times 0 + 0,8 \times 1 = 0,8$.

Меньший ожидаемый ущерб — у варианта «выбросить», поэтому гриб выбрасываем.

Ответ: Съесть: ∞ ; выбросить: 0,8. Выбросить.

Задачи

1. Другой классификатор говорит, что гриб — бледная поганка с вероятностью 0,05. При тех же ценах (съесть поганку — ∞ , выбросить безопасный гриб — 1) каков ожидаемый ущерб от того, чтобы съесть гриб, и от того, чтобы его выбросить?

Подсказка: Сначала найдите вероятность того, что гриб безопасен.

Ответ: Гриб безопасен с вероятностью 0,95. Съесть: $0,05 \times \infty + 0,95 \times 0 = \infty$. Выбросить: $0,05 \times 0 + 0,95 \times 1 = 0,95$. Выбросить по-прежнему лучше, потому что любой риск бесконечного вреда перевешивает ужин.

2. Для каждой задачи скажите, бинарная это классификация или многоклассовая и сколько в ней классов: кошка или собака; рукописные цифры; рукописные цифры и строчные буквы от a до z.

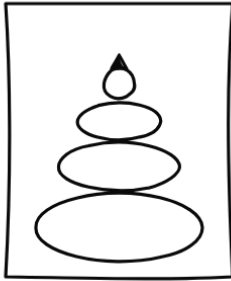
Подсказка: Посчитайте возможные категории.

Ответ: Кошка или собака: бинарная, 2 класса. Цифры: многоклассовая, 10 классов (от 0 до 9). Цифры и строчные буквы: многоклассовая, $10 + 26 = 36$ классов.

Разметка тегами, поиск и рекомендательные системы

Некоторые задачи классификации аккуратно укладываются в бинарную или многоклассовую постановку. Например, мы могли бы обучить обычный бинарный классификатор отличать кошек от собак. При нынешнем уровне компьютерного зрения это легко сделать готовыми инструментами. И всё же, какой бы точной ни стала наша модель, мы можем оказаться в затруднении, когда классификатору попадётся изображение бременских музыкантов, героев популярной немецкой сказки о четырёх животных, как на рисунке.

МНОГО МЕТОК



КОШКА



СОБАКА



ОСЁЛ



ПЕТУХ



ОБЛАКО

AWS

МО

GPU

Много меток сразу

Как видите, на фотографии кошка, петух, собака и осёл, а на заднем плане деревья. Если мы ожидаем встретить такие изображения, многоклассовая классификация может оказаться неправильной постановкой задачи. Вместо этого мы, возможно, захотим дать модели возможность сказать, что на изображении кошка, собака, осёл и петух.

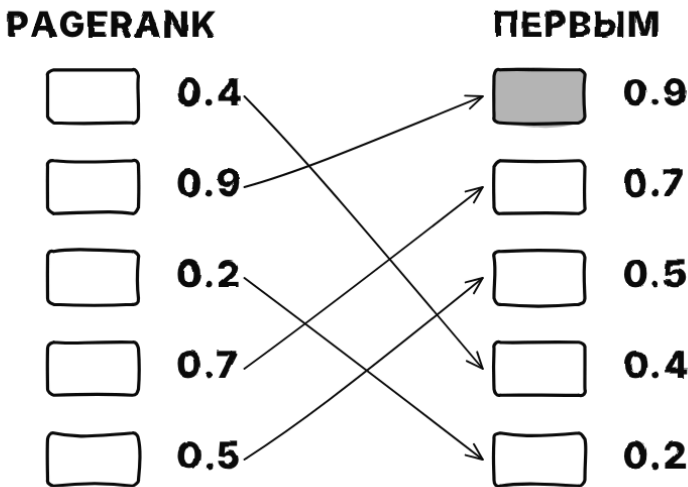
Задача обучения предсказывать классы, которые не исключают друг друга, называется классификацией с несколькими метками. Задачи автоматической разметки тегами обычно лучше всего описываются как классификация с несколькими метками. Вспомните теги, которые люди ставят к записям технического блога, например «машинное обучение», «технологии»,

«гаджеты», «языки программирования», «Linux», «облачные вычисления», «AWS». К типичной статье может быть поставлено от 5 до 10 тегов. Как правило, теги так или иначе связаны друг с другом. Записи об «облачных вычислениях» скорее всего упоминают «AWS», а записи о «машинном обучении» — «GPU».

Иногда такие задачи разметки опираются на огромные наборы меток. В Национальной медицинской библиотеке работает много профессиональных разметчиков, которые сопоставляют каждой статье, индексируемой в PubMed, набор тегов из онтологии Medical Subject Headings (MeSH) — собрания примерно из 28 000 тегов. Правильная разметка статей важна, потому что позволяет исследователям проводить исчерпывающие обзоры литературы. Это долгий процесс, и обычно между помещением статьи в архив и её разметкой проходит год. Машинное обучение может давать предварительные теги, пока каждая статья не пройдёт полноценную ручную проверку. И действительно, организация BioASQ уже несколько лет проводит соревнования по этой задаче.

В области информационного поиска мы часто упорядочиваем наборы объектов по рангу. Возьмём, например, веб-поиск. Цель здесь не столько определить, относится ли конкретная страница к запросу, сколько решить, какой из набора подходящих результатов показать конкретному пользователю на самом видном месте. Один из способов — сначала присвоить оценку каждому элементу набора, а затем выдать элементы с наивысшими оценками. PageRank, изначальный секретный ингредиент поисковой системы Google, был ранним примером такой системы оценок. Как ни странно, оценка PageRank не зависела от самого запроса. Вместо этого простой фильтр

релевантности определял набор подходящих кандидатов, а затем PageRank выдвигал вперёд более авторитетные страницы. Сегодня поисковые системы используют машинное обучение и поведенческие модели, чтобы получать оценки релевантности, зависящие от запроса. Этой теме посвящены целые научные конференции.



Оценить, затем ранжировать

Рекомендательные системы — ещё одна постановка задачи, родственная поиску и ранжированию. Задачи похожи в том, что цель — показать набор объектов, подходящих пользователю. Главное отличие — в рекомендательных системах упор делается на персонализацию под конкретных пользователей. Например, при рекомендации фильмов страница результатов для любителя

научной фантастики и страница для ценителя комедий с Питером Селлерсом могут сильно различаться. Похожие задачи возникают и в других рекомендациях, например товаров в магазинах, музыки и новостей.



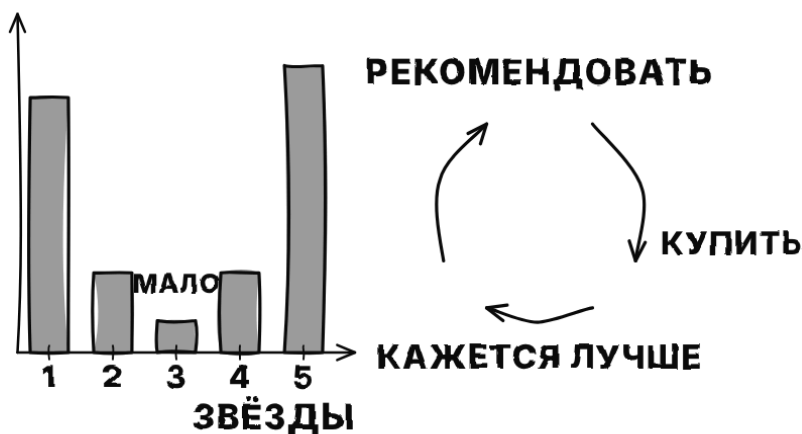
Один магазин, разные пользователи

В некоторых случаях покупатели дают явную обратную связь, сообщая, насколько им понравился тот или иной товар (например, оценки и отзывы на Amazon, IMDb или Goodreads). В других случаях обратная связь неявная, например когда пользователь пропускает песни в плейлисте, что может говорить о недовольстве, а может — просто о том, что песня была не к месту. В простейших постановках такие системы обучают оценивать некую величину, например ожидаемое число звёзд в

оценке или вероятность того, что данный пользователь купит определённый товар.

Имея такую модель, для любого пользователя можно найти объекты с наибольшими оценками и порекомендовать их ему. Промышленные системы значительно сложнее и при вычислении таких оценок учитывают подробную активность пользователя и характеристики товаров. На рисунке показаны книги по глубокому обучению, которые Amazon рекомендует на основе алгоритмов персонализации, настроенных под предпочтения Астона.

Несмотря на огромную экономическую ценность, рекомендательные системы, наивно построенные поверх предсказательных моделей, страдают серьёзными концептуальными недостатками. Начнём с того, что мы наблюдаем лишь цензурированную обратную связь: пользователи охотнее оценивают фильмы, которые вызвали у них сильные чувства. Например, по пятибалльной шкале можно заметить, что у товаров много оценок в одну и в пять звёзд, но подозрительно мало оценок в три звезды. Более того, нынешние покупательские привычки часто сами являются результатом действующего алгоритма рекомендаций, но алгоритмы обучения не всегда учитывают эту деталь. Поэтому могут возникать петли обратной связи: рекомендательная система продвигает какой-то товар, его начинают считать лучше (потому что его больше покупают), и в результате его рекомендуют ещё чаще. Многие из этих задач — как справляться с цензурированием, стимулами и петлями обратной связи — важные открытые вопросы исследований.



Цензурированные оценки, петли обратной связи

Вопросы

1. Почему многоклассовая классификация — неправильная постановка для фотографии бременских музыкантов?
 - a. Потому что на заднем плане деревья
 - b. Потому что животные нарисованы, а не сфотографированы
 - c. Потому что на ней сразу несколько классов, а многоклассовая выбирает только один
 - d. Потому что классов слишком мало

2. Что было странного в том, как PageRank оценивал страницы?
 - a. Его оценки не зависели от самого запроса
 - b. Он оценивал страницы только по их длине
 - c. Он ранжировал страницы случайно
 - d. Он вообще не использовал фильтр релевантности
3. Что главным образом отличает рекомендательные системы от поиска и ранжирования?
 - a. Они никогда не используют оценки
 - b. Они работают только для фильмов
 - c. Они показывают неподходящие объекты
 - d. Упор на персонализацию под конкретных пользователей
4. Как в рекомендательной системе может возникнуть петля обратной связи?
 - a. Пользователи совсем перестают ставить оценки
 - b. Товар, который продвигает система, покупают больше, считают лучше и рекомендуют ещё чаще
 - c. Система забывает старые покупки
 - d. Пользователи ставят всем товарам по три звезды

Ответы

1. **с)** На фотографии кошка, петух, собака и осёл, поэтому мы хотим, чтобы модель могла назвать их всех: классы не исключают друг друга.
2. **а)** Оценка PageRank не зависела от запроса; простой фильтр релевантности отбирал кандидатов, а затем PageRank выдвигал вперёд авторитетные.

3. d) И те и другие показывают подходящие объекты; главное отличие — упор на персонализацию, поэтому любитель фантастики и поклонник Питера Селлера видят разные страницы.

4. b) Покупки отчасти вызваны текущими рекомендациями; товар, который активно продвигают, больше покупают, его считают лучше, и в результате рекомендуют ещё чаще.

Задачи

1. Своими словами объясните разницу между многоклассовой классификацией и классификацией с несколькими метками на примере тегов блога.

Подсказка: Спросите себя, к скольким классам одновременно может относиться один пример.

Ответ: В многоклассовой классификации каждый пример относится ровно к одному классу из нескольких. В классификации с несколькими метками классы не исключают друг друга, так что у одного примера их может быть много; у записи в блоге может быть сразу от 5 до 10 тегов, например «облачные вычисления» и «AWS» вместе.

2. Объясните, что значит «цензурированная обратная связь» для системы рекомендаций фильмов и какое распределение оценок звёздами она даёт.

Подсказка: Кто берёт на себя труд оценить фильм?

Ответ: Пользователи в основном оценивают фильмы, которые вызвали у них сильные чувства, поэтому система видит лишь часть мнений. По пятибалльной шкале получается много оценок в одну и пять звёзд и подозрительно мало в три, так что оценки не дают честной картины того, что люди чувствуют.

3. Приведите пример явной и неявной обратной связи и объясните, почему неявная обратная связь может быть двусмысленной.

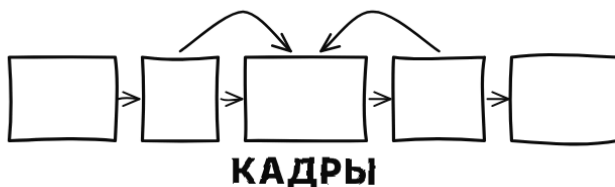
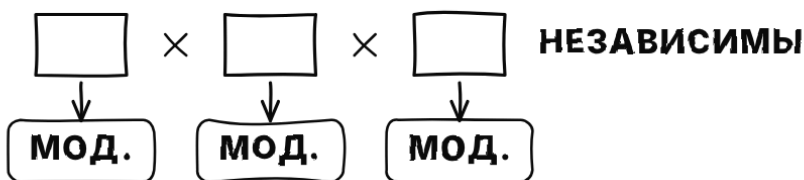
Подсказка: Подумайте об оценках и о пропуске песни.

Ответ: Явная обратная связь — это оценка звёздами или отзыв, которые прямо говорят, насколько человеку понравился товар.

Неявная — это поведение, например пропуск песни в плейлисте; оно может означать недовольство, а может — лишь то, что песня была не к месту.

Обучение на последовательностях

До сих пор мы рассматривали задачи, в которых есть фиксированное число входов и получается фиксированное число выходов. Например, мы рассматривали предсказание цен домов по фиксированному набору признаков: площади, числу спален, числу ванных и времени в пути до центра. Мы обсуждали также переход от изображения (фиксированной размерности) к предсказанным вероятностям того, что оно относится к каждому из фиксированного числа классов, и предсказание оценок звёздами при покупках только по ID пользователя и ID товара. В этих случаях, когда модель обучена, каждый тестовый пример после подачи в модель тут же забывается. Мы предполагали, что последовательные наблюдения независимы, поэтому держать этот контекст не нужно.



ВХОД И ВЫХОД — ПОСЛЕДОВАТЕЛЬНОСТИ

Независимые входы и последовательности

Но как быть с фрагментами видео? Здесь каждый фрагмент может состоять из разного числа кадров. И наша догадка о том, что происходит в каждом кадре, может стать гораздо увереннее, если учесть предыдущие или последующие кадры. То же самое с языком. Например, одна из популярных задач глубокого обучения — машинный перевод: принять предложения на исходном языке и предсказать их перевод на другой язык.

Такие задачи встречаются и в медицине. Мы можем захотеть, чтобы модель следила за пациентами в реанимации и поднимала тревогу всякий раз, когда их риск умереть в ближайшие 24 часа превышает некоторый порог. Здесь мы не стали бы каждый час выбрасывать всё, что знаем об истории пациента, потому что

вряд ли захотим делать предсказания только по последним измерениям.

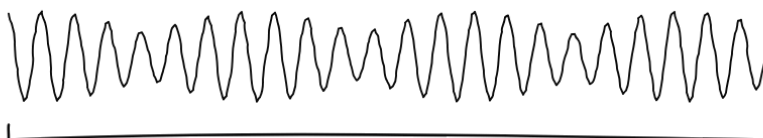
Подобные вопросы — среди самых захватывающих применений машинного обучения, и это примеры обучения на последовательностях. Они требуют от модели либо принимать последовательности входов, либо выдавать последовательности выходов (либо и то и другое). В частности, обучение «последовательность в последовательность» рассматривает задачи, в которых и входы, и выходы — последовательности переменной длины. Примеры — машинный перевод и преобразование речи в текст. Хотя рассмотреть все виды преобразований последовательностей невозможно, стоит упомянуть следующие частные случаи.

Разметка и синтаксический разбор. Здесь текстовая последовательность размечается атрибутами. Входы и выходы выровнены, то есть их одинаковое число и они идут в соответствующем порядке. Например, при разметке частей речи мы помечаем каждое слово предложения соответствующей частью речи, то есть «существительное» или «прямое дополнение». Или мы можем захотеть узнать, какие группы соседних слов обозначают именованные сущности, например людей, места или организации. В нарочито простом примере ниже мы, возможно, хотим лишь указать, входит ли каждое слово предложения в именованную сущность (помечено как «Ent»). В предложении «Tom has dinner in Washington with Sally» слова Tom, Washington и Sally помечены «Ent», а все остальные получают прочерк.

ВЫРОВНЕННЫ

TOM HAS DINNER IN
ENT — — —

WASHINGTON WITH SALLY
ENT — ENT



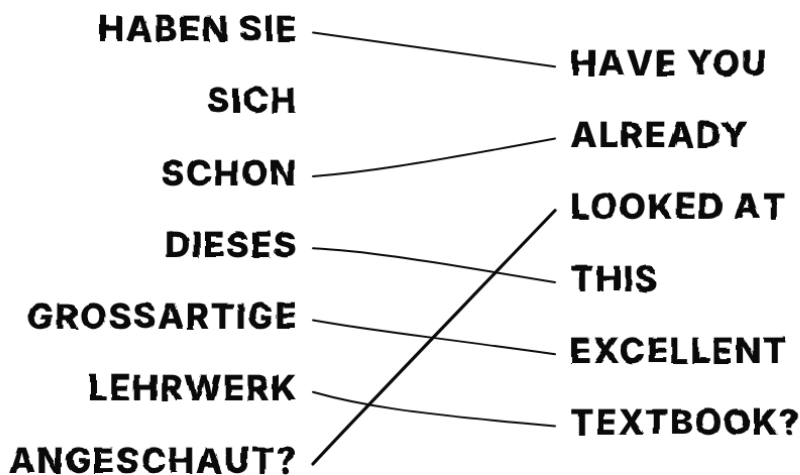
HELLO THERE EVERYONE

МНОГО ОТСЧЁТОВ → МАЛО СЛОВ

Выровненные метки, невыровненный звук

Автоматическое распознавание речи. При распознавании речи входная последовательность — это аудиозапись говорящего, как на рисунке, а выход — запись того, что он сказал. Трудность в том, что аудиокадров гораздо больше, чем текста (звук обычно оцифровывают с частотой 8 кГц или 16 кГц), то есть взаимно однозначного соответствия между звуком и текстом нет: одному произнесённому слову могут соответствовать тысячи отсчётов. Это задачи «последовательность в последовательность», в которых выход гораздо короче входа. Люди замечательно распознают речь даже в некачественной записи, а заставить компьютер сделать то же самое — серьёзнейшая задача.

Синтез речи. Это задача, обратная автоматическому распознаванию речи. Здесь на входе текст, а на выходе — аудиофайл. В этом случае выход гораздо длиннее входа.



Перевод переставляет слова

Машинный перевод. В отличие от распознавания речи, где соответствующие входы и выходы идут в одном и том же порядке, в машинном переводе новую трудность создают невыровненные данные. Здесь входная и выходная последовательности могут быть разной длины, а соответствующие участки последовательностей могут идти в разном порядке. Рассмотрим наглядный пример того, как немцы любят ставить глаголы в конец предложения. По-немецки: «Haben Sie sich schon dieses grossartige Lehrwerk angeschaut?» По-

английски: «Have you already looked at this excellent textbook?» С неправильным выравниванием: «Have you yourself already this excellent textbook looked at?»

Многие родственные задачи возникают и в других задачах обучения. Например, определить, в каком порядке пользователь читает веб-страницу, — это задача анализа двумерной вёрстки. В задачах диалога возникают всевозможные дополнительные сложности: чтобы решить, что сказать дальше, нужно учитывать знания о реальном мире и предшествующее состояние разговора на больших промежутках времени. Такие темы — область активных исследований.

Вопросы

1. Почему модели, следящей за пациентами в реанимации, не следует каждый час забывать историю пациента?
 - a. Потому что больница требует хранить записи
 - b. Потому что предсказания только по последним измерениям могут упустить важное
 - c. Потому что каждое измерение не зависит от остальных
 - d. Потому что у модели фиксированное число входов
2. Почему при разметке частей речи входы и выходы называют выровненными?
 - a. Их одинаковое число, и они идут в соответствующем порядке
 - b. И те и другие написаны по-английски

- c. Выход гораздо короче входа
 - d. Они могут идти в разном порядке
3. Что делает распознавание речи задачей «последовательность в последовательность» без взаимно однозначного соответствия?
- a. Запись текста длиннее аудиозаписи
 - b. Речь всегда записывают в плохом качестве
 - c. Одному произнесённому слову могут соответствовать тысячи аудиоотсчётов
 - d. Каждый аудиокадр соответствует ровно одной букве
4. Какую новую трудность создаёт машинный перевод по сравнению с распознаванием речи?
- a. На входе звук, а не текст
 - b. Выход всегда короче входа
 - c. У последовательностей фиксированная длина
 - d. Соответствующие части могут идти в разном порядке

Ответы

1. **b)** Мы не стали бы выбрасывать всё, что знаем об истории пациента, потому что вряд ли захотим делать предсказания только по последним измерениям.
2. **a)** При разметке и разборе входы и выходы выровнены: по одной метке на слово, в том же порядке, что и слова.
3. **c)** Аудиокадров гораздо больше, чем текста, поскольку одному слову могут соответствовать тысячи отсчётов, так что выход гораздо короче входа.

4. d) В машинном переводе данные не выровнены: последовательности могут быть разной длины, а соответствующие участки — идти в разном порядке, как немецкие глаголы в конце.

Разобранные примеры

Сколько отсчётов в одном слове Звук оцифровывается с частотой 16 кГц, то есть 16 000 отсчётов в секунду. Говорящий произносит одно слово за полсекунды. Сколько аудиоотсчётов соответствует этому одному слову?

16 кГц означает 16 000 отсчётов каждую секунду.

За полсекунды получается $16\,000 \times 0,5 = 8000$ отсчётов.

Все 8000 отсчётов соответствуют одному слову в записи текста, поэтому взаимно однозначного соответствия между звуком и текстом нет.

Ответ: 8000 отсчётов на одно слово

Задачи

1. Для каждой задачи скажите, выход примерно такой же длины, как вход, гораздо короче или гораздо длиннее: разметка частей речи, распознавание речи, синтез речи.

Подсказка: Сравните число элементов на входе и на выходе.

Ответ: Разметка частей речи: такой же длины, по одной метке на слово. Распознавание речи: гораздо короче, тысячи отсчётов превращаются в несколько слов. Синтез речи: гораздо длиннее, несколько слов превращаются в тысячи аудиоотсчётов.

2. Разметьте предложение «Sally visits Paris with Tom» по простой схеме именованных сущностей из текста: «Ent» для слов, входящих в именованную сущность, и прочерк для остальных.

Подсказка: Люди и места — именованные сущности.

Ответ: Sally: Ent, visits: прочерк, Paris: Ent, with: прочерк, Tom: Ent.

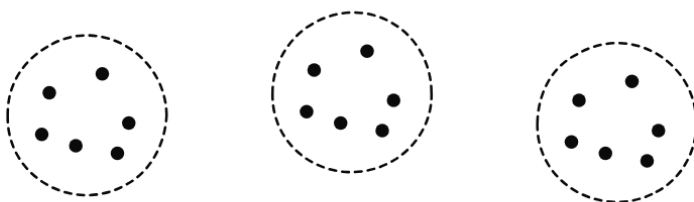
Обучение без учителя и самообучение

Предыдущие примеры были посвящены обучению с учителем, где мы подаём модели огромный набор данных, содержащий и признаки, и соответствующие значения меток. Можно представить, что у ученика при обучении с учителем крайне узкая работа и крайне деспотичный начальник. Начальник стоит у ученика за спиной и в каждой ситуации говорит ему, что именно делать, пока тот не научится переходить от ситуаций к действиям. Работать у такого начальника — довольно унылое занятие. С другой стороны, угодить такому начальнику довольно легко. Нужно лишь как можно быстрее уловить закономерность и подражать действиям начальника.

В обратной ситуации может быть неприятно работать у начальника, который понятия не имеет, чего он от вас хочет. Но если вы собираетесь стать специалистом по данным, лучше к этому привыкнуть. Начальник может просто вручить вам огромную свалку данных и велеть заняться с ними наукой о данных! Звучит расплывчато, потому что это и есть расплывчато. Этот класс задач мы называем обучением без учителя, и вид и число вопросов, которые мы можем задать, ограничены лишь нашей фантазией. Методы обучения без учителя мы рассмотрим в следующих главах. А пока, чтобы разжечь ваш интерес, опишем несколько вопросов, которые можно задать.

Можно ли найти небольшое число прототипов, которые точно обобщают данные? Можно ли разбить набор фотографий на

пейзажи, снимки собак, младенцев, кошек и горных вершин? Точно так же можно ли по набору данных о том, как пользователи просматривают сайты, разбить их на группы со схожим поведением? Эта задача обычно называется кластеризацией.



КЛАСТЕРЫ

РИМ – ИТАЛИЯ + ФРАНЦИЯ = ПАРИЖ



Кластеры и арифметика слов

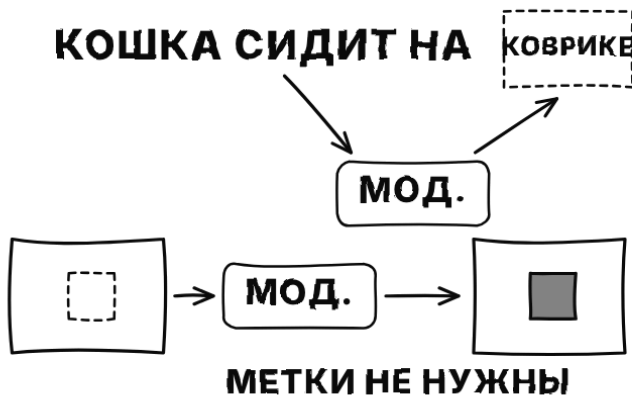
Можно ли найти небольшое число параметров, которые точно передают существенные свойства данных? Траектории мяча хорошо описываются его скоростью, диаметром и массой. Портные выработали небольшое число параметров, довольно точно описывающих форму человеческого тела для подгонки одежды. Такие задачи называют оценкой подпространства. Если

зависимость линейная, это называется методом главных компонент.

Существует ли такое представление объектов (произвольной структуры) в евклидовом пространстве, при котором хорошо согласуются их символические свойства? Это можно использовать для описания сущностей и их отношений, например «Рим» – «Италия» + «Франция» = «Париж».

Существует ли описание первопричин большей части наблюдаемых нами данных? Например, если у нас есть демографические данные о ценах на жильё, загрязнении, преступности, местоположении, образовании и зарплатах, можно ли выяснить, как они связаны, опираясь только на эмпирические данные? Такими вопросами занимаются области, связанные с причинностью и вероятностными графическими моделями.

Ещё одно важное и увлекательное недавнее достижение в обучении без учителя — появление глубоких порождающих моделей. Эти модели явно или неявно оценивают плотность распределения данных. Обученную порождающую модель можно использовать либо для оценки примеров по тому, насколько они вероятны, либо для порождения синтетических примеров из выученного распределения. Первые прорывы глубокого обучения в порождающем моделировании пришли с изобретением вариационных автокодировщиков и продолжились развитием порождающих состязательных сетей. Среди более поздних достижений — нормализующие потоки и диффузионные модели.



Учиться на самих данных

Ещё одним шагом в обучении без учителя стал подъём самообучения — методов, которые используют какую-то сторону неразмеченных данных как источник учителя. Для текста можно обучать модели «заполнять пропуски», предсказывая случайно скрытые слова по окружающим их словам (контексту) в больших корпусах, без всякой разметки! Для изображений можно обучать модели определять взаимное расположение двух вырезанных участков одного изображения, предсказывать закрытую часть изображения по остальным частям или предсказывать, являются ли два примера искажёнными версиями одного и того же исходного изображения. Самообучающиеся модели часто выучивают представления, которые затем используются при

дообучении полученных моделей на какой-то интересующей нас прикладной задаче.

Вопросы

1. Чем, если судить по сравнению с начальником, обучение без учителя отличается от обучения с учителем?
 - a. Начальник проверяет каждый ответ ученика
 - b. Ученик не получает меток, только свалку данных и расплывчатую просьбу
 - c. Ученику в каждой ситуации говорят, что именно делать
 - d. Ученик подражает действиям начальника
2. Разбиение пользователей на группы по схожему поведению на сайтах, без всяких меток, — пример чего?
 - a. Кластеризации
 - b. Метода главных компонент
 - c. Классификации
 - d. Дообучения
3. Для чего, согласно тексту, можно использовать обученную порождающую модель?
 - a. Только для кластеризации фотографий
 - b. Только для предсказания меток
 - c. Для оценки примеров по их вероятности или для порождения синтетических примеров
 - d. Для выяснения первопричин цен на жильё

4. Как самообучение получает учителя без усилий по разметке?
- a. Оно использует деспотичного начальника
 - b. Оно просит людей-разметчиков поставить несколько меток
 - c. Оно работает только на полностью размеченных данных
 - d. Оно использует какую-то сторону самих неразмеченных данных, например скрытые слова

Ответы

1. **b)** Начальник при обучении без учителя просто вручает огромную свалку данных и велит заняться с ними наукой о данных; меток, говорящих, что делать, нет.
2. **a)** Поиск небольшого числа прототипов, например групп пользователей со схожим поведением, называется кластеризацией.
3. **c)** Обученную порождающую модель можно использовать для оценки примеров по тому, насколько они вероятны, или для порождения синтетических примеров из выученного распределения.
4. **d)** Самообучение использует неразмеченные данные как источник учителя, например предсказывая случайно скрытые слова по их контексту.

Задачи

1. Своими словами объясните, что пример «Рим – Италия + Франция = Париж» показывает о представлении слов в евклидовом пространстве.

Подсказка: Подумайте, кем Рим приходится Италии.

Ответ: Если расположить слова точками в пространстве так, чтобы отношения совпадали, то, вычтя Италию из Рима, получим что-то вроде «столица чего-то», а прибавив Францию — столицу Франции, Париж. Такое представление передаёт символические отношения между сущностями.

2. Опишите две задачи самообучения для изображений из текста и скажите, откуда берётся учитель.

Подсказка: Ответ составляет само изображение.

Ответ: Одна задача — определить взаимное расположение двух вырезанных участков одного изображения; другая — предсказать закрытую часть по остальному изображению (или сказать, являются ли два примера искажёнными версиями одного изображения).

Ответы берутся из самих изображений, поэтому разметка людьми не нужна.

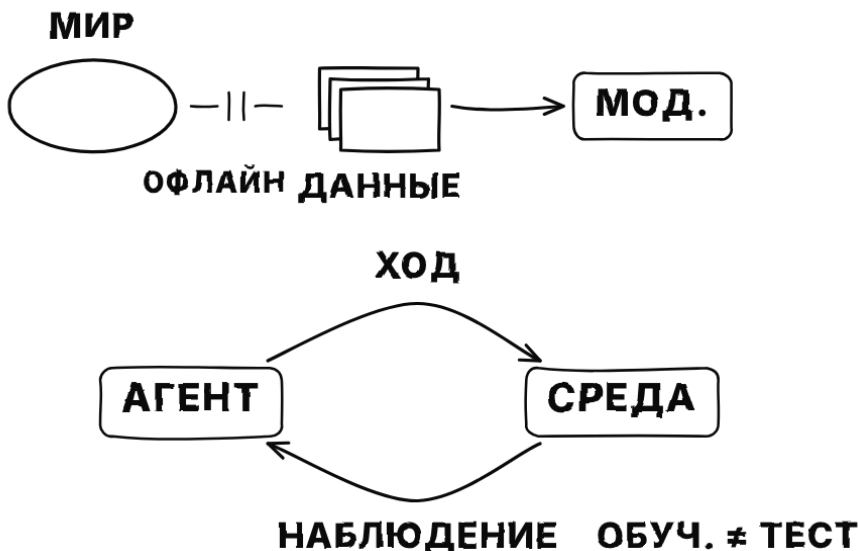
3. Объясните идею оценки подпространства на примере портного.

Подсказка: Сколько чисел нужно портному?

Ответ: У тела бесчисленное множество подробностей, но портные пользуются небольшим числом параметров — несколькими мерками, — которые описывают фигуру достаточно хорошо, чтобы подогнать одежду. Оценка подпространства ищет такой небольшой набор параметров, который передаёт существенные свойства данных.

Взаимодействие со средой и обучение с подкреплением

До сих пор мы не обсуждали, откуда на самом деле берутся данные и что на самом деле происходит, когда модель машинного обучения выдаёт результат. Дело в том, что обучение с учителем и обучение без учителя решают эти вопросы не слишком изощрённо. В обоих случаях мы заранее берём большую кучу данных, а затем запускаем наши машины распознавания закономерностей, больше ни разу не взаимодействуя со средой. Поскольку всё обучение происходит после того, как алгоритм отключён от среды, его иногда называют офлайн-обучением. Например, обучение с учителем предполагает простую схему взаимодействия, изображённую на рисунке.



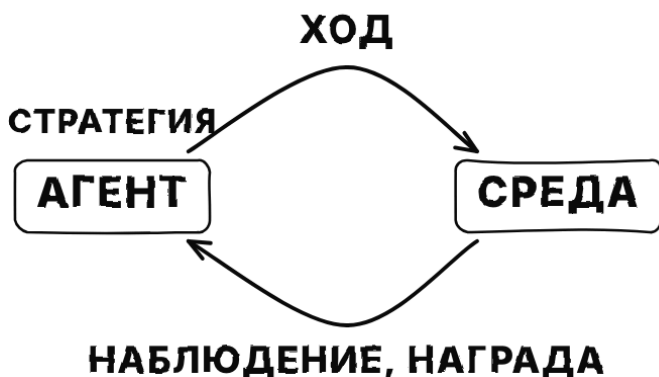
Офлайн-обучение и действие

У этой простоты офлайн-обучения есть свои прелести. Плюс в том, что можно заниматься распознаванием закономерностей обособленно, не заботясь о сложностях, возникающих при взаимодействии с изменчивой средой. Но такая постановка задачи ограничивает. Если вы выросли на романах Азимова о роботах, то, вероятно, представляете себе искусственно разумных агентов, способных не только делать предсказания, но и совершать действия в мире. Мы хотим думать о разумных агентах, а не просто о предсказательных моделях. Это значит, что нужно думать о выборе действий, а не только о предсказаниях. В отличие от простых предсказаний, действия на самом деле влияют на среду. Если мы хотим обучить разумного агента, нужно учитывать, как его действия могут повлиять на его

будущие наблюдения, поэтому офлайн-обучение здесь не годится.

Учёт взаимодействия со средой открывает целый ряд новых вопросов моделирования. Вот лишь несколько примеров. Помнит ли среда, что мы делали раньше? Хочет ли среда нам помочь, как, например, пользователь, диктующий текст распознавателю речи? Хочет ли среда нас обыграть, как, например, спамеры, подстраивающие свои письма, чтобы обойти спам-фильтры? Меняется ли динамика среды? Например, всегда ли будущие данные будут похожи на прошлые или закономерности будут со временем меняться — либо сами собой, либо в ответ на наши автоматические инструменты?

Эти вопросы поднимают проблему сдвига распределения, когда обучающие и тестовые данные различаются. Пример этого, с которым многие из нас сталкивались, — экзамены, составленные лектором, тогда как домашние задания придумывали его ассистенты. Далее мы кратко опишем обучение с подкреплением — богатую схему постановки задач обучения, в которых агент взаимодействует со средой.



СТРАТЕГИЯ: НАБЛЮДЕНИЕ → ДЕЙСТВИЕ

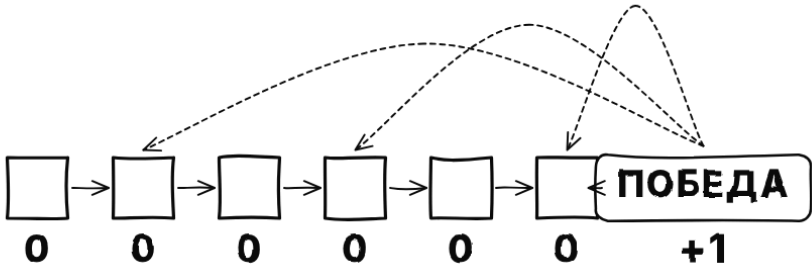
Цикл обучения с подкреплением

Если вы хотите с помощью машинного обучения создать агента, который взаимодействует со средой и совершает действия, то, скорее всего, сосредоточитесь на обучении с подкреплением. Сюда могут относиться применения в робототехнике, в диалоговых системах и даже в создании искусственного интеллекта (ИИ) для видеоигр. Глубокое обучение с подкреплением, применяющее глубокое обучение к задачам обучения с подкреплением, стремительно набрало популярность. Прорывная глубокая Q-сеть, обыгравшая людей в играх Atari по одному лишь изображению, и программа AlphaGo, свергнувшая чемпиона мира по настольной игре го, — два ярких примера.

Обучение с подкреплением даёт очень общую постановку задачи, в которой агент взаимодействует со средой на протяжении ряда шагов времени. На каждом шаге агент получает от среды некоторое наблюдение и должен выбрать действие, которое затем передаётся обратно в среду через некий механизм (иногда его называют исполнительным устройством), и после каждого витка агент получает от среды награду. Этот процесс показан на рисунке. Затем агент получает следующее наблюдение, выбирает следующее действие и так далее. Поведение агента обучения с подкреплением определяется стратегией. Вкратце, стратегия — это просто функция, отображающая наблюдения среды в действия. Цель обучения с подкреплением — получить хорошие стратегии.

Трудно переоценить общность схемы обучения с подкреплением. Например, обучение с учителем можно переформулировать как обучение с подкреплением. Допустим, у нас задача классификации. Мы могли бы создать агента обучения с подкреплением с одним действием на каждый класс. Затем мы могли бы создать среду, которая даёт награду, в точности равную функции потерь исходной задачи обучения с учителем.

ЗАСЛУГА?



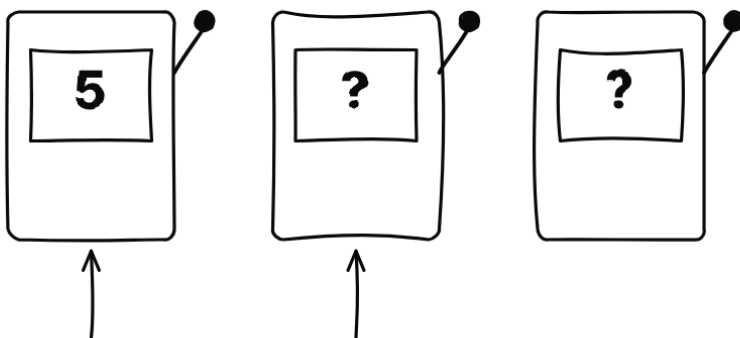
Награда только в конце

Более того, обучение с подкреплением может решать и многие задачи, которые обучению с учителем не под силу. Например, в обучении с учителем мы всегда ожидаем, что обучающий вход снабжён правильной меткой. А в обучении с подкреплением мы не предполагаем, что для каждого наблюдения среда сообщает нам оптимальное действие. Как правило, мы просто получаем какую-то награду. Более того, среда может даже не сообщать, какие действия привели к награде.

Возьмём шахматы. Единственный настоящий сигнал награды приходит в конце партии, когда мы либо выигрываем, получая награду, скажем, 1, либо проигрываем, получая награду, скажем, минус 1. Поэтому обучающимся с подкреплением приходится

решать задачу распределения заслуг: определять, каким действиям поставить исход в заслугу или в вину. То же касается сотрудника, которого повысили 11 октября. Это повышение, скорее всего, отражает ряд удачно выбранных действий за прошедший год. Чтобы получить повышение в будущем, нужно понять, какие действия по пути привели к прошлым повышениям.

Обучающимся с подкреплением может приходиться иметь дело и с проблемой частичной наблюдаемости. То есть текущее наблюдение может не говорить всего о вашем текущем состоянии. Допустим, ваш робот-уборщик оказался заперт в одной из многих одинаковых кладовок вашего дома. Чтобы спасти робота, нужно вычислить его точное местоположение, а для этого, возможно, придётся учесть более ранние наблюдения, сделанные до того, как он попал в кладовку.



ИСПОЛЬЗОВАТЬ ИССЛЕДОВАТЬ

МНОГОРУКИЙ БАНДИТ

Использовать или исследовать

Наконец, в любой момент обучающийся с подкреплением может знать одну хорошую стратегию, но может существовать множество других, лучших стратегий, которых агент никогда не пробовал. Обучающийся с подкреплением должен постоянно выбирать: использовать лучшую (на данный момент) известную стратегию или исследовать пространство стратегий, возможно, жертвуя частью краткосрочной награды ради знаний.

Общая задача обучения с подкреплением имеет очень общую постановку. Действия влияют на последующие наблюдения. Награды наблюдаются только для выбранных действий. Среда может быть наблюдаема полностью или частично. Учесть всю эту сложность сразу — возможно, слишком большое требование. К

тому же не каждая практическая задача обладает всей этой сложностью. Поэтому исследователи изучили ряд частных случаев задач обучения с подкреплением.

Когда среда наблюдаема полностью, задачу обучения с подкреплением называют марковским процессом принятия решений. Когда состояние не зависит от предыдущих действий, её называют задачей контекстного бандита. Когда состояния нет, а есть лишь набор доступных действий с заранее неизвестными наградами, получается классическая задача многорукого бандита.

Вопросы

1. Почему офлайн-обучение не подходит для обучения разумного агента?
 - a. Потому что офлайн-обучению не нужны данные
 - b. Потому что действия агента влияют на его будущие наблюдения
 - c. Потому что агенты не умеют делать предсказания
 - d. Потому что офлайн-обучение слишком медленное
2. Спамеры меняют свои письма, чтобы обойти спам-фильтры. Какой вопрос о среде это иллюстрирует?
 - a. Помнит ли среда, что мы делали?
 - b. Хочет ли среда нам помочь?
 - c. Хочет ли среда нас обыграть?
 - d. Полностью ли наблюдаема среда?

3. Почему в шахматах обучающемуся с подкреплением приходится решать задачу распределения заслуг?
- a. Единственная настоящая награда приходит в конце, поэтому нужно понять, какие ходы заслуживают заслуги или вины
 - b. Доска видна лишь частично
 - c. Каждый ход получает свою награду
 - d. Правила меняются по ходу партии
4. Как называется задача, когда состояния нет, а есть лишь набор действий с заранее неизвестными наградами?
- a. Марковский процесс принятия решений
 - b. Задача контекстного бандита
 - c. Офлайн-обучение
 - d. Задача многорукого бандита

Ответы

1. **b)** Действия на самом деле влияют на среду, поэтому обученный агент должен учитывать, как его действия скажутся на его будущих наблюдениях, а офлайн-обучение это игнорирует.
2. **c)** Спамеры, подстраивающие письма, чтобы обойти спам-фильтры, — пример из текста о среде, которая хочет нас обыграть.
3. **a)** Единственная настоящая награда — победа или поражение — приходит в конце партии, поэтому обучающийся должен определить, каким действиям поставить исход в заслугу или в вину.
4. **d)** Когда состояния нет, а есть лишь доступные действия с заранее неизвестными наградами, получается классическая задача многорукого бандита.

Задачи

1. Своими словами объясните, что такое стратегия и какова цель обучения с подкреплением.

Подсказка: Стратегия связывает то, что агент видит, с тем, что он делает.

Ответ: Стратегия — это функция, отображающая наблюдения среды в действия; она определяет поведение агента. Цель обучения с подкреплением — получить хорошие стратегии, приносящие высокие награды со временем.

2. Опишите, как задачу классификации можно переформулировать как обучение с подкреплением.

Подсказка: Представьте каждый класс как действие.

Ответ: Создайте агента с одним действием на каждый класс и среду, которая даёт награду, равную функции потерь исходной задачи обучения с учителем. Тогда выбрать правильный класс — значит выбрать действие, приносящее лучшую награду.

3. Приложение доставки еды всегда рекомендует ресторан, который до сих пор работал лучше всех. Что оно делает в терминах текста и что может упустить?

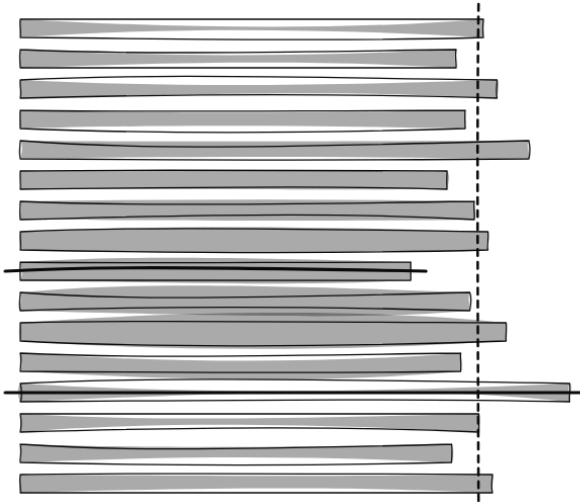
Подсказка: Подумайте об использовании и исследовании.

Ответ: Оно использует лучшую известную на данный момент стратегию. Оно может упустить лучшие рестораны, которых никогда не пробовало; исследование пожертвовало бы частью краткосрочной награды ради знаний о других вариантах.

Истоки

Мы только что рассмотрели небольшую часть задач, которые может решать машинное обучение. Для самых разных задач машинного обучения глубокое обучение даёт мощные инструменты решения. Хотя многие методы глубокого обучения изобретены недавно, основные идеи обучения на данных изучаются уже много веков. На самом деле люди издавна стремились анализировать данные и предсказывать будущие исходы, и именно это стремление лежит в основе значительной части естественных наук и математики. Два примера — распределение Бернулли, названное в честь Якоба Бернулли (1655–1705), и нормальное распределение, открытое Карлом Фридрихом Гауссом (1777–1855). Гаусс, например, изобрёл метод наименьших квадратов, который и сегодня применяется в множестве задач — от страховых расчётов до медицинской диагностики. Такие инструменты усилили экспериментальный подход в естественных науках; например, закон Ома, связывающий ток и напряжение на резисторе, прекрасно описывается линейной моделью.

УСЕЧЁННОЕ СРЕДНЕЕ



Шестнадцать ступней, один фут

Даже в Средние века математики хорошо чувствовали, как делать оценки. Например, в книге по геометрии Якоба Кёбеля (1460–1533) показано, как усреднить длину ступней 16 взрослых мужчин, чтобы оценить типичную длину ступни в населении, как на рисунке.

Когда люди выходили из церкви, 16 взрослых мужчин попросили встать в ряд и измерили их ступни. Затем сумму этих измерений разделили на 16 и получили оценку того, что теперь называется одним футом. Позже этот «алгоритм» улучшили, чтобы справиться с ногами неправильной формы: двух мужчин с самой короткой и самой длинной ступнёй отпускали и усредняли

только по остальным. Это один из самых ранних примеров усечённого среднего.

Статистика по-настоящему развернулась, когда данные стали доступны и их начали собирать. Один из её первопроходцев, Рональд Фишер (1890–1962), внёс большой вклад в её теорию, а также в её применения в генетике. Многие его алгоритмы (например, линейный дискриминантный анализ) и понятия (например, информационная матрица Фишера) до сих пор занимают видное место в основаниях современной статистики. Даже его наборы данных оказали долгое влияние. Набор данных об ирисах, опубликованный Фишером в 1936 году, до сих пор иногда используют, чтобы показать работу алгоритмов машинного обучения. Фишер был также сторонником евгеники, и это должно напоминать нам, что сомнительное с нравственной точки зрения использование науки о данных имеет столь же долгую историю, как и её плодотворное использование в промышленности и естественных науках.

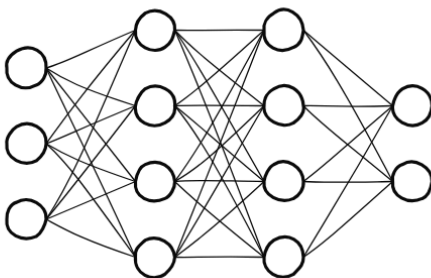
Другие влияния на машинное обучение пришли из теории информации Клода Шеннона (1916–2001) и теории вычислений, предложенной Аланом Тьюрингом (1912–1954). Тьюринг поставил вопрос «может ли машина мыслить?» в своей знаменитой статье «Вычислительные машины и разум». Описывая то, что сегодня называют тестом Тьюринга, он предложил считать машину разумной, если человеку-оценщику трудно отличить ответы машины от ответов человека, опираясь только на текстовое общение.

Ещё одно влияние пришло из нейронауки и психологии. В конце концов, люди явно ведут себя разумно. Многие учёные

задавались вопросом, можно ли объяснить эту способность и, возможно, воспроизвести её устройство. Один из первых алгоритмов, вдохновлённых биологией, сформулировал Дональд Хебб (1904–1985). В своей новаторской книге «Организация поведения» он предположил, что нейроны учатся через положительное подкрепление. Это стало известно как правило обучения Хебба. Эти идеи вдохновили более поздние работы, например алгоритм обучения перцептрона Розенблатта, и заложили основы многих алгоритмов стохастического градиентного спуска, на которых сегодня держится глубокое обучение: подкреплять желательное поведение и ослаблять нежелательное, чтобы получить хорошие значения параметров нейронной сети.



ВМЕСТЕ



ЛИНЕЙНЫЙ НЕЛИНЕЙНЫЙ
ЛИНЕЙНЫЙ

**ЦЕПНОЕ
ПРАВИЛО**



Слои и обратное распространение

Именно биологическое вдохновение дало нейронным сетям их название. Уже больше века (начиная с моделей Александра Бэна, 1873, и Джеймса Шеррингтона, 1890) исследователи пытаются собирать вычислительные схемы, похожие на сети взаимодействующих нейронов. Со временем биологию стали понимать менее буквально, но название закрепилось. В основе лежат несколько ключевых принципов, которые можно найти в большинстве современных сетей. Первый — чередование линейных и нелинейных блоков обработки, которые часто называют слоями. Второй — использование цепного правила (известного также как обратное распространение ошибки) для настройки параметров всей сети сразу.

После начального быстрого прогресса исследования нейронных сетей зачихли примерно с 1995 по 2005 год. В основном по двум причинам. Во-первых, обучение сети требует очень больших вычислений. Оперативной памяти в конце прошлого века было достаточно, а вычислительной мощности не хватало. Во-вторых, наборы данных были сравнительно малы. Собственно, набор данных об ирисах Фишера 1936 года всё ещё был популярным средством проверки эффективности алгоритмов. Набор данных MNIST с его 60 000 рукописных цифр считался огромным.

При нехватке данных и вычислений сильные статистические инструменты, такие как ядерные методы, деревья решений и графические модели, на практике оказывались лучше во многих задачах. К тому же, в отличие от нейронных сетей, их не нужно обучать неделями, и они давали предсказуемые результаты с сильными теоретическими гарантиями.

Вопросы

1. Зачем в улучшенной процедуре измерения ступней отпускали мужчин с самой короткой и самой длинной ступнёй?
 - a. Чтобы ряд стал короче
 - b. Чтобы справиться с ногами неправильной формы, которые исказили бы среднее
 - c. Потому что в ряд помещалось только 14 мужчин
 - d. Чтобы результат получился целым числом
2. Какие два принципа, согласно тексту, лежат в основе большинства современных нейронных сетей?
 - a. Усечённое среднее и метод наименьших квадратов
 - b. Ядерные методы и деревья решений
 - c. Чередование линейных и нелинейных слоёв и обратное распространение для настройки всех параметров сразу
 - d. Тест Тьюринга и обучение по Хеббу
3. Почему исследования нейронных сетей зачихли примерно с 1995 по 2005 год?
 - a. Обучение требовало больших вычислений, а наборы данных были малы
 - b. Не хватало памяти, чтобы хранить сети
 - c. Было доказано, что нейронные сети не работают
 - d. Никто не знал об обратном распространении

4. Почему в то время ядерные методы, деревья решений и графические модели побеждали во многих задачах?
- a. Им нужно было гораздо больше данных, чем нейронным сетям
 - b. Они были вдохновлены биологией
 - c. Только у них были слои
 - d. При малых данных и вычислениях они работали лучше, быстро обучались и имели сильные гарантии

Ответы

1. **b)** Алгоритм улучшили, чтобы справиться с ногами неправильной формы: два крайних значения отбрасывали и усредняли по остальным — раннее усечённое среднее.
2. **c)** Ключевые принципы — чередование линейных и нелинейных блоков обработки, называемых слоями, и цепное правило, то есть обратное распространение, для настройки параметров всей сети сразу.
3. **a)** Две причины: обучение требовало очень больших вычислений при нехватке вычислительной мощности, а наборы данных были сравнительно малы.
4. **d)** При нехватке данных и вычислений они на практике оказывались лучше, не требовали недель обучения и давали предсказуемые результаты с сильными теоретическими гарантиями.

Разобранные примеры

Фут Кёбеля, обычный и усечённый Ступни шестнадцати мужчин имеют длину в сантиметрах: 27, 28, 29, 30, 30, 31, 31, 31, 32, 32, 33, 33, 34, 34, 35, 39. Найдите обычное среднее, а затем

усечённое среднее, отбросив самую короткую и самую длинную ступню.

Сложим все шестнадцать: сумма равна 509 сантиметрам.

Разделим на 16: $509 \div 16 \approx 31,81$ сантиметра.

Отбросим самую короткую (27) и самую длинную (39): сумма остальных четырнадцати равна $509 - 27 - 39 = 443$.

Разделим на 14: $443 \div 14 \approx 31,64$ сантиметра; необычная ступня в 39 больше не тянет оценку вверх.

Ответ: Обычное среднее около 31,81 см; усечённое среднее около 31,64 см

Задачи

1. Пять измерений равны 10, 11, 12, 13 и 34. Вычислите обычное среднее и усечённое среднее без наименьшего и наибольшего.

Что лучше описывает типичное значение?

Подсказка: Для усечённого среднего отбросьте 10 и 34.

Ответ: Обычное среднее: $80 \div 5 = 16$. Усечённое среднее: $(11 + 12 + 13) \div 3 = 12$. Усечённое среднее типичнее, потому что одно странное значение 34 подтянуло обычное среднее вверх.

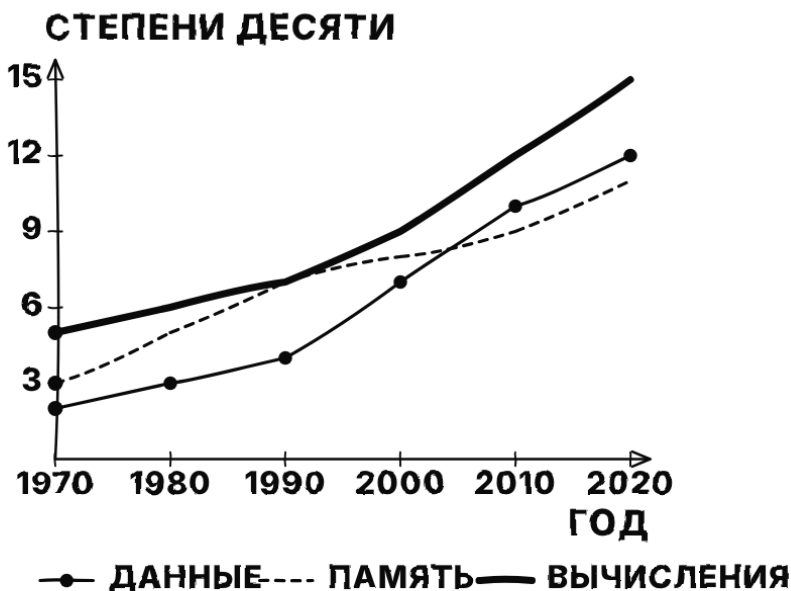
2. Своими словами опишите тест Тьюринга.

Подсказка: Кто должен заметить разницу и по чему?

Ответ: Человек-оценщик обменивается текстовыми сообщениями с машиной и с человеком. Если ему трудно отличить ответы машины от ответов человека, опираясь только на текст, машину можно считать разумной.

Путь к глубокому обучению

Многое изменилось с появлением огромных объёмов данных благодаря Всемирной паутине, компаниям, обслуживающим сотни миллионов пользователей онлайн, распространению дешёвых качественных датчиков, недорогому хранению данных (закон Крайдера) и дешёвым вычислениям (закон Мура). В частности, картину вычислений в глубоком обучении перевернули достижения в области GPU, изначально созданных для компьютерных игр. Внезапно алгоритмы и модели, казавшиеся вычислительно неосуществимыми, оказались в пределах досягаемости. Лучше всего это видно из таблицы, где размер наборов данных сопоставлен с памятью и вычислительной мощностью компьютеров.



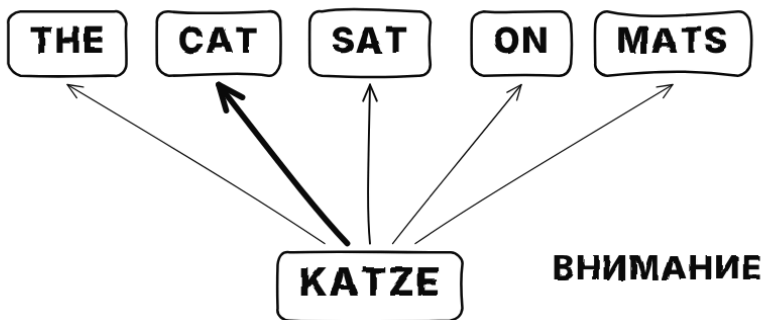
Данные, память и вычисления, 1970–2020

В 1970 году набор данных содержал около 100 примеров (данные об ирисах), память составляла 1 КБ, а компьютер выполнял около 100 тысяч операций с плавающей точкой в секунду (Intel 8080). В 1980 году набор данных содержал около 1 тысячи примеров (цены домов в Бостоне), память составляла 100 КБ, а компьютеры выполняли 1 миллион операций в секунду (Intel 80186). В 1990 году наборы данных содержали 10 тысяч примеров (оптическое распознавание символов), память составляла 10 МБ, а компьютеры выполняли 10 миллионов операций в секунду (Intel 80486). В 2000 году наборы данных содержали 10 миллионов примеров (веб-страницы), память составляла 100 МБ, а компьютеры выполняли 1 миллиард операций в секунду (Intel Core). В 2010 году наборы данных

содержали 10 миллиардов примеров (реклама), память составляла 1 ГБ, а компьютеры выполняли 1 триллион операций в секунду (NVIDIA C2050). В 2020 году наборы данных содержали 1 триллион примеров (социальная сеть), память составляла 100 ГБ, а компьютеры выполняли 1 квадриллион операций в секунду (NVIDIA DGX-2).

Заметьте, что оперативная память не поспевала за ростом данных. В то же время вычислительная мощность росла быстрее, чем наборы данных. Это значит, что статистическим моделям нужно стать экономнее в расходе памяти, и они могут тратить больше тактов процессора на оптимизацию параметров благодаря выросшему бюджету вычислений. Поэтому золотая середина в машинном обучении и статистике сместилась от (обобщённых) линейных моделей и ядерных методов к глубоким нейронным сетям. Это также одна из причин, по которым многие опоры глубокого обучения, такие как многослойные перцептроны, свёрточные нейронные сети, долгая краткосрочная память и Q-обучение, по сути были «открыты заново» за последнее десятилетие, после того как довольно долго пребывали в спячке.

Недавний прогресс в статистических моделях, применениях и алгоритмах иногда сравнивают с кембрийским взрывом — моментом стремительного развития в эволюции видов. И действительно, нынешний передний край — не просто следствие того, что доступные ресурсы применили к алгоритмам многолетней давности. Заметьте, что приведённый ниже список идей лишь слегка касается того, что помогло исследователям добиться огромного прогресса за последнее десятилетие.



ТРАНСФОРМЕР: ТОЛЬКО ВНИМАНИЕ

Внимание как указатель

Новые методы управления ёмкостью модели, такие как дропаут, помогли ослабить переобучение. Здесь во время обучения шум вносится по всей нейронной сети.

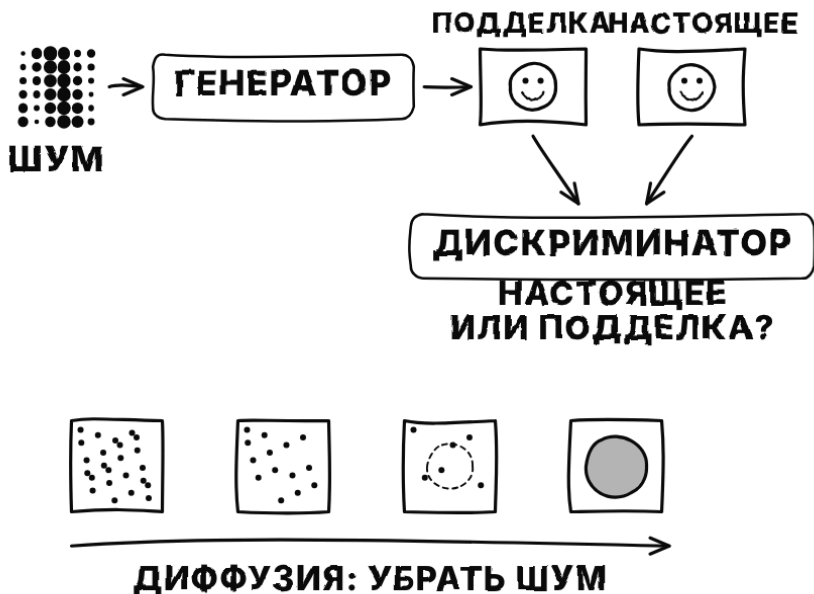
Механизмы внимания решили вторую проблему, которая больше века мучила статистику: как увеличить память и сложность системы, не увеличивая числа обучаемых параметров.

Исследователи нашли изящное решение, используя то, что можно рассматривать только как обучаемую структуру указателей. Вместо того чтобы запоминать целую текстовую последовательность, например для машинного перевода, в представлении фиксированной размерности, достаточно было хранить указатель на промежуточное состояние процесса

перевода. Это позволило значительно повысить точность для длинных последовательностей, поскольку модели больше не нужно было запоминать всю последовательность, прежде чем начать порождать новую.

Архитектура «трансформер», построенная исключительно на механизмах внимания, показала превосходное масштабирование: она работает лучше с ростом размера набора данных, размера модели и объёма вычислений на обучение. Эта архитектура показала убедительный успех в самых разных областях, таких как обработка естественного языка, компьютерное зрение, распознавание речи, обучение с подкреплением и графовые нейронные сети. Например, один трансформер, предобученный на таких разных модальностях, как текст, изображения, моменты в суставах и нажатия кнопок, может играть в Atari, подписывать изображения, вести беседу и управлять роботом.

Моделируя вероятности текстовых последовательностей, языковые модели могут предсказывать текст по другому тексту. Масштабирование данных, модели и вычислений открыло всё больше возможностей языковых моделей выполнять нужные задачи, порождая похожий на человеческий текст по входному тексту. Например, ChatGPT от OpenAI, согласованный с намерениями людей, позволяет пользователям общаться с ним в форме беседы, решая задачи вроде отладки кода и творческого письма.



Генератор против дискриминатора

Многоступенчатые конструкции, например сети с памятью и нейронный программист-интерпретатор, позволили специалистам по статистическому моделированию описывать итеративные подходы к рассуждению. Эти инструменты позволяют многократно менять внутреннее состояние глубокой нейронной сети, выполняя таким образом последовательные шаги цепочки рассуждений, подобно тому как процессор меняет память при вычислении.

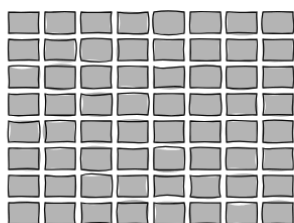
Ключевым шагом в глубоком порождающем моделировании стало изобретение порождающих состязательных сетей. Традиционно статистические методы оценки плотности и порождающие модели были сосредоточены на поиске

подходящих распределений вероятностей и (часто приближённых) алгоритмов выборки из них. В результате эти алгоритмы были во многом ограничены негибкостью, присущей статистическим моделям. Ключевым новшеством порождающих состязательных сетей стала замена алгоритма выборки произвольным алгоритмом с дифференцируемыми параметрами. Затем эти параметры настраиваются так, чтобы дискриминатор (по сути, двухвыборочный тест) не мог отличить поддельные данные от настоящих. Благодаря возможности использовать для порождения данных произвольные алгоритмы оценка плотности открылась для самых разных методов. Примеры скачущих зебр и поддельных лиц знаменитостей — свидетельства этого прогресса. Даже рисующие любители могут получать фотореалистичные изображения всего лишь по наброскам, описывающим расположение сцены.

Кроме того, если процесс диффузии постепенно добавляет к образцам данных случайный шум, то диффузионные модели выучивают процесс удаления шума, чтобы постепенно строить образцы данных из случайного шума, обращая процесс диффузии. Они начали вытеснять порождающие состязательные сети в более новых глубоких порождающих моделях, таких как DALL-E 2 и Imagen, предназначенных для творческого искусства и создания изображений по текстовым описаниям.

Во многих случаях одного GPU недостаточно, чтобы обработать большие объёмы данных, доступные для обучения. За последнее десятилетие способность строить параллельные и распределённые алгоритмы обучения значительно выросла. Одна из главных трудностей при создании масштабируемых алгоритмов в том, что рабочая лошадка оптимизации в глубоком

обучении, стохастический градиентный спуск, опирается на обработку сравнительно небольших мини-пакетов данных. В то же время малые пакеты ограничивают эффективность GPU. Поэтому обучение на 1024 GPU с размером мини-пакета, скажем, 32 изображения на пакет даёт общий мини-пакет около 32 000 изображений. Работы сначала Ли (2017), а затем Ю с коллегами (2017) и Цзя с коллегами (2018) довели этот размер до 64 000 наблюдений, сократив время обучения модели ResNet-50 на наборе данных ImageNet до менее чем 7 минут. Для сравнения, поначалу время обучения измерялось днями.



1024 GPU

$$1,024 \times 32 \approx 32,000$$

ИЗОБРАЖЕНИЙ ЗА ШАГ

ДНИ



■ МИНУТЫ

THEANO → TENSORFLOW → PYTORCH, JAX

Много GPU, один большой пакет

Способность распараллеливать вычисления помогла и прогрессу в обучении с подкреплением. Это привело к значительным

успехам: компьютеры достигли сверхчеловеческого уровня в таких задачах, как го, игры Atari, Starcraft, и в физических симуляциях (например, с помощью MuJoCo), где доступны симуляторы среды. Описание таких достижений в AlphaGo см., например, у Сильвера с коллегами (2016). Короче говоря, обучение с подкреплением работает лучше всего, когда доступно множество кортежей (состояние, действие, награда). Симуляция даёт такую возможность.

Решающую роль в распространении идей сыграли фреймворки глубокого обучения. Первое поколение открытых фреймворков для моделирования нейронных сетей составляли Caffe, Torch и Theano. С помощью этих инструментов было написано много основополагающих статей. Теперь их вытеснили TensorFlow (часто используемый через свой высокоуровневый API Keras), CNTK, Caffe 2 и Apache MXNet. Третье поколение фреймворков составляют так называемые императивные инструменты глубокого обучения; эту тенденцию, по-видимому, начал Chainer, в котором модели описывались синтаксисом, похожим на Python NumPy. Эту идею переняли и PyTorch, и Gluon API от MXNet, и JAX.

Вопросы

1. Поскольку память росла медленнее данных, а вычисления — быстро, что пришлось делать статистическим моделям?
 - а. Использовать меньше вычислений и больше памяти

- b. Стать экономнее в расходе памяти и тратить больше тактов на оптимизацию параметров
 - c. Отказаться от больших наборов данных
 - d. Вернуться к линейным моделям
2. Какую задачу, согласно тексту, решили механизмы внимания?
- a. Как вносить шум во время обучения
 - b. Как обучать сразу на многих GPU
 - c. Как увеличить память и сложность системы, не увеличивая числа обучаемых параметров
 - d. Как делать выборку из распределения вероятностей
3. Как в порождающей состязательной сети настраиваются параметры генератора?
- a. Так, чтобы дискриминатор не мог отличить поддельные данные от настоящих
 - b. Так, чтобы дискриминатор всегда замечал подделки
 - c. Так, чтобы они совпадали с фиксированным распределением вероятностей
 - d. Так, чтобы шум добавлялся шаг за шагом
4. Чем симуляция помогает обучению с подкреплением?
- a. Она избавляет от необходимости в наградах
 - b. Она заменяет стратегию
 - c. Она делает данные независимыми
 - d. Она предоставляет множество кортежей «состояние, действие, награда»

Ответы

1. **b)** Статистическим моделям нужно стать экономнее в расходе памяти, и они могут тратить больше тактов процессора на оптимизацию параметров благодаря выросшему бюджету вычислений.
2. **c)** Внимание решило, как увеличить память и сложность без новых обучаемых параметров, используя обучаемую структуру указателей вместо запоминания всей последовательности.
3. **a)** Алгоритм выборки заменяется алгоритмом с дифференцируемыми параметрами, которые настраиваются так, чтобы дискриминатор не мог отличить поддельные данные от настоящих.
4. **d)** Обучение с подкреплением работает лучше всего, когда доступно множество коротежей «состояние, действие, награда», и симуляция даёт такую возможность.

Разобранные примеры

Общий мини-пакет Обучение идёт на 1024 GPU, каждый обрабатывает мини-пакет из 32 изображений. Сколько изображений составляет общий мини-пакет?

Каждый GPU обрабатывает 32 изображения за шаг.

Одновременно работают 1024 GPU.

$1024 \times 32 = 32\,768$ изображений; в тексте это округлено примерно до 32 000.

Ответ: 32 768 изображений, около 32 000

Задачи

1. По таблице: во сколько раз вырос размер наборов данных с 1970 года (100 примеров) до 2020 года (1 триллион примеров) и

во сколько раз выросла память с 1 КБ до 100 ГБ?

Подсказка: Считайте степени десяти; 1 ГБ — это миллион КБ.

Ответ: Данные выросли с 10^2 до 10^{12} , то есть в 10^{10} раз, в десять миллиардов. Память выросла с 1 КБ до 100 ГБ, то есть до 100 миллионов КБ, в 10^8 раз. Значит, память выросла в сто раз меньше, чем данные, поэтому она и не поспевала за ними.

2. Своими словами объясните, как диффузионная модель создаёт изображение.

Подсказка: Начните с того, что процесс диффузии делает с данными.

Ответ: Процесс диффузии постепенно добавляет к данным случайный шум, пока не останется только шум. Диффузионная модель учится обращать это, шаг за шагом убирая шум, поэтому может начать со случайного шума и постепенно построить новый образец данных, например изображение по текстовому описанию.

Истории успеха

Разделение труда между исследователями систем, создающими более совершенные инструменты, и специалистами по статистическому моделированию, создающими более совершенные нейронные сети, сильно всё упростило. Например, обучение линейной модели логистической регрессии было когда-то нетривиальной домашней задачей, которую в 2014 году не стыдно было дать новым аспирантам по машинному обучению в Университете Карнеги — Меллона. Сегодня эту задачу можно решить меньше чем в 10 строк кода, так что она вполне по силам любому программисту.

У искусственного интеллекта давняя история результатов, которых иначе было бы трудно добиться. Например, системы сортировки почты с оптическим распознаванием символов применяются с 1990-х годов. Именно отсюда, в конце концов, и происходит знаменитый набор данных рукописных цифр MNIST. То же относится к чтению чеков при зачислении денег в банк и к оценке кредитоспособности заявителей. Финансовые операции автоматически проверяются на мошенничество. На этом держатся многие платёжные системы электронной торговли, такие как PayPal, Stripe, AliPay, WeChat, Apple, Visa и MasterCard. Компьютерные программы для шахмат уже десятилетиями на равных соперничают с людьми. Машинное обучение питает поиск, рекомендации, персонализацию и ранжирование в интернете. Иными словами, машинное обучение повсюду, хотя часто скрыто от глаз.

Лишь недавно ИИ оказался в центре внимания, в основном благодаря решению задач, которые прежде считались неразрешимыми и прямо касаются потребителей. Многие из таких достижений приписывают глубокому обучению.

Умные помощники, такие как Siri от Apple, Alexa от Amazon и ассистент Google, способны отвечать на устные просьбы с разумной точностью. Это и простые поручения вроде включения света, и более сложные задачи, например записать к парикмахеру или вести диалог телефонной поддержки. Вероятно, это самый заметный признак того, что ИИ влияет на нашу жизнь.

Ключевая составляющая цифровых помощников — способность точно распознавать речь. Точность таких систем постепенно выросла до уровня человека в некоторых применениях.

Распознавание объектов тоже прошло долгий путь. Определить, что за объект на картинке, в 2010 году было довольно трудной задачей. На эталонном наборе ImageNet исследователи из NEC Labs и Иллинойского университета в Урбане-Шампейне добились доли ошибок топ-5 в 28%. К 2017 году эту долю ошибок снизили до 2,25%. Столь же поразительные результаты получены в распознавании птичьего пения и в диагностике рака кожи.

ОШИБКА IMAGENET

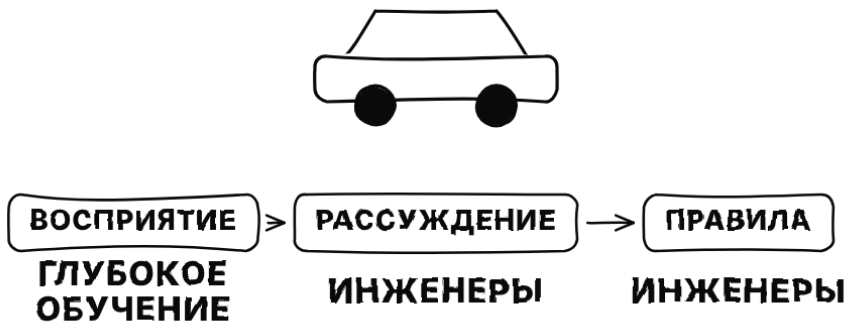


С 28% до 2,25%, и игры

Мастерство в играх когда-то служило мерилом человеческих способностей. Начиная с TD-Gammon, программы для игры в нарды, использовавшей обучение с подкреплением на временных разностях, алгоритмический и вычислительный прогресс привёл к алгоритмам для самых разных применений. По сравнению с нардами у шахмат гораздо более сложное пространство состояний и набор действий. DeepBlue обыграл Гарри Каспарова с помощью массового параллелизма, специализированного оборудования и эффективного поиска по дереву игры. Го ещё труднее из-за огромного пространства состояний. AlphaGo достиг уровня человека в 2015 году, используя глубокое обучение в сочетании с выборкой по дереву методом Монте-Карло. Трудность покера в том, что пространство

состояний велико и наблюдаемо лишь частично (мы не знаем карт соперников). Libratus превзошёл людей в покере с помощью эффективно устроенных стратегий.

Ещё один признак прогресса ИИ — появление беспилотных автомобилей. Хотя полная автономность ещё недостижима, в этом направлении сделаны отличные успехи: такие компании, как Tesla, NVIDIA и Waymo, выпускают продукты с частичной автономностью. Полная автономность так трудна потому, что для правильного вождения нужно уметь воспринимать, рассуждать и встраивать правила в систему. Сейчас глубокое обучение используется в основном в зрительной стороне этих задач. Остальное тщательно настраивают инженеры.



СОЗДАН ДЛЯ ОДНОЙ ЦЕЛИ

Вождению нужно больше, чем зрение

Это лишь верхушка значимых применений машинного обучения. Например, робототехника, логистика, вычислительная биология, физика частиц и астрономия обязаны некоторыми из своих самых впечатляющих недавних успехов хотя бы отчасти машинному обучению, которое тем самым становится вездесущим инструментом инженеров и учёных.

В нетехнических статьях часто поднимаются вопросы о грядущем апокалипсисе ИИ и правдоподобности сингулярности. Опасение в том, что системы машинного обучения каким-то образом обретут сознание и начнут принимать решения, независимо от своих программистов, напрямую влияющие на жизнь людей. В какой-то мере ИИ уже напрямую влияет на жизнь людей: кредитоспособность оценивается автоматически, автопилоты в основном ведут транспорт, решения об освобождении под залог используют статистические данные. Если о более легкомысленном, мы можем попросить Alexa включить кофеварку.

К счастью, мы далеки от обладающей сознанием системы ИИ, которая могла бы намеренно манипулировать своими создателями-людьми. Во-первых, системы ИИ проектируются, обучаются и развёртываются под конкретную цель. Хотя их поведение может создавать иллюзию общего интеллекта, в основе их устройства лежит сочетание правил, эвристик и статистических моделей. Во-вторых, сейчас просто нет инструментов общего искусственного интеллекта, которые могли бы сами себя совершенствовать, рассуждать о себе и менять, расширять и улучшать собственную архитектуру, пытаясь решать общие задачи.

Гораздо более насущный вопрос — как ИИ используется в нашей повседневной жизни. Вероятно, многие рутинные задачи, которые сейчас выполняют люди, могут быть и будут автоматизированы. Сельскохозяйственные роботы, вероятно, снизят затраты фермеров, выращивающих органическую продукцию, но они же автоматизируют уборку урожая. Этот этап промышленной революции может иметь глубокие последствия для широких слоёв общества, поскольку во многих странах простой труд даёт много рабочих мест. Кроме того, статистические модели, применённые неосторожно, могут приводить к расовым, гендерным или возрастным перекосам и вызывать обоснованные опасения о процедурной справедливости, если на них автоматически опираются важные решения. Важно следить за тем, чтобы эти алгоритмы применялись осторожно. С учётом того, что мы знаем сегодня, это кажется нам гораздо более насущной заботой, чем возможность того, что злонамеренный сверхинтеллект уничтожит человечество.

Вопросы

1. Почему текст называет машинное обучение «вездесущим, хотя часто скрытым от глаз»?
 - а. Потому что оно используется только в исследовательских лабораториях
 - б. Потому что оно незаметно управляет сортировкой почты, проверкой на мошенничество, поиском и рекомендациями

- c. Потому что люди отказываются им пользоваться
 - d. Потому что оно стало возможным только в 2017 году
2. Что, согласно тексту, делало покер трудным для компьютеров?
- a. Его пространство состояний крошечное
 - b. Он требует специализированного оборудования
 - c. Его пространство состояний велико и наблюдаемо лишь частично, ведь карты соперников скрыты
 - d. В нём нет наград
3. Какую главную причину называют авторы, чтобы не бояться обладающего сознанием ИИ, который манипулирует своими создателями?
- a. Системы ИИ создаются под конкретные цели, а инструментов для самосовершенствующегося общего интеллекта пока нет
 - b. Системы ИИ всегда выключают на ночь
 - c. ИИ никогда не использовался в реальных решениях
 - d. Законы запрещают ИИ принимать решения
4. Что авторы считают более насущной заботой, чем злонамеренный сверхинтеллект?
- a. Что у компьютеров кончится память
 - b. Что игры станут слишком лёгкими
 - c. Что ИИ перестанет совершенствоваться
 - d. Автоматизацию рутинного труда и предвзятые модели, принимающие важные решения

Ответы

1. **b)** Сортировка почты, чтение чеков, оценка кредитоспособности, проверка на мошенничество, поиск, рекомендации и ранжирование — всё это использует машинное обучение, в основном незаметно.
2. **c)** Трудность покера в том, что пространство состояний велико и наблюдаемо лишь частично: мы не знаем карт соперников.
3. **a)** Системы ИИ проектируются и развёртываются под конкретную цель, и нет инструментов общего искусственного интеллекта, которые могли бы сами себя улучшать и менять.
4. **d)** Насущная забота — как ИИ используется в повседневной жизни: автоматизация многих рутинных задач и модели, применённые неосторожно и ведущие к расовым, гендерным или возрастным перекосам.

Разобранные примеры

Прогресс на ImageNet На ImageNet доля ошибок топ-5 была 28% в 2010 году и 2,25% к 2017 году. Примерно сколько из 1000 тестовых картинок модель не распознала в каждом году и во сколько раз уменьшилась ошибка?

В 2010 году: 28% от 1000 — это $0,28 \times 1000 = 280$ нераспознанных картинок.

К 2017 году: 2,25% от 1000 — это $0,0225 \times 1000 = 22,5$, то есть около 22 или 23 нераспознанных картинок.

Отношение: $28 \div 2,25 \approx 12,4$.

Ответ: Около 280 ошибок в 2010 году и около 22 в 2017-м; ошибка уменьшилась примерно в 12 раз

Задачи

1. Объясните, почему, согласно тексту, полностью беспилотное вождение труднее, чем распознавание объектов на картинках.

Подсказка: Для вождения нужно больше, чем видеть.

Ответ: Для правильного вождения нужно уметь воспринимать, рассуждать и встраивать правила в систему. Глубокое обучение используется в основном для зрительной части, восприятия; остальное по-прежнему тщательно настраивают инженеры.

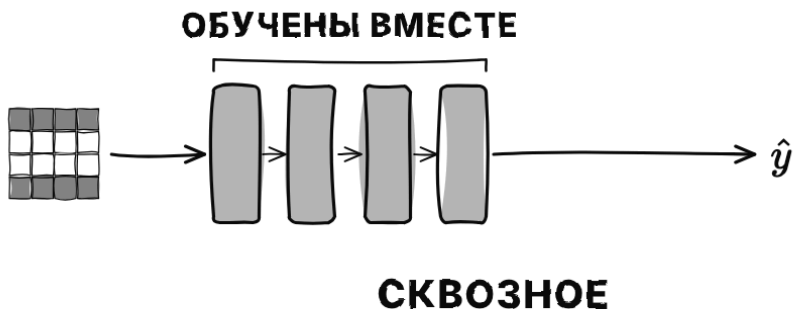
2. Назовите три способа, упомянутых в тексте, которыми ИИ уже напрямую влияет на жизнь людей.

Подсказка: Подумайте о деньгах, поездках и судах.

Ответ: Кредитоспособность оценивается автоматически, автопилоты в основном ведут транспорт, а решения об освобождении под залог используют статистические данные.

Суть глубокого обучения

До сих пор мы говорили о машинном обучении в общих чертах. Глубокое обучение — это часть машинного обучения, занимающаяся моделями на основе многослойных нейронных сетей. Оно глубокое именно в том смысле, что его модели выучивают много слоёв преобразований. Хотя это может звучать узко, глубокое обучение породило головокружительное множество моделей, приёмов, постановок задач и применений. Было выработано много интуитивных объяснений пользы глубины. Можно сказать, что в любом машинном обучении есть много слоёв вычислений, первый из которых — шаги обработки признаков. Глубокое обучение отличается тем, что операции на каждом из многих слоёв представлений выучиваются по данным совместно.



Ручные признаки и сквозное обучение

Задачи, которые мы обсуждали до сих пор, — обучение на необработанном звуковом сигнале, на необработанных значениях пикселей изображений или сопоставление предложений произвольной длины с их соответствиями на других языках, — это те задачи, где глубокое обучение блистает, а традиционные методы буксуют. Оказывается, эти многослойные модели способны работать с низкоуровневыми данными восприятия так, как прежние инструменты не могли. Пожалуй, самое важное, что объединяет методы глубокого обучения, — сквозное обучение. То есть вместо того, чтобы собирать систему из компонентов, настроенных по отдельности, систему строят целиком и затем настраивают её работу совместно. Например, в компьютерном зрении учёные раньше отделяли процесс

конструирования признаков от процесса построения моделей машинного обучения. Детектор границ Кэнни и экстрактор признаков SIFT Лоу больше десятилетия безраздельно царили как алгоритмы перевода изображений в векторы признаков. В былые времена главное в применении машинного обучения к этим задачам состояло в том, чтобы придумать вручную сконструированные способы перевести данные в форму, удобную для неглубоких моделей. К сожалению, человеческая изобретательность может достичь лишь немногого по сравнению с последовательной проверкой миллионов вариантов, которую автоматически выполняет алгоритм. Когда глубокое обучение взяло верх, эти экстракторы признаков заменили автоматически настраиваемые фильтры, давшие лучшую точность.

Таким образом, одно из ключевых преимуществ глубокого обучения в том, что оно заменяет не только неглубокие модели в конце традиционных конвейеров обучения, но и трудоёмкий процесс конструирования признаков. Более того, заменив значительную часть специфической для предметной области предобработки, глубокое обучение стёрло многие границы, разделявшие прежде компьютерное зрение, распознавание речи, обработку естественного языка, медицинскую информатику и другие прикладные области, и тем самым предложило единый набор инструментов для решения разных задач.

Помимо сквозного обучения, мы переживаем переход от параметрических статистических описаний к полностью непараметрическим моделям. Когда данных мало, чтобы получить полезные модели, приходится опираться на упрощающие предположения о реальности. Когда данных много, их можно заменить непараметрическими моделями, которые

лучше соответствуют данным. В какой-то мере это повторяет прогресс, который физика пережила в середине прошлого века с появлением компьютеров. Вместо того чтобы вручную решать параметрические приближения поведения электронов, теперь можно прибегнуть к численному моделированию соответствующих уравнений в частных производных. Это привело к гораздо более точным моделям, хотя часто ценой понятности.

Ещё одно отличие от прежних работ — готовность принимать неоптимальные решения, иметь дело с невыпуклыми нелинейными задачами оптимизации и пробовать что-то, прежде чем это доказывать. Этот новообретённый эмпиризм в обращении со статистическими задачами вместе с быстрым притоком талантов привёл к стремительному прогрессу в разработке практических алгоритмов, хотя во многих случаях ценой переделки и повторного изобретения инструментов, существовавших десятилетиями.

В конце концов, сообщество глубокого обучения гордится тем, что делится инструментами поверх академических и корпоративных границ, выпуская в открытый доступ множество отличных библиотек, статистических моделей и обученных сетей. Именно в этом духе блокноты, из которых состоит эта книга, свободно доступны для распространения и использования. Мы много работали, чтобы снизить барьеры для всех, кто хочет изучить глубокое обучение, и надеемся, что нашим читателям это пойдёт на пользу.

Подведём итог: машинное обучение изучает, как компьютерные системы могут использовать опыт (часто данные), чтобы лучше

выполнять конкретные задачи. Оно объединяет идеи статистики, интеллектуального анализа данных и оптимизации. Часто оно служит средством реализации решений ИИ. Обучение представлений, как класс машинного обучения, сосредоточено на том, как автоматически находить подходящий способ представления данных. Глубокое обучение, понимаемое как многоуровневое обучение представлений через выучивание многих слоёв преобразований, заменяет не только неглубокие модели в конце традиционных конвейеров машинного обучения, но и трудоёмкий процесс конструирования признаков. Значительную часть недавнего прогресса в глубоком обучении вызвали изобилие данных от дешёвых датчиков и приложений масштаба интернета, а также существенный прогресс вычислений, в основном благодаря GPU. Кроме того, появление эффективных фреймворков глубокого обучения значительно упростило проектирование и реализацию оптимизации всей системы, а это ключевая составляющая высокой производительности.

Вопросы

1. Если в любом машинном обучении, можно сказать, есть много слоёв вычислений, что отличает глубокое обучение?
 - a. Оно использует только один слой
 - b. Операции на каждом из его многих слоёв представлений выучиваются по данным совместно
 - c. Оно никогда не использует признаки
 - d. Каждый его слой должен проектировать человек

2. Что означает сквозное обучение?
- a. Построить систему целиком и настроить её работу совместно
 - b. Настроить каждый компонент отдельно, а потом их собрать
 - c. Обучать только последний слой модели
 - d. Один раз пройти набор данных от начала до конца
3. Почему автоматически настраиваемые фильтры победили ручную сконструированные экстракторы признаков вроде SIFT?
- a. Люди перестали заниматься зрением
 - b. Ручные признаки работали слишком быстро
 - c. Алгоритм может последовательно проверить миллионы вариантов — больше, чем позволяет человеческая изобретательность
 - d. Экстракторы признаков было запрещено использовать
4. Когда упрощающие предположения о реальности можно заменить непараметрическими моделями?
- a. Когда данных мало
 - b. Когда модели должны быть легко понятны
 - c. Когда нет компьютеров
 - d. Когда данных много

Ответы

1. b) Глубокое обучение отличается тем, что операции на каждом из многих слоёв представлений выучиваются по данным совместно.

- 2. а)** Вместо того чтобы собирать систему из компонентов, настроенных по отдельности, систему строят и затем настраивают её работу совместно.
- 3. с)** Человеческая изобретательность может достичь лишь немногого по сравнению с последовательной проверкой миллионов вариантов, которую автоматически выполняет алгоритм.
- 4. d)** Когда данных мало, приходится опираться на упрощающие предположения; когда данных много, их можно заменить непараметрическими моделями, которые лучше соответствуют данным.

Задачи

- 1.** Своими словами объясните, какие две вещи заменяет глубокое обучение в традиционном конвейере машинного обучения.

Подсказка: Подумайте, что происходит до конвейера и в его конце.

Ответ: Оно заменяет неглубокую модель в конце конвейера, а также трудоёмкое конструирование признаков в его начале, где люди вручную придумывали, как превратить необработанные данные, например изображения, в векторы признаков.

- 2.** Текст сравнивает переход к непараметрическим моделям с переменной в физике. Опишите это сравнение и цену, которую пришлось заплатить в обоих случаях.

Подсказка: Что компьютеры позволили физикам делать вместо решения вручную?

Ответ: Физики перестали вручную решать упрощённые параметрические приближения поведения электронов и перешли к численному моделированию исходных уравнений, так же как машинное обучение переходит от упрощённых параметрических моделей к моделям, подстраивающимся под обильные данные. В

обоих случаях модели становятся гораздо точнее, но часто ценой понятности.