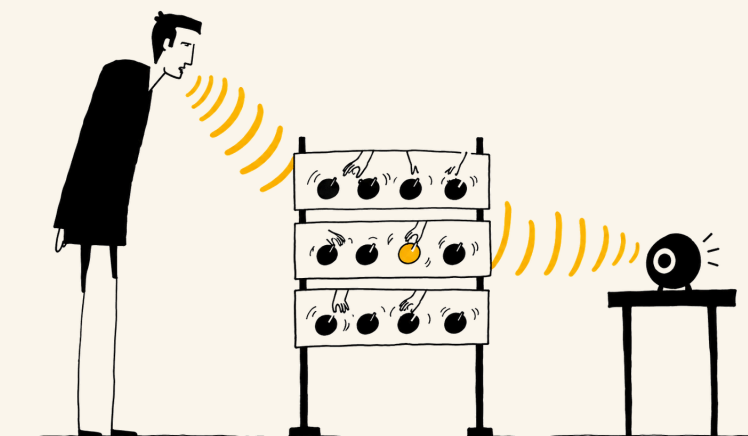


Погружение в глубокое обучение



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

Правила, которые никогда не учатся

Ручки вместо правил

На чём учится машина

Как оценить машину

Нащупывая путь вниз

Сколько

Что из этого и насколько уверенно

Много меток и лучшее первым

Ваша лента и то, что разворачивается во времени

Всё наоборот и обучение без ответов

Когда машина действует

Слепые пятна, риск и старые корни

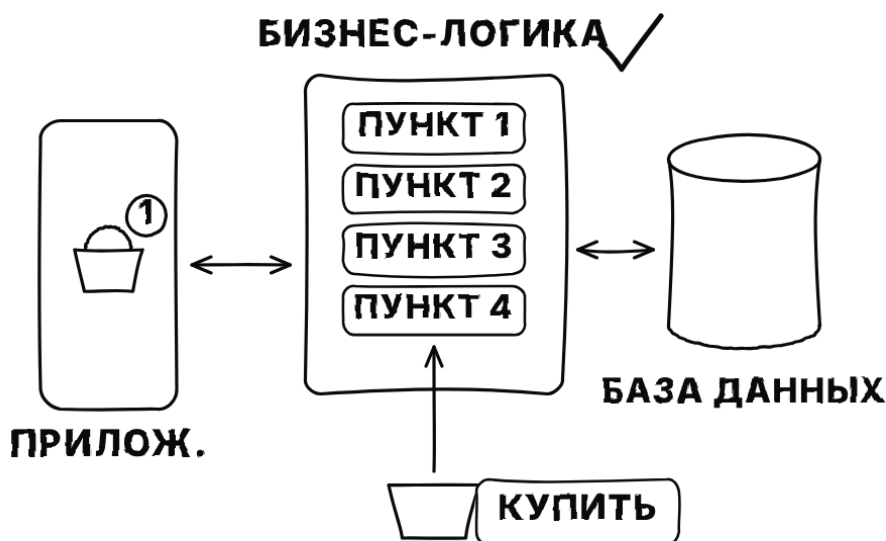
Из тупика — к взрыву

Тихие успехи и настоящие тревоги

Что делает его глубоким

Правила, которые никогда не учатся

Откройте на телефоне любое приложение магазина, положите в корзину пару носков и посмотрите, как маленькая цифра на значке корзины прыгает с нуля на единицу. Можно повторить это сто раз, и она будет прыгать каждый раз, ровно одинаково. Где-то программист написал правило для этого момента: когда этот человек нажимает «добавить», записать в таблицу новую строку, которая связывает этого человека с этими носками. От того, что вы этим правилом пользуетесь, в нём ничего не меняется. Приложение не начинает лучше добавлять носки; оно просто делает то, что ему велели.



Магазин, построенный из правил

Почти всю историю вычислительной техники едва ли не каждая программа, с которой вы сталкивались за день, была устроена именно так: как длинный список правил, написанных людьми. Книга рисует команду перед маркерной доской, которая проектирует интернет-магазин. Они решают, что покупатели будут видеть магазин в браузере или в приложении на телефоне, что база данных будет помнить каждого покупателя и каждую прошлую покупку, а посередине будет то, что авторы называют бизнес-логикой, — мозг магазина. Этот мозг — набор правил, которые говорят программе, что делать в каждой ситуации, какую команда сможет придумать. Потом команда выискивает неудобные случаи, например покупателя, который нажимает «купить» с пустой корзиной, и добавляет правило на каждый. С

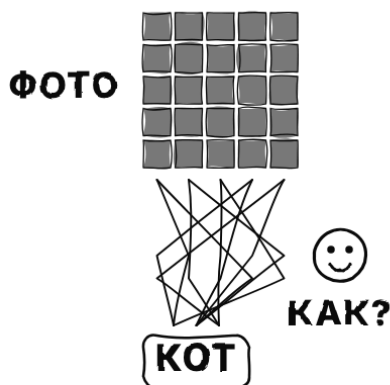
первого раза всё редко работает идеально, но после некоторого тестирования работает, и магазин может открыться раньше, чем в него зайдёт хоть один настоящий покупатель. Авторы называют это замечательным достижением человеческой мысли и добавляют меткий совет: если вы умеете писать правила, которые верны в ста процентах случаев, машинное обучение вам вообще не нужно.

А теперь представьте ту же умную команду у той же доски, но с другими задачами. Предсказать завтрашнюю погоду по картам, спутниковым снимкам и измерениям за последние несколько недель. Прочитать вопрос, набранный обычными словами, и правильно на него ответить. Посмотреть на фотографию, найти на ней каждого человека и обвести каждого контуром. Показать покупателю вещи, которые ему, скорее всего, понравятся, но на которые он никогда не наткнулся бы, просто листая каталог. Последнюю задачу ваша лента рекомендаций пытается решать каждый раз, когда вы её открываете. Для таких задач даже лучшим программистам мира было бы трудно написать правила с нуля.

ЦЕЛЬ ДВИЖЕТСЯ



ЗАПУТАНО

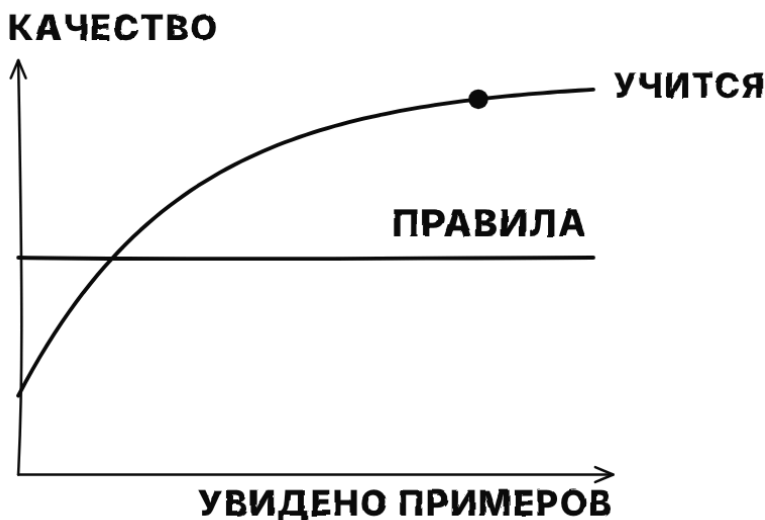


Две причины, почему правила не работают

Книга называет две причины. Первая: правильный ответ может всё время смещаться. То, чего покупатели хотят этой весной, — не то, чего они хотели прошлой весной, поэтому любой застывший набор правил устаревает, и хорошее решение должно подстраиваться, когда меняется мир. Спам-фильтр хорошо это знает: стоит ему научиться ловить один вид мусорных писем, как отправители меняют формулировки, и вчерашнее правило перестаёт работать. Вторая причина: связь между тем, что поступает на вход, и тем, что должно выйти, может быть слишком запутанной, чтобы её записать. Фотография — это просто сетка цветных точек, а слово «кошка» — это понятие; между ними могут лежать миллионы мелких вычислений по принципам, которые никто никогда не формулировал. Вы сами

без труда узнаёте лицо друга в толпе, но не смогли бы записать шаги, которые проделал для этого ваш мозг. Умение есть, но оно скрыто от вашего собственного сознания.

Вот здесь и появляется другая идея. Машинное обучение — это наука о методах, которые учатся на опыте. Опыт здесь обычно означает данные, то есть множество записанных примеров, или результаты действий в какой-то среде и наблюдение за тем, что получилось. Чем больше опыта набирает такой метод, тем лучше он работает. Сравните это с кнопкой для носков. Интернет-магазин следует одной и той же бизнес-логике, сколько бы миллионов покупателей через него ни прошло, и меняется он только тогда, когда его программисты что-то узнают и решают её переписать. Обучающаяся система, наоборот, улучшается на тех самых примерах, которые видит. Когда голосовой помощник начинает лучше понимать ваш акцент, ни один программист не писал особого правила для вашего акцента; он стал лучше, потому что его обучили на большом объёме записанной речи.



Застывшие правила против обучения

Книга ставит целью научить основам машинного обучения, и прежде всего одному мощному семейству его методов, которое называется глубоким обучением. Именно эти методы стоят за большей частью сегодняшних успехов в том, чтобы научить компьютеры видеть, работать с человеческим языком, помогать в медицине и читать генетическую информацию. Чат-бот, которому вы печатаете вопросы, — один из плодов глубокого обучения: никто не писал его ответы в виде правил; он усвоил закономерности языка из огромного количества текста. Так что разделительную линию провести просто. С одной стороны — программы, поведение которых люди задали заранее; с другой — программы, поведение которых сложилось из примеров.

Запомним

Большинство программ — это списки правил, написанных людьми, и ведут они себя одинаково, сколько бы ими ни пользовались. Правила хорошо работают, когда люди могут предвидеть каждый случай и каждый раз решать его верно, и тогда машинное обучение не нужно. Некоторые задачи не поддаются правилам: либо правильный ответ всё время меняется, либо путь от входа к ответу слишком запутан, чтобы его записать, даже если наш собственный мозг справляется без усилий. Машинное обучение — это наука о методах, которые улучшаются с опытом, обычно на данных. Глубокое обучение — семейство таких методов, которому посвящена эта книга.

Вопросы

1. Почему кнопка «добавить в корзину» не становится лучше, сколько бы покупателей ею ни пользовалось?
 - a. Потому что она следует заранее написанным правилам, которые меняются, только когда их переписывают программисты
 - b. Потому что она ещё не набрала достаточно покупателей, чтобы учиться
 - c. Потому что база данных забывает каждую покупку сразу после неё

- d. Потому что покупки — задача, которая меняется слишком быстро, чтобы ей научиться
2. Команда может написать для задачи правила, верные каждый раз без исключения. Что советует книга?
- a. Всё равно обучить обучающуюся систему, потому что она справится ещё лучше
 - b. Для надёжности смешать правила с обучающейся системой
 - c. Использовать правила: для этой задачи машинное обучение не нужно
 - d. Сначала собрать данные, а потом решить, какие правила оставить
3. Узнать лицо друга вам легко. Почему же это трудно записать в виде правил?
- a. Потому что лица меняются слишком часто для любой программы
 - b. Потому что шаги, которые проделывает ваш мозг, скрыты от вашего сознания
 - c. Потому что компьютеры не умеют хранить фотографии
 - d. Потому что лиц слишком мало, чтобы на них учиться
4. Какую вторую причину, кроме запутанной связи между входом и ответом, лекция называет для того, почему правила не работают?
- a. Правила слишком долго выполняются на компьютере
 - b. Программисты редко проверяют свои правила
 - c. В базах данных не помещается достаточно правил

d. Правильный ответ всё время меняется, поэтому застывшие правила устаревают

Ответы

1. a) Магазин следует одной и той же бизнес-логике, сколько бы покупателей через него ни прошло; меняется он, только когда программисты решают её переписать.
2. c) Если вы умеете писать правила, которые верны в ста процентах случаев, машинное обучение вам вообще не нужно.
3. b) Вы узнаете лицо без труда, но не смогли бы записать шаги, которые проделал ваш мозг; умение скрыто от вашего сознания.
4. d) Иногда правильный ответ всё время смещается, поэтому любой застывший набор правил устаревает, и хорошее решение должно подстраиваться, когда меняется мир.

Задачи

1. Возьмите четыре задачи с доски: завтрашняя погода, ответ на набранный вопрос, контур каждого человека на фото и подбор товаров, которые покупатель не нашёл бы сам, листая каталог. Для каждой скажите, какая из двух причин мешает писать правила вручную: ответ всё время меняется или связь входа с ответом слишком запутана, чтобы её записать.

Подсказка: Спросите про каждую задачу: смещается ли правильный ответ со временем или вы просто не смогли бы перечислить шаги, хотя ответ неизменен?

Ответ: Погода и вкусы покупателей со временем меняются, так что застывшие правила устаревают; к тому же погода зависит от слишком многих запутанно связанных измерений. Ответы на свободные вопросы и поиск людей на фото — в основном

запутанный случай: мы сами делаем это легко, но не можем расписать шаги. У нескольких задач есть обе трудности сразу.

2. Объясните своими словами, почему случай с пустой корзиной относится к обычному программированию, а не к машинному обучению.

Подсказка: Может ли команда предвидеть этот случай и раз и навсегда записать правильную реакцию?

Ответ: Нажатие «купить» с пустой корзиной — ситуация, которую команда может предвидеть ещё до прихода покупателей, а правильную реакцию, например показать сообщение, можно записать, и она будет верна каждый раз. Такие задачи решаются правилами; машинное обучение нужно для задач, где таких правил не написать никому.

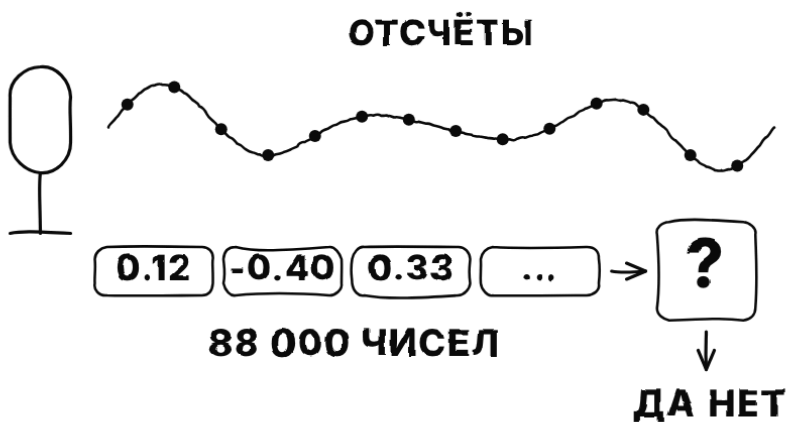
3. Спам-фильтр сделан двумя способами: один — с неизменным списком запрещённых фраз, написанным программистами, другой — обученный на множестве писем с пометками «спам» и «не спам». Что происходит с каждым, когда спамеры месяц за месяцем меняют формулировки?

Подсказка: Какой из них меняется, только когда его переписывает человек, а какой может улучшаться на новых примерах?

Ответ: Неизменный список остаётся прежним, пока кто-то его не перепишет, поэтому новые формулировки проскальзывают, и он устаревает. Обученный фильтр может улучшаться, когда ему дают новые размеченные примеры, потому что обучающийся метод становится лучше с опытом.

Ручки вместо правил

Скажите через всю комнату «Hey Siri» или «OK Google» и посмотрите, как загорается ваш телефон. Он проснулся, потому что выделил одну определённую фразу из всего окружающего звука: телевизора, чайника, разговоров других людей. А теперь попробуйте представить, что пишете такую программу сами, в одиночку, в комнате, где нет ничего, кроме компьютера. Каким было бы первое правило?



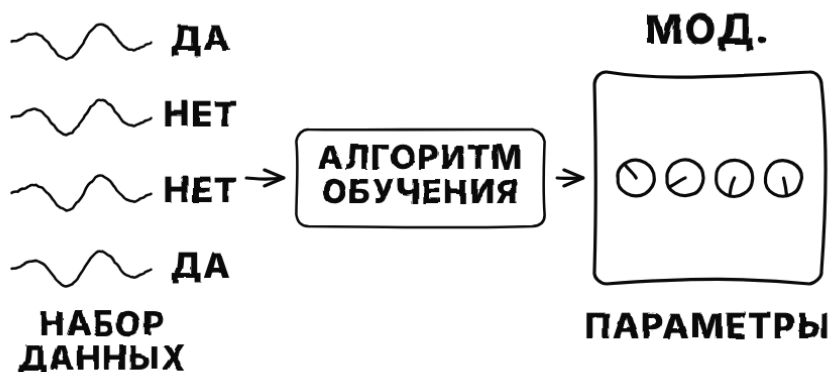
На входе звук, на выходе «да» или «нет»

Авторы начинают раздел с небольшой сценки, которую, как они признаются, придумали для наглядности. По дороге за кофе один из них говорит телефону «Hey Siri», и телефон просыпается. Другой спрашивает дорогу до кофейни. Телефон записывает, что он сказал, понимает, что ему нужен маршрут, открывает карты, находит несколько путей и рядом с каждым показывает предсказанное время в пути. За эти несколько секунд поработали несколько отдельных обученных программ: одна разбудила телефон, другая превратила речь в текст, третья поняла, чего хотят, четвёртая оценила время в пути. То же самое происходит, когда вы просите голосового помощника поставить таймер: прислушивание к фразе пробуждения, запись ваших слов и понимание того, что вы имеете в виду, — всё это делает обученная программа.

Возьмём только первую из этих задач — слово пробуждения. Микрофон телефона измеряет звук примерно сорок четыре тысячи раз в секунду, и каждое измерение — одно число: насколько звуковая волна отклонена вверх или вниз в это мгновение. Значит, двухсекундный отрывок звука — это сорок четыре тысячи, умноженные на два, около восьмидесяти восьми тысяч чисел подряд. Программа должна посмотреть на эти восемьдесят восемь тысяч чисел и ответить только «да» или «нет»: было ли там слово пробуждения? Никто не знает правила, которое делало бы это надёжно. Книга честно это признаёт: авторы тоже не знают, как написать такую программу с нуля, и именно поэтому применяют машинное обучение.

И вот в чём хитрость. Мы не можем объяснить компьютеру, как распознать «Алеха», но сами распознаём это слово без труда. Поэтому люди прослушивают огромное количество коротких

записей и помечают каждую: «да» или «нет». Эту грудку записей с пометками называют набором данных, а каждую пометку — меткой. Затем вместо того чтобы писать распознаватель, мы пишем гибкую программу, поведение которой зависит от множества настраиваемых чисел — параметров. Набор данных используют, чтобы найти такие значения параметров, при которых программа работает лучше всего по выбранной мере качества.



Метки, ручки и ученик

Книга предлагает представлять параметры как ручки. Поверните ручки в одну сторону — программа ведёт себя одним образом; поверните иначе — ведёт себя по-другому. Когда ручки выставлены и больше не трогаются, программу называют

моделью. Все программы, которые можно получить, вращая ручки одной конструкции, образуют семейство моделей. А программа, которая по данным выбирает положения ручек, называется алгоритмом обучения — это программа, результатом работы которой становится другая программа.

Прежде чем всё это запустить, задачу нужно точно определить: что поступает на вход, что выходит и какое семейство моделей использовать. Здесь на входе — отрывок звука, а на выходе — «да» или «нет». Если семейство выбрано удачно, при одном положении ручек модель будет говорить «да» каждый раз, когда слышит «Alexa». А поскольку выбор слова пробуждения произволен, достаточно богатое семейство должно иметь и другое положение ручек, при котором «да» звучит только на «Apricot». Эти две задачи так похожи, что одно и то же семейство должно справиться с обеими. Но превращение фотографий в подписи или английских предложений в китайские связано с совсем другими входами и выходами и может потребовать совершенно другого семейства.

Опыт

Попросите кого-нибудь прошептать слово в шумной комнате, потом пробормотать его, потом выкрикнуть из соседней комнаты. Вы узнаете его каждый раз, но если попытаете описать словами, что делает звук именно «Alexa», а не «Alexis», окажется, что записать нечего. Этот разрыв между тем, чтобы уметь что-то делать, и тем, чтобы уметь сформулировать правило, и есть причина, по которой существует обучение на примерах.

Когда ручки выставлены случайно, модель вообще ничего не распознаёт. Обучение — это поиск такого положения ручек, при котором модель ведёт себя так, как мы хотим, а делать это с помощью данных называется тренировкой. Книга изображает тренировку в виде цикла. Начните с модели, у которой ручки выставлены случайно и которая не делает ничего полезного. Возьмите горсть примеров — записи с метками «да» или «нет». Чуть подкрутите ручки, чтобы модель работала на этих примерах немного лучше. Потом возьмите ещё примеров и снова подкрутите, и так круг за кругом, пока модель не станет очень хорошей.

Так что вместо того чтобы писать распознаватель слова пробуждения, мы пишем программу, которая учится стать им, если показать ей достаточно размеченных примеров. Авторы называют это программированием с помощью данных. Точно так же можно «запрограммировать» детектор, отличающий кошек от собак, просто показав ему много картинок тех и других. В итоге он выдаёт большое положительное число, когда видит кошку, большое отрицательное — когда видит собаку, и число около нуля, когда не уверен. Ваш спам-фильтр работает так же: никто не перечислял все спамерские фразы; ему показали много писем с пометками «спам» и «не спам», и его ручки встали туда, где эти письма разделяются. Книга добавляет, что глубокое обучение, о котором речь пойдёт дальше, — лишь один из многих популярных способов это делать.

Запомним

Даже несколько секунд с телефоном могут задействовать несколько обученных программ, от пробуждения по фразе до оценки времени в пути. Слово пробуждения приходит как десятки тысяч чисел в секунду, и никто не знает правила, которое надёжно превращало бы их в «да» или «нет». Мы сами можем узнать слово, поэтому размечаем много примеров и даём гибкой программе с настраиваемыми ручками, её параметрами, учиться на них. Модель — это программа с выставленными ручками; алгоритм обучения выбирает положения; тренировка повторяет небольшие подстройки на примерах, пока модель не начнёт работать хорошо. Это программирование с помощью данных, а глубокое обучение — один из способов его делать.

Вопросы

1. Зачем люди размечают записи метками «да» или «нет» перед тренировкой детектора слова пробуждения?
 - a. Потому что компьютер не может хранить неразмеченный звук
 - b. Потому что мы сами узнаём слово, хотя не можем записать правило
 - c. Потому что метки делают каждую запись короче
 - d. Потому что микрофон записывает только размеченный звук

2. Чем модель отличается от семейства моделей?

- a. Семейство — это модель, обученная на большем количестве данных
- b. Модель — это программа с выставленными ручками; семейство — все программы, которыми может стать одна конструкция при повороте ручек
- c. Модель отвечает «да» или «нет», а семейство — числами
- d. Семейство — это группа наборов данных, используемых вместе

3. Почему одно семейство моделей должно справиться и с «Alexa», и с «Apricot», но, возможно, не с подписями к фотографиям?

- a. Потому что подписям нужно больше ручек, чем может быть у любого семейства
- b. Потому что два слова пробуждения — похожие задачи с одинаковым видом входа и выхода, а у подписей к фото входы и выходы совсем другие
- c. Потому что «Apricot» легче расслышать, чем «Alexa»
- d. Потому что фотографии нельзя превратить в числа

4. Что происходит с ручками на каждом круге цикла тренировки?

- a. Им всем дают новые случайные значения
- b. Их чуть подкручивают, чтобы модель немного лучше работала на только что показанных примерах
- c. Их выставляет программист, прослушав примеры
- d. Их не трогают, пока собираются новые данные

Ответы

- 1. b)** Мы не можем объяснить компьютеру, как узнавать слово, но сами его узнаём, поэтому помечаем много записей и даём программе учиться на них.
- 2. b)** Когда ручки выставлены, программу называют моделью; все программы, которые можно получить, вращая ручки одной конструкции, образуют семейство.
- 3. b)** Две задачи со словами пробуждения похожи, так что хватает другого положения тех же ручек, а превращение фото в подписи связано с совсем другими входами и выходами.
- 4. b)** Берут горсть размеченных примеров и чуть подкручивают ручки, чтобы модель работала на них немного лучше, а потом повторяют.

Разобранные примеры

Сколько чисел в отрывке звука Микрофон измеряет звук примерно 44 000 раз в секунду, и каждое измерение — одно число. На сколько чисел должен посмотреть детектор слова пробуждения в трёхсекундном отрывке?

Каждая секунда даёт около 44 000 измерений, каждое — одно число.

Три секунды дают в три раза больше.

44 000 умножить на 3 — это 132 000.

Все эти числа нужно превратить в одно «да» или «нет», и потому никакое простое правило не работает.

Ответ: около 132 000 чисел

Задачи

- 1.** Расставьте по порядку четыре шага цикла тренировки и объясните, почему первая модель бесполезна: подкрутить ручки;

начать со случайных ручек; повторять, пока не станет хорошо; взять несколько размеченных примеров.

Подсказка: Что может модель, пока она не видела никаких данных?

Ответ: Начать со случайных ручек, взять несколько размеченных примеров, подкрутить ручки, чтобы модель работала на них лучше, и повторять, пока она не станет хорошей. Первая модель бесполезна, потому что её ручки выставлены случайно и она ещё ничему не научилась.

2. Детектор кошек и собак выдаёт число для каждой картинки. Для одной картинки он даёт 9, для другой минус 8, для третьей 0,3. Что, по его мнению, изображено на каждой?

Подсказка: Большое положительное число — кошка, большое отрицательное — собака, около нуля — не уверен.

Ответ: На первой — кошка (большое положительное число), на второй — собака (большое отрицательное), а насчёт третьей он не уверен, так как число близко к нулю.

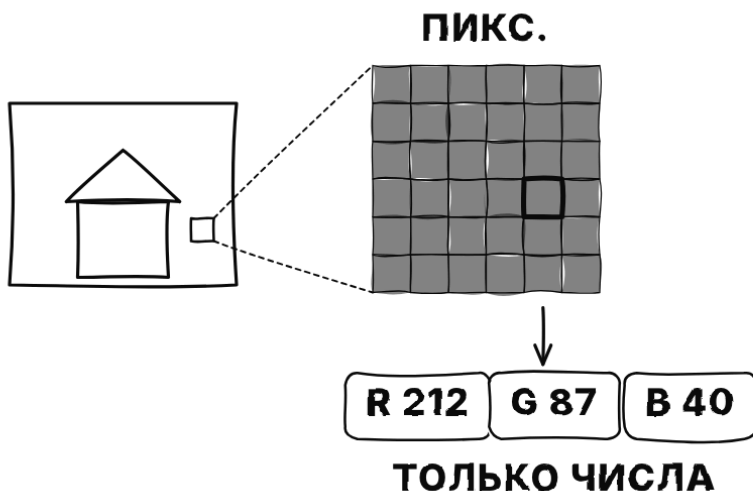
3. Объясните своими словами, почему авторы называют тренировку «программированием с помощью данных».

Подсказка: Кто определяет поведение программы: человек, пишущий правила, или что-то другое?

Ответ: Поведение программы не пишет человек правило за правилом; его задают, показывая программе размеченные примеры и позволяя алгоритму обучения выбрать положения ручек. Данные занимают место написанных вручную правил.

На чём учится машина

Сделайте снимок телефоном и увеличьте его, насколько получится. Где-то по пути картинка рассыпается на маленькие цветные квадратики — пиксели. Каждый квадратик на самом деле — три числа: насколько ярок в нём красный свет, насколько зелёный и насколько синий. Для обучающейся программы ваша фотография — только это: длинный список чисел.

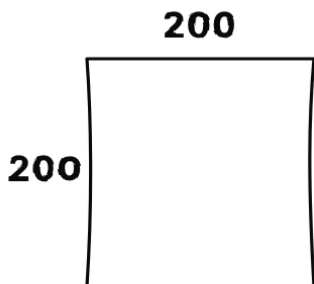


Фотография — это только числа

Детектор слова пробуждения из прошлой части учился на записях с пометками «да» или «нет». У такого рода задач есть

название: обучение с учителем. У нас есть много примеров, правильные ответы для которых известны, и мы хотим программу, которая предскажет правильный ответ для новых примеров, где он неизвестен. Это лишь один из многих видов задач машинного обучения, но сначала книга называет четыре составляющие, которые встречаются в любом виде. Есть данные, на которых учатся. Есть модель, которая превращает данные в ответ. Есть целевая функция — оценка, которая говорит, насколько хорошо или плохо работает модель. И есть алгоритм, который подстраивает параметры модели, её ручки, чтобы эта оценка стала как можно лучше. Эта часть — о первой из четырёх составляющих, о данных.

Данные — это собрание примеров, и чтобы от них была польза, каждый пример нужно превратить в числа. То, что модели позволено видеть в примере, называют его признаками. В обучении с учителем есть ещё одна особая величина, которую модель должна предсказать и которую ей не показывают, — метка. В сортировщике фотографий признаки — это сама фотография, а метка — число, обозначающее её категорию. Книга подсчитывает, во что превращается фотография. В цветной картинке шириной двести пикселей и высотой двести пикселей двести на двести пикселей, то есть сорок тысяч. Каждый пиксель несёт три числа, для красного, зелёного и синего, так что сорок тысяч на три даёт сто двадцать тысяч чисел для одной маленькой картинки.

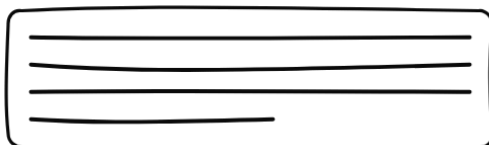


$$200 \times 200 = 40,000$$

$$40,000 \times 3 = 120,000$$

ОТСТОЙ!

КОРОТКО



ДЛИННО

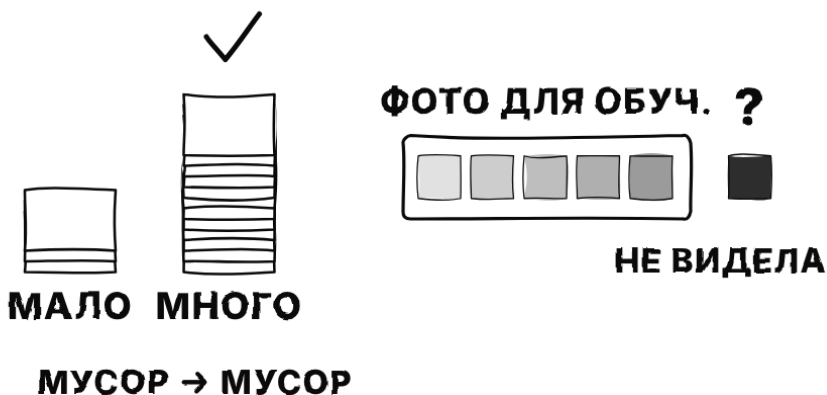
$$200 \times 200 \times 3 = 120\,000$$

Второй пример книги — из больницы. Задача — предсказать, проживёт ли пациент следующие тридцать дней. Признаки — то, что уже есть в карте пациента: возраст, пульс и давление, другие болезни, текущие лекарства, недавние операции. Метка для тренировки — просто прожил ли каждый прошлый пациент эти тридцать дней, «да» или «нет». Та же схема лежит в основе вашего спам-фильтра: признаки описывают каждое письмо, а метка — «спам» или «не спам».

Когда каждый пример описан одинаковым числом признаков, примеры называют примерами фиксированной длины, а это число — размерностью данных. Для маленьких фотографий она равна ста двадцати тысячам; для карт пациентов — числу

пунктов в каждой карте. Фиксированная длина облегчает жизнь, но многие данные в неё не укладываются. Снимки с микроскопа могут получаться одного размера, но картинки, собранные в интернете, бывают любой формы и размера. Их можно обрезать до стандартного размера, но всё отрезанное теряется. Текст ещё упрямее. Отзыв на сайте магазина может состоять из двух слов — «полный отстой!» — а может тянуться на несколько страниц. Книга считает одним из больших преимуществ глубокого обучения то, что его модели сравнительно хорошо справляются с примерами разной длины. Чат-бот, которому вы пишете, — наглядный пример: он принимает ваш вопрос, будь в нём три слова или три абзаца.

Чем больше данных, тем обычно легче задача. На большем числе примеров можно обучать более мощные модели и меньше опираться на заранее сделанные допущения. Переход от довольно небольших наборов данных к очень большим, по словам книги, — одна из главных причин успеха глубокого обучения. Многие из самых впечатляющих его моделей вообще не работают без огромных наборов данных, а некоторые из тех, что работают на маленьких, не лучше старых методов.



Больше данных, правильные данные

Но одного количества мало; данные должны быть правильными. Если они полны ошибок или признаки никак не связаны с тем, что мы хотим предсказать, обучение провалится. Книга приводит старую поговорку: «мусор на входе — мусор на выходе». А в некоторых применениях вред не ограничивается неверным ответом: при прогнозировании преступлений, отборе резюме или решении, кому дать кредит, плохие данные могут навредить людям. Одна частая беда — какой-то группы людей среди примеров просто нет. Представьте систему для распознавания рака кожи, которой на тренировке ни разу не показывали тёмную кожу; она ничего не узнала о людях, на которых будет ошибаться. Другая беда тоньше: в данных есть все, но они несут предассудки прошлого. Если модель обучить на прошлых

решениях компании о найме, а потом использовать для сортировки новых резюме, она может незаметно скопировать и автоматизировать старую несправедливость. Авторы подчёркивают, что это может случиться, хотя никто этого не хотел и даже не заметил.

Запомним

Обучение с учителем — это предсказание неизвестной метки по известным признакам после обучения на примерах, метки которых известны. У каждой задачи машинного обучения четыре составляющие: данные, модель, целевая функция, оценивающая модель, и алгоритм, подстраивающий её параметры. Данные должны стать числами: маленькая цветная фотография двести на двести пикселей — это уже сто двадцать тысяч чисел. С примерами фиксированной длины легко, но тексты и картинки из интернета различаются по размеру, а глубокое обучение сравнительно хорошо справляется с такими данными разной длины. Больше данных — лучше, но только правильных данных: пропущенные группы или прежние предубеждения в примерах становятся ошибками и несправедливостью модели, даже если никто этого не хотел.

Вопросы

1. Что в больничном примере является меткой?

- a. Возраст и пульс пациента
 - b. Список текущих лекарств
 - c. Прожил ли пациент следующие тридцать дней
 - d. Число недавних операций
2. Почему с текстом труднее, чем с фотографиями с одного микроскопа?
- a. Потому что текст нельзя превратить в числа
 - b. Потому что тексты бывают очень разной длины и не укладываются в фиксированное число признаков
 - c. Потому что у текста нет меток
 - d. Потому что отзывы всегда слишком коротки, чтобы на них учиться
3. Детектор рака кожи хорошо работает на проверках, но ошибается на пациентах с тёмной кожей. Что пошло не так, согласно лекции?
- a. У модели было слишком много параметров
 - b. Целевая функция была неправильной оценкой
 - c. Среди тренировочных примеров не было тёмной кожи
 - d. Картинки не обрезали до стандартного размера
4. Почему модель, обученная на прошлых решениях компании о найме, может быть несправедливой, даже если никто этого не хотел?
- a. Потому что данные несут прежние предрассудки, а модель их усваивает и повторяет
 - b. Потому что данных о найме всегда слишком мало
 - c. Потому что резюме — это текст разной длины

d. Потому что модели так и не дали меток

Ответы

1. **с)** Признаки — то, что уже есть в карте; метка — прожил ли каждый прошлый пациент тридцать дней.
2. **б)** Отзыв может состоять из двух слов или тянуться на страницы, так что текст сопротивляется фиксированной длине даже сильнее, чем картинки из интернета.
3. **с)** Одна частая беда — какой-то группы людей среди примеров просто нет, как у системы, которой ни разу не показывали тёмную кожу.
4. **а)** Если данные несут предрассудки прошлого, обученная на них модель может незаметно скопировать и автоматизировать старую несправедливость.

Разобранные примеры

Считаем числа в фотографии Цветная фотография имеет ширину 200 пикселей и высоту 200 пикселей, и каждый пиксель хранится как три числа: для красного, зелёного и синего.

Сколько чисел описывают фотографию?

Сосчитайте пиксели: 200 по ширине на 200 по высоте — это 40 000 пикселей.

У каждого пикселя три числа яркости, по одному на цвет.

Значит, умножаем число пикселей на три: 40 000 на 3 — это 120 000.

У любой фотографии такого размера счёт одинаков, так что её размерность — 120 000.

Ответ: 120 000 чисел

Задачи

1. Цветная фотография поменьше имеет ширину 100 пикселей и высоту 100 пикселей. Сколько чисел её описывают и как это соотносится с фотографией 200 на 200?

Подсказка: Сначала сосчитайте пиксели, потом вспомните, что у каждого пикселя три числа.

Ответ: 100 на 100 — это 10 000 пикселей, на 3 — 30 000 чисел. Если уменьшить вдвое и ширину, и высоту, пикселей станет вчетверо меньше, то есть четверть от 120 000 чисел.

2. Объясните, почему обрезка картинок из интернета до одного стандартного размера лишь отчасти решает проблему данных разной длины.

Подсказка: Что происходит с частями картинки за пределами обрезки?

Ответ: Обрезка делает все примеры одинаковой длины, но всё отрезанное теряется, а там может быть именно то, что важно для предсказания. И с текстом, длина которого меняется ещё сильнее, она вообще не помогает.

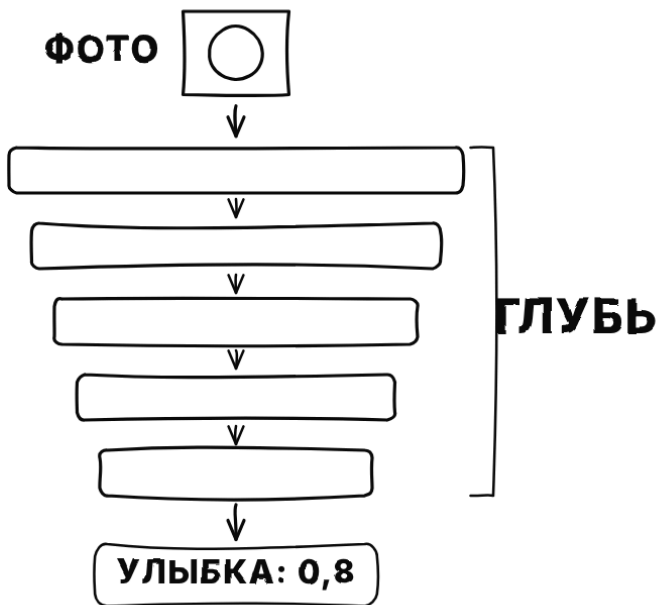
3. Назовите четыре составляющие, которые есть у каждой задачи машинного обучения, и одной фразой скажите, что делает каждая.

Подсказка: Одна — то, на чём учатся, другая превращает это в ответы, третья оценивает, четвёртая подстраивает.

Ответ: Данные — то, на чём учится машина; модель превращает данные в ответы; целевая функция оценивает, насколько хорошо или плохо работает модель; а алгоритм подстраивает параметры модели, чтобы эта оценка стала как можно лучше.

Как оценить машину

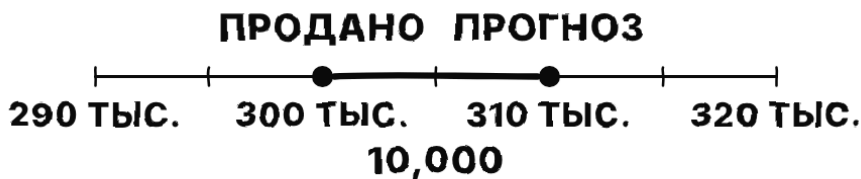
Представьте студентку, которая готовится к выпускному экзамену по прошлогодним пробным вариантам. Через неделю она решает каждый из них на высший балл. А потом на настоящем экзамене, перед вопросами, которых она никогда не видела, она теряется. Она не выучила предмет; она выучила ответы на те самые варианты. Запомните её, потому что обучающиеся машины могут проваливаться точно так же.



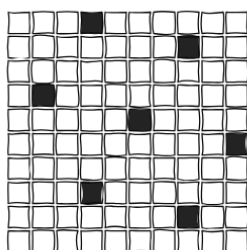
Глубокий — значит много шагов

Сначала — что такое модель? Большая часть машинного обучения берёт данные одного рода и превращает их в предсказание, часто другого рода. Примеры в книге — система, которая принимает фотографии и говорит, насколько человек улыбается, и система, которая принимает показания датчиков и говорит, насколько они обычны или необычны. Модель — это механизм, который выполняет такое превращение, а книгу интересуют статистические модели, ручки которых выставляются по данным. Простые модели годятся для простых задач, но задачи этой книги выходят за пределы возможностей классических методов. Глубокое обучение выделяется своим семейством мощных моделей, построенных как множество преобразований данных, одно за другим, сцепленных сверху вниз. Именно из-за этой стопки шагов его и называют глубоким. По пути книга знакомит и с некоторыми более старыми и простыми методами.

Обучение мы описали как улучшение в задаче с опытом. Но кто решает, что считается лучше? Один человек может считать изменение модели улучшением, а другой — нет. Чтобы строить обучающиеся машины, нужно точное число, которое говорит, насколько модель хороша или плоха. Это число называют целевой функцией. По традиции её устраивают так, что меньше — лучше, и тогда её также называют функцией потерь — мерой того, сколько потеряно. Это лишь соглашение: любую оценку, где больше — лучше, можно превратить в такую, где меньше — лучше, просто поставив перед ней минус.



$$10,000^2 = 100,000,000$$



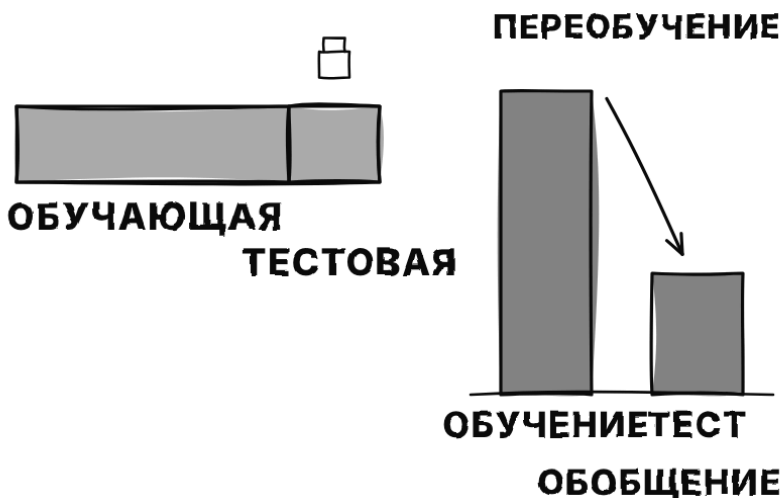
7 ОШИБОК = 7 %

Сколько потеряно

Книга называет две самые распространённые функции потерь. Когда модель предсказывает число, обычная функция потерь — квадратичная ошибка: возьмите разность между предсказанием и истинным значением и умножьте её саму на себя. Допустим, модель предсказывает, что квартира продастся за триста десять тысяч, а она продаётся за триста тысяч. Разность — десять тысяч; десять тысяч на десять тысяч — сто миллионов, это и есть потеря на этом одном примере. Когда модель раскладывает вещи по видам, обычная мера — доля ошибок, то есть доля примеров, на которых она ошибается: семь ошибок на сотню — это доля ошибок семь процентов. Квадратичную ошибку легко оптимизировать, а долю ошибок трудно улучшать напрямую, поэтому на практике часто минимизируют заменяющую оценку

— суррогатную целевую функцию, которая обычно меняется в ту же сторону. Спам-фильтр в итоге оценивают по доле писем, которые он рассортировал неверно, даже если тренируют его на более гладкой заменяющей оценке.

Пока модель тренируют, примеры держат неизменными, а потерю считают зависящей только от ручек. Тренировка означает поиск таких положений ручек, при которых потеря на тренировочных примерах как можно меньше. Но успех на этих примерах не гарантирует успеха на новых. Поэтому данные обычно делят на две стопки. Обучающая выборка служит для настройки ручек; тестовую выборку откладывают, не показывают модели и используют только для проверки результата. Результаты сообщают для обеих.



Обучающая и тестовая выборки

Тут возвращается наша студентка. Оценка на обучающей выборке — это её оценка за пробные варианты: обнадёживает, но ничего не обещает. Если модель по сути зазубрила свои тренировочные примеры, на них она блистает, а на всём новом спотыкается. Умение хорошо справляться с примерами, которых модель никогда не видела, называется обобщением. Хорошие результаты на обучающей выборке при неспособности обобщать называются переобучением. Когда лента рекомендаций, кажется, идеально знает ваши прошлые клики, но не предлагает ничего, что вам на самом деле нужно дальше, — это как раз тот провал, который должна поймать тестовая выборка.

Запомним

Модель превращает данные одного рода в предсказания, а модели глубокого обучения — длинные цепочки преобразований, поэтому их и называют глубокими. Чтобы сказать, стала ли модель лучше, нужно число — целевая функция, обычно устроенная так, что меньше — лучше, и тогда называемая функцией потерь. Для предсказания чисел обычная функция потерь — квадрат отклонения от истины; для раскладывания по видам — доля ошибок, которую часто улучшают через более гладкую заменяющую оценку. Ручки настраивают на обучающей выборке и проверяют на отдельной тестовой, потому что успех на знакомых примерах — не успех на новых. Модель, которая блистает на тренировочных примерах, но проваливается на новых, переобучена, как студентка, зазубрившая пробные варианты.

Вопросы

1. Почему глубокое обучение называют глубоким?
 - a. Потому что его моделям нужны очень большие наборы данных
 - b. Потому что его модели — множество преобразований данных, сцепленных одно за другим
 - c. Потому что оно изучает скрытые свойства данных
 - d. Потому что его функции потерь очень малы

2. Оценку, у которой больше — лучше, превращают в функцию потерь. Как?
 - a. Возводят её в квадрат
 - b. Делят на число примеров
 - c. Ставят перед ней минус
 - d. Измеряют её на тестовой выборке
3. Почему вместо доли ошибок часто используют суррогатную целевую функцию?
 - a. Потому что долю ошибок трудно улучшать напрямую, а более гладкая замена обычно меняется в ту же сторону
 - b. Потому что на тренировочных данных доля ошибок всегда равна нулю
 - c. Потому что доля ошибок годится только для чисел
 - d. Потому что суррогат измеряют на тестовой выборке
4. Модель показывает отличный результат на обучающей выборке, но плохой на тестовой. Как это называется?
 - a. Обобщение
 - b. Суррогатная целевая функция
 - c. Высокая доля ошибок на тренировке
 - d. Переобучение

Ответы

1. b) Модели глубокого обучения построены как множество преобразований данных, сцепленных сверху вниз, и именно из-за этой стопки шагов его называют глубоким.

- 2. c)** Любую оценку, где больше — лучше, можно превратить в такую, где меньше — лучше, просто поставив перед ней минус.
- 3. a)** Долю ошибок трудно улучшать напрямую, поэтому часто минимизируют заменяющую оценку, которая обычно меняется в ту же сторону.
- 4. d)** Хорошие результаты на обучающей выборке при неспособности обобщать на невиданные данные называются переобучением.

Разобранные примеры

Одна квадратичная ошибка Модель предсказывает, что квартира продается за 310 000, а на самом деле она продаётся за 300 000. Какова квадратичная ошибка на этом примере?

Найдите разность между предсказанием и истиной: 310 000 минус 300 000 — это 10 000.

Возведите её в квадрат, то есть умножьте саму на себя: 10 000 на 10 000.

Получается 100 000 000.

Потеря велика, потому что возведение в квадрат делает большие промахи гораздо весомее маленьких.

Ответ: 100 000 000 (сто миллионов)

Задачи

1. Модель, раскладывающая фотографии на кошек и собак, ошибается на 14 из 200 тестовых фото. Какова её доля ошибок?

Подсказка: Доля ошибок — это доля примеров, на которых модель ошибается.

Ответ: 14 разделить на 200 — это 0,07, то есть доля ошибок семь процентов.

2. Модель предсказывает 8, когда истина 5, а на другом примере 2, когда истина 3. Какой пример вносит больший вклад в квадратичную функцию потерь и какой вклад у каждого?

Подсказка: Возьмите каждую разность и умножьте саму на себя.

Ответ: Первый промахивается на 3, так что его квадратичная ошибка — 9; второй промахивается на 1, так что его квадратичная ошибка — 1. Первый вносит в девять раз больше.

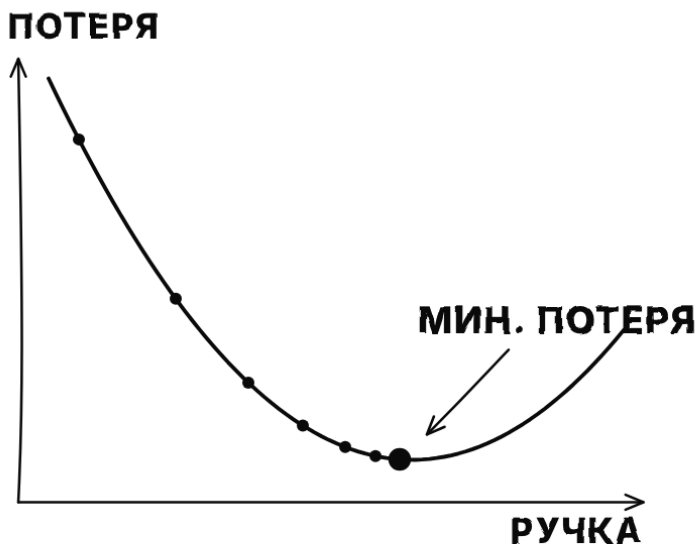
3. Объясните на примере студентки и пробных вариантов, зачем держать тестовую выборку отдельно от обучающей.

Подсказка: Что говорит идеальная оценка за зазубренные пробные варианты о настоящем экзамене?

Ответ: Модель, настроенная на обучающей выборке, может её просто зазубрить, как студентка зазубрила пробные варианты. Только примеры, которых она никогда не видела, — тестовая выборка — показывают, научилась ли она чему-то переносимому, то есть обобщает ли она.

Нащупывая путь вниз

Встаньте на склоне холма в густом тумане, желая добраться до дна долины. Вы не видите дальше одного шага, поэтому не видите, где долина. Но вы чувствуете склон под ногами: в одну сторону земля уходит вниз, в другую поднимается. Сделайте маленький шаг туда, где она уходит вниз, снова прощупайте и снова шагните. Рано или поздно вы окажетесь на дне.

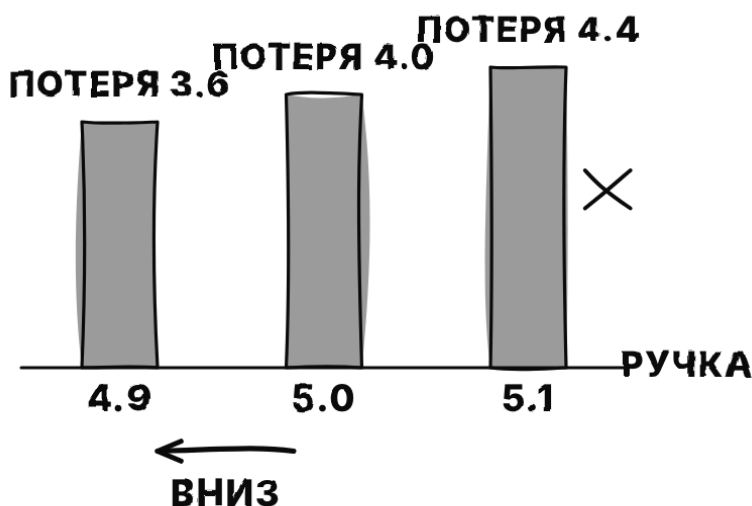


Спуск по склону потерь

В этом и состоит идея метода, который книга теперь называет, — последней из четырёх составляющих. У нас есть данные, модель с

ручками и функция потерь, которая говорит, насколько плохо она работает; теперь нужен способ найти такие положения ручек, при которых потеря как можно меньше. Большинство методов, применяемых в глубоком обучении, построены на градиентном спуске. На каждом шаге он спрашивает для каждой ручки по очереди, как изменилась бы потеря на обучающей выборке, если бы эту ручку повернули совсем чуть-чуть. Потом он немного поворачивает каждую ручку в ту сторону, где потеря уменьшается. Потеря — это высота склона, а положения ручек — то место, где вы стоите.

Вот один шаг в числах для модели с единственной ручкой. Пусть ручка стоит на пяти, и потеря там равна четырём. Поверните её чуть-чуть вверх, до пяти целых одной десятой, и потеря вырастет примерно до четырёх целых четырёх десятых; значит, вверх — не туда. Поверните её вместо этого чуть-чуть вниз, до четырёх целых девяти десятых, и потеря упадёт примерно до трёх целых шести десятых. Значит, метод сдвигает ручку вниз, а затем снова проверяет наклон со своего нового места. У настоящей модели глубокого обучения миллионы ручек, но на каждом шаге со всеми ними поступают одинаково.

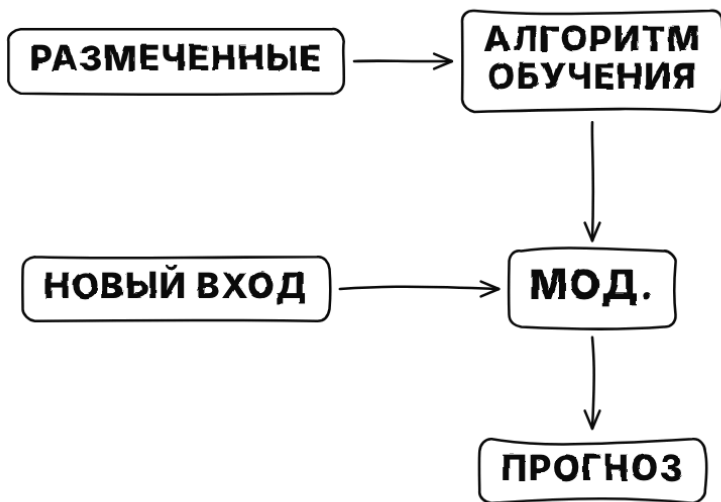


Один шаг в числах

Разобравшись с четырьмя составляющими, книга переходит к видам задач, за которые может взяться машинное обучение, начиная с самого распространённого — обучения с учителем. Здесь нам даны примеры, признаки которых известны и метки которых тоже известны, и задача — построить модель, которая предсказывает метку по признакам. Каждая пара из признаков и метки — это один пример. Оно называется обучением с учителем, потому что мы, учителя, даём модели размеченные примеры, по которым она выбирает положения своих ручек. На языке теории вероятностей нам нужно знать, насколько вероятна каждая метка при тех признаках, которые мы видим.

Обучение с учителем — лишь один из нескольких подходов, но на него приходится большая часть успешных применений машинного обучения в промышленности. Многие важные задачи укладываются в его простую схему: угадать что-то неизвестное по имеющимся данным. Есть ли рак, судя по снимку с медицинского сканера? Каков правильный французский перевод этого английского предложения? Какой будет цена акции в следующем месяце, судя по финансовым отчётам этого месяца? Ваш спам-фильтр следует той же схеме: по письму предсказать «спам» или «не спам». Схема проста, но форм у неё огромное множество; например, для последовательностей любой длины нужны другие модели, чем для входов фиксированного размера.

Книга описывает процесс и изображает его картинкой: размеченные примеры поступают в метод обучения, из него выходит модель, а новые входы поступают в эту модель и выходят как предсказания. Сначала соберите большое собрание примеров с известными признаками, выберите из них случайно несколько и получите для каждого истинную метку. Иногда метки уже записаны, например умер ли пациент в течение следующего года; иногда людям приходится платить за ручную разметку данных, например за раскладывание картинок по категориям. Эти примеры с метками и составляют обучающую выборку. Обучающая выборка поступает в алгоритм обучения, который сам является функцией, результатом которой становится другая функция — обученная модель. Наконец, в модель подают входы, которых она никогда не видела, и её выходы принимают за предсказания их меток.



От размеченных примеров к предсказаниям

Запомним

Градиентный спуск — это то, как настраивают ручки: для каждой ручки проверить, увеличивает или уменьшает потерю крошечный поворот, потом немного повернуть её в сторону уменьшения и повторить. Это похоже на поиск долины в тумане, когда нащупываешь склон под ногами. Обучение с учителем — это обучение предсказывать метки по признакам на примерах, метки которых даны модели. Большинство успешных применений в промышленности — такого рода, от медицинских снимков до перевода. Процесс идёт от размеченных примеров через алгоритм обучения к обученной модели, которая делает предсказания на новых входах.

Вопросы

1. Что в картинке градиентного спуска со склоном в тумане соответствует высоте земли?
 - a. Число ручек
 - b. Потеря на обучающей выборке
 - c. Размер набора данных
 - d. Число сделанных шагов
2. Крошечный поворот ручки вверх увеличивает потерю. Что градиентный спуск делает дальше?
 - a. Поворачивает ручку ещё выше

- b. Навсегда оставляет ручку в покое
 - c. Немного поворачивает ручку вниз, а потом снова проверяет наклон
 - d. Возвращает ручку к случайному значению
3. Почему такое обучение называют обучением с учителем?
- a. Потому что человек наблюдает за моделью, пока она работает
 - b. Потому что мы даём модели размеченные примеры, по которым она выбирает положения ручек
 - c. Потому что ошибки модели потом исправляют вручную
 - d. Потому что модель работает только под контролем учителя
4. Что выдаёт в конце алгоритм обучения?
- a. Обученную модель — саму по себе функцию, которая превращает новые входы в предсказания
 - b. Список меток для обучающей выборки
 - c. Очищенную копию набора данных
 - d. Одно число — потерю

Ответы

1. **b)** Потеря — это высота склона, а положения ручек — то место, где вы стоите.
2. **c)** Если вверх — не туда, метод сдвигает ручку вниз, а затем снова проверяет наклон со своего нового места.
3. **b)** Оно называется обучением с учителем, потому что мы, учителя, даём модели размеченные примеры, по которым она выбирает положения своих ручек.

4. а) Алгоритм обучения — это функция, результатом которой становится другая функция, обученная модель.

Разобранные примеры

Один шаг вниз У модели одна ручка, выставленная на 5, и потеря там равна 4. Если повернуть ручку до 5,1, потеря станет около 4,4; если до 4,9 — около 3,6. В какую сторону градиентный спуск сдвинет ручку?

Сравните каждый крошечный поворот с текущей потерей, равной 4.

Вверх до 5,1 — потеря растёт примерно до 4,4, значит, вверх — это подъём.

Вниз до 4,9 — потеря падает примерно до 3,6, значит, вниз — это спуск.

Градиентный спуск немного поворачивает ручку вниз, а потом снова проверяет наклон.

Ответ: вниз, к меньшим значениям ручки

Задачи

1. Для каждой задачи назовите признаки и метку: найти рак на снимке; перевести английское предложение на французский; предсказать цену акции в следующем месяце по финансовым отчётам этого месяца.

Подсказка: Признаки — то, что показывают модели; метка — то, что она должна предсказать.

Ответ: Снимок: признаки — изображение снимка, метка — рак или не рак. Перевод: признаки — английское предложение, метка — его французский перевод. Акции: признаки — финансовые отчёты этого месяца, метка — цена акции в следующем месяце.

2. Расставьте по порядку процесс обучения с учителем: подать модели новые входы; получить истинную метку для каждого выбранного примера; собрать примеры и случайно выбрать несколько; подать обучающую выборку в алгоритм обучения.

Подсказка: Метки должны появиться прежде, чем можно чему-то научиться.

Ответ: Собрать примеры и случайно выбрать несколько; получить для каждого истинную метку; подать размеченную обучающую выборку в алгоритм обучения, который выдаёт обученную модель; затем подать модели новые входы и использовать её выходы как предсказания.

Сколько

Сантехник три часа прочищает ваши засорившиеся трубы и выставляет счёт на триста пятьдесят долларов. Ваш друг нанимает того же сантехника на два часа и платит двести пятьдесят. Теперь сосед спрашивает, сколько будет стоить работа на четыре часа. Прежде чем читать дальше, попробуйте ответить сами.

3 ч → 350 \$ · 2 ч → 250 \$ · 4 ч → ?

ПЛОЩ. СПАЛЬНИВАННЫЕ ПЕШКОМ

600	1	1	60
3000	4	3	10



ВЕКТОР ПРИЗНАКОВ

Счета и дома как строки

Самый простой для понимания вид обучения с учителем называется регрессией. Представьте таблицу продаж домов: по

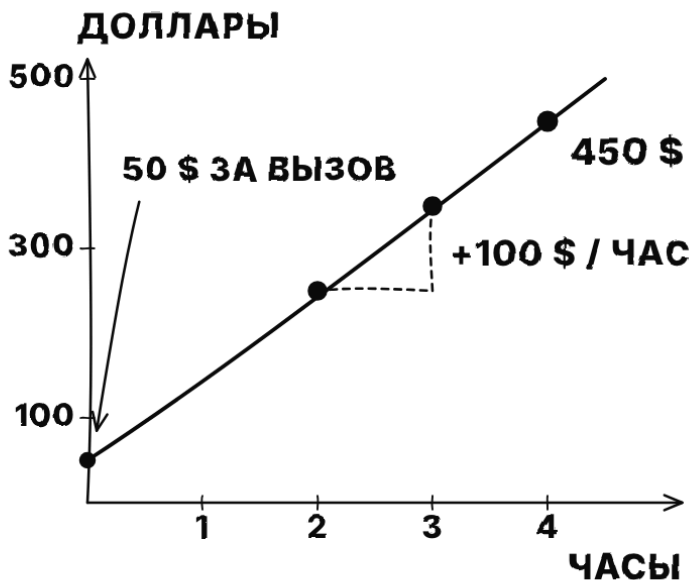
строке на каждый дом и по столбцу на каждый факт о нём — площадь, число спален, число ванных и сколько минут пешком до центра города. Каждый дом — один пример, а его строка фактов — вектор признаков, фиксированный список чисел в фиксированном порядке. Книга шутит, что в Нью-Йорке или Сан-Франциско, если только вы не возглавляете одну из гигантских технологических компаний, ваше жильё может выглядеть так: шестьсот квадратных футов, одна спальня, одна ванная, шестьдесят минут пешком. В Питтсбурге оно может выглядеть так: три тысячи квадратных футов, четыре спальни, три ванные, десять минут пешком. Большинству классических методов обучения нужно, чтобы каждый пример приходил в виде такого списка, всегда одной и той же длины.

Регрессией задачу делают не признаки, а вид ответа. Если вы ищете дом, вам может понадобиться оценить справедливую цену по таким фактам, и модель могла бы научиться этому по прошлым объявлениям, где метками служат цены, за которые дома на самом деле продались. Когда метка может быть любым числом, пусть и в каком-то диапазоне, задача — регрессия, и цель — модель, догадки которой близки к истинным значениям. Полезное правило: вопрос, который начинается со «сколько?», — скорее всего, регрессия. Сколько часов займёт эта операция? Сколько дождя выпадет в этом городе за следующие шесть часов? Сколько этот пациент пробудет в больнице? Угадать, сколько звёзд вы поставили бы фильму, — тоже регрессия, и книга напоминает, что в 2009 году, сделав это хорошо, можно было выиграть приз Netflix в миллион долларов.

Вернёмся к сантехнику. Вы, вероятно, предположили две вещи: больше часов стоит дороже, а счёт — это фиксированная плата за

приход плюс столько-то за каждый час. Два счёта различаются на один час и на сто долларов, значит, каждый час стоит сто долларов. Три часа по сто — это триста, а счёт был на триста пятьдесят, значит, за приход берут пятьдесят долларов.

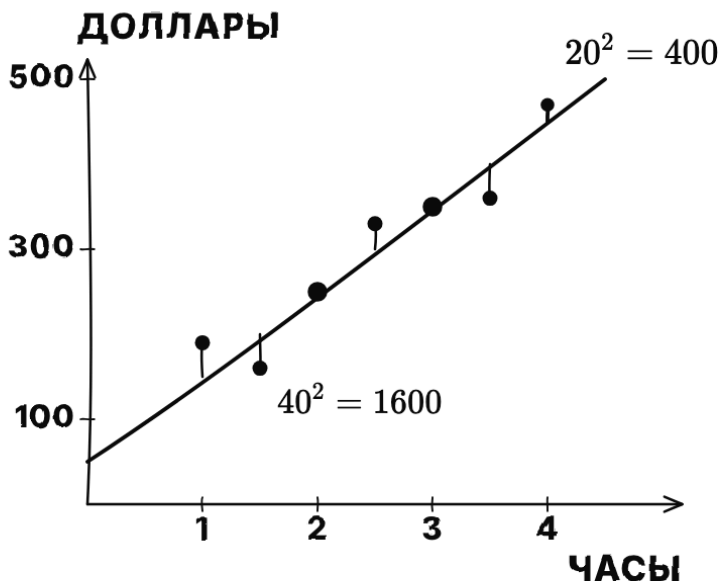
Проверим на счёте друга: пятьдесят плюс два часа по сто — это двести пятьдесят, ровно столько. Значит, работа на четыре часа стоит пятьдесят плюс четыреста, то есть четыреста пятьдесят долларов. Если вы это проследили, вы уже понимаете идею линейной регрессии: ответ — фиксированная сумма плюс определённая сумма за каждую единицу каждого признака.



Плата за вызов плюс ставка

Здесь два параметра, почасовая ставка и плата за вызов, точно совпали со счётами. Настоящие данные редко бывают такими

аккуратными, потому что часть разброса идёт от того, чего нет в таблице: упрямый засор, долгая дорога, пробки. Тогда никакое положение ручек не подходит ко всем примерам, и вместо этого мы ищем такое, при котором предсказания как можно ближе к тому, что наблюдалось на самом деле. Книга в основном измеряет близость квадратичной ошибкой — промахом, умноженным сам на себя. Допустим, третья работа, на четыре часа, обошлась в четыреста семьдесят: правило предсказывает четыреста пятьдесят, промах — двадцать долларов, а квадратичная ошибка — двадцать на двадцать, четыреста. Вдвое больший промах весил бы вчетверо больше, так что возведение в квадрат делает большие промахи гораздо весомее маленьких. Авторы замечают, что такой выбор функции потерь естествен, когда случайные отклонения ведут себя как самый распространённый вид случайного шума — колоколообразный.



Прوماхи в квадрате

Запомним

Регрессия — обучение с учителем, где ответ — число; вопросы «сколько?» обычно относятся к регрессии. Каждый пример — вектор признаков, список фактов фиксированной длины, вроде площади, спален, ванных и времени пешком для дома. Два счёта сантехника раскрывают плату за вызов в пятьдесят долларов и ставку сто в час — это и есть идея линейной регрессии. Когда реальные данные не укладываются точно, мы выбираем параметры, при которых предсказания ближе всего к истине, обычно делая квадратичную ошибку как можно меньше.

Вопросы

1. Что делает задачу регрессией, а не каким-то другим видом?
 - a. То, что признаки — числа
 - b. То, что метка, нужный ответ, — число
 - c. То, что данные даны в таблице
 - d. То, что признаков всего два
2. Почему модель, обученная на настоящих счетах сантехников, обычно не может точно совпасть с каждым счётом?
 - a. Потому что счета меняет и то, чего нет в таблице, например упрямый засор или пробки
 - b. Потому что признаков слишком много
 - c. Потому что квадратичную ошибку нельзя вычислить для денег
 - d. Потому что по закону у каждого счёта своя плата за вызов
3. Одно предсказание промахивается на 10, другое на 20. Как соотносятся их квадратичные ошибки?
 - a. Второе весит вдвое больше
 - b. Они весят одинаково
 - c. Второе весит вчетверо больше
 - d. Второе весит вдвое меньше

Ответы

1. **б)** Регрессией задачу делают не признаки, а вид ответа: когда метка может быть любым числом, это регрессия.
2. **а)** Часть разброса идёт от того, чего нет в таблице, поэтому никакое положение ручек не подходит ко всем примерам.
3. **с)** Вдвое больший промах весит вчетверо больше, потому что промах умножается сам на себя.

Разобранные примеры

Расценки сантехника Сантехник взял 350 долларов за работу на 3 часа и 250 долларов за работу на 2 часа. Если считать, что есть фиксированная плата за приход плюс ставка за час, найдите плату, ставку и стоимость работы на 4 часа.

Счета различаются на один час и на 100 долларов, значит, ставка — 100 долларов в час.

Три часа — это 300 долларов из 350, значит, фиксированная плата — 50 долларов.

Проверка: 50 плюс 2 часа по 100 — это 250, как и в счёте.

Работа на 4 часа стоит 50 плюс 400.

Ответ: фиксированная плата 50 долларов и 100 долларов в час; работа на 4 часа стоит 450 долларов

Задачи

1. Третья работа на 4 часа обошлась в 470 долларов, а правило «50 долларов плюс 100 в час» предсказывает 450. Четвёртая работа, на 1 час, обошлась в 140 долларов. Какова квадратичная ошибка правила на каждой работе?

Подсказка: Сначала найдите, что правило предсказывает для одного часа, потом возьмите каждый промах и умножьте сам на себя.

Ответ: Для работы на 4 часа промах — 20 долларов, так что квадратичная ошибка — 400. Для работы на 1 час правило предсказывает 150, промах — 10 долларов, и квадратичная ошибка — 100.

2. Запишите вектор признаков дома площадью 1200 квадратных футов, с 2 спальнями, 1 ванной и 25 минутами пешком до центра, в порядке, принятом в книге, и скажите, какую метку для него предсказала бы модель регрессии.

Подсказка: Порядок в книге: площадь, спальни, ванны, минуты пешком.

Ответ: Вектор признаков — 1200, 2, 1, 25. Модель регрессии, обученная на прошлых продажах, предсказала бы число — цену дома.

Что из этого и насколько уверенно

Представьте красивый бледный гриб у себя на заднем дворе. Вы фотографируете его приложением, обученным распознавать ядовитые грибы, и приложение говорит, что на восемьдесят процентов уверено: гриб безвреден. Значит, скорее всего, он безопасен. Приготовили бы вы его на ужин? Запомните свой ответ; книга вернётся именно к этому грибу.



Какой это класс?

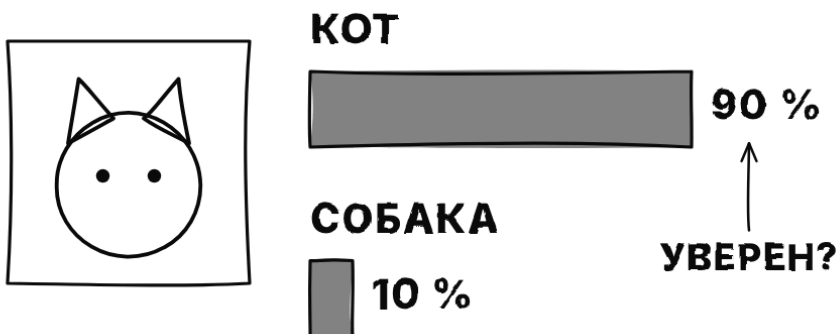
Регрессия отвечает на вопрос «сколько?», но немало вопросов другого рода. Возьмём банк, который хочет, чтобы его мобильное

приложение читало чек по фотографии. Допустим, картинка уже разрезана на маленькие кусочки, по одному на каждый рукописный знак. Остаётся для каждого кусочка решить, какой знак из известного набора на нём изображён. Такие вопросы «что из этого?» называются классификацией. Им нужны свои инструменты, хотя многие идеи переносятся из регрессии.

При классификации модель смотрит на признаки, например на пиксели изображения, и говорит, к какой категории, иначе классу, относится пример, из фиксированного набора вариантов. Для рукописных цифр классов десять, от нуля до девяти. Когда классов только два, например кошка и собака, это называется бинарной классификацией. Ваш спам-фильтр — бинарная классификация: каждое письмо попадает в один из двух классов, спам или не спам. Модель регрессии, регрессор, выдаёт число; модель классификации, классификатор, выдаёт класс. Когда классов больше двух, как при чтении рукописного текста со всеми десятью цифрами и всеми буквами, это называется многоклассовой классификацией.

Модель, которая может дать только твёрдый ответ, «кошка» или «собака», оказывается трудно обучать, по причинам, которые книга объяснит позже. Поэтому её обычно заставляют вместо этого выдавать вероятность для каждого класса. Увидев фото, классификатор может оценить вероятность кошки в ноль целых девять десятых, что можно понимать как «на девяносто процентов уверен, что это кошка», а значит, десять процентов — собака. Величина вероятности выбранного класса говорит нам, насколько модель не уверена, хотя книга замечает, что это не единственная мера неопределённости. Регрессию тренировали, делая квадратичную ошибку маленькой; у классификации своя

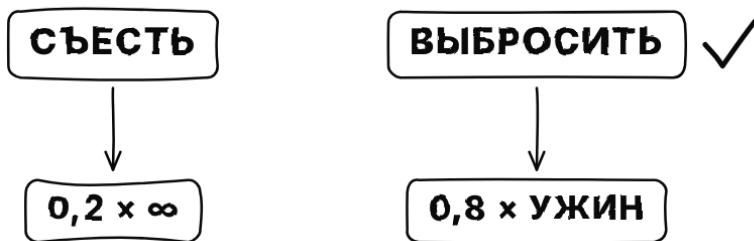
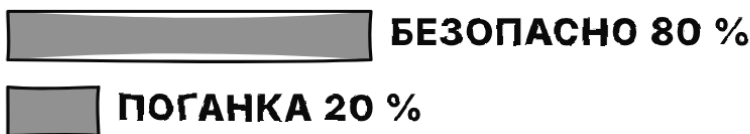
обычная функция потерь, называемая перекрёстной энтропией, — название, которое книга обещает объяснить, когда дойдёт до теории информации.



Вероятность для каждого класса

Теперь гриб. Классификатор ядовитых грибов даёт вероятность ноль целых две десятых, что это бледная поганка, то есть на восемьдесят процентов уверен, что нет. И всё же съест его только безумец, потому что верное удовольствие от хорошего ужина не стоит риска один к пяти умереть. Чтобы решить, книга взвешивает каждое действие: для каждого возможного исхода умножить его вероятность на то, сколько вреда он причинит, и сложить. Если вы съедите гриб, с вероятностью двадцать процентов вред будет так велик, что его можно назвать

бесконечным, а с вероятностью восемьдесят процентов вреда не будет; двадцать процентов от бесконечности — всё равно бесконечность. Если вы его выбросите, с вероятностью двадцать процентов вы ничего не теряете, ведь он и так был ядовит, а с вероятностью восемьдесят процентов теряете один ужин, который считаем за единицу: ноль целых восемь десятых на единицу даёт потерю всего ноль целых восемь десятых. Ноль целых восемь десятых против бесконечности: выбросить — намного лучше. Действовать нужно не обязательно по самому вероятному классу. И осторожность была верной: гриб на картинке в книге действительно бледная поганка.

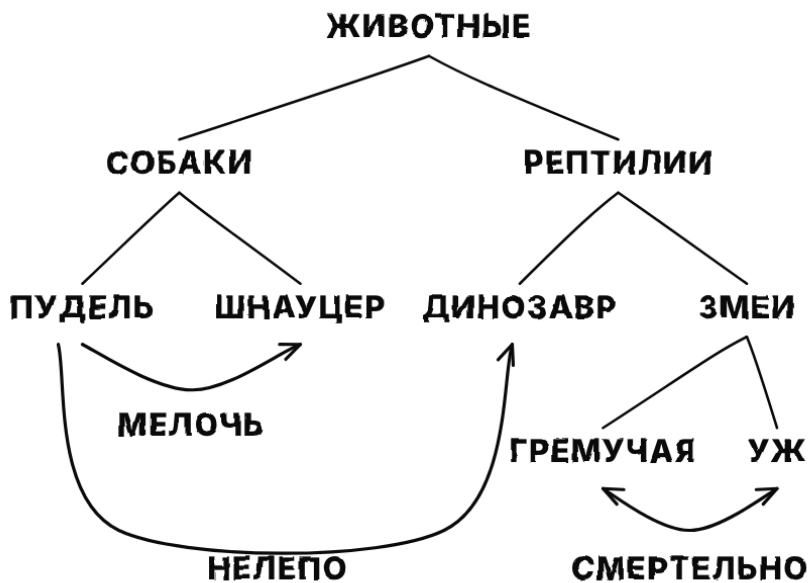


Взвесьте каждое действие по вреду

Опыт

В следующий раз, когда приложение покажет вам процент уверенности, спросите, чего вам будет стоить каждый исход. Девяноста процентов «вам понравится эта песня» вполне хватит, чтобы нажать «играть»; девяноста процентов «это лекарство сочетается с другими вашими таблетками» самих по себе далеко не достаточно. Одна и та же вероятность может требовать противоположных действий, когда цена ошибки разная.

Классификация может быть ещё запутаннее. Иногда классы образуют родословное древо, примерно как натуралист Линней раскладывал животных по вложенным группам. Тогда не все ошибки равны: если модель должна ошибиться, лучше попасть на близкого родственника, чем на дальнего. Это называется иерархической классификацией. Принять пуделя за шнауцера — мелкая оплошность; принять пуделя за динозавра заслуживает огромного штрафа. Но какое древо имеет значение, зависит от того, для чего модель. Гремучие змеи и безобидные подвязочные ужи стоят рядом в родословном древе змей, но принять гремучую змею за ужа может стоить жизни.



Одни ошибки хуже других

Запомним

Классификация отвечает на вопрос «что из этого?», относя каждый пример к одному классу из фиксированного набора: бинарная — с двумя классами, многоклассовая — с большим числом. Классификаторы обычно дают вероятность для каждого класса, которая заодно говорит, насколько они уверены, и их обычно тренируют с функцией потерь, называемой перекрёстной энтропией. Действовать разумно — значит взвешивать каждый выбор по вероятности каждого исхода, умноженной на его вред, а не просто брать самый вероятный класс. Для гриба съесть его — бесконечный ожидаемый вред, а выбросить — лишь ноль целых восемь десятых ужина. Когда классы образуют древо, одни ошибки хуже других, и какие именно — зависит от применения, как с гремучей змеей и ужом.

Вопросы

1. Классификатор говорит, что на фото кошка, с вероятностью 0,9. Как это понимать?
 - a. Девять из десяти фото, которые он видит, — кошки
 - b. Он на девяносто процентов уверен, что на этом фото кошка
 - c. Фото на девяносто процентов кошка и на десять процентов собака

- d. Он был прав на девяносто процентах своей обучающей выборки
2. Почему разумно выбросить гриб, который, по мнению приложения, безопасен с вероятностью восемьдесят процентов?
- a. Потому что восемьдесят процентов значит, что приложение сломано
 - b. Потому что взвешенный вред от того, чтобы его съесть, — небольшой шанс смерти — намного перевешивает взвешенный вред от потерянного ужина
 - c. Потому что самый вероятный класс всегда неверен
 - d. Потому что грибы нельзя классифицировать
3. Какая ошибка при иерархической классификации заслуживает наименьшего штрафа?
- a. Принять пуделя за динозавра
 - b. Принять пуделя за шнауцера
 - c. Принять собаку за рыбу
 - d. Все ошибки заслуживают одинакового штрафа
4. Почему спутать гремучую змею с подвязочным ужом считается серьёзной ошибкой, хотя они близкие родственники?
- a. Потому что цена ошибки зависит от того, для чего используют модель, а здесь она может убить
 - b. Потому что они принадлежат к разным семействам животных

с. Потому что модель обучали на слишком малом числе змей

д. Потому что змеи — всегда многоклассовая задача

Ответы

1. **б)** Вероятность кошки ноль целых девять десятых можно понимать как «на девяносто процентов уверен, что это кошка».

2. **б)** Если взвесить каждое действие по вероятности каждого исхода, умноженной на его вред, съесть гриб — бесконечный ожидаемый вред, а выбросить — лишь ноль целых восемь десятых.

3. **б)** Если модель должна ошибиться, лучше попасть на близкого родственника: принять пуделя за шнауцера — мелкая оплошность.

4. **а)** Какое древо имеет значение, зависит от того, для чего модель; принять гремучую змею за ужа может быть смертельно.

Разобранные примеры

Съесть или не съесть Классификатор даёт вероятность 0,2, что гриб — бледная поганка, и 0,8, что нет. Съесть бледную поганку — бесконечный вред; съесть хороший гриб ничего не стоит.

Выбросить бледную поганку ничего не стоит; выбросить хороший гриб стоит одного ужина, который считаем за 1. У какого действия ожидаемый вред меньше?

Для каждого действия умножьте вред каждого исхода на его вероятность и сложите.

Съесть: 0,2 на бесконечный вред плюс 0,8 на 0; уже первое слагаемое бесконечно.

Выбросить: 0,2 на 0 плюс 0,8 на 1, то есть 0,8.

0,8 намного меньше бесконечного вреда, так что гриб нужно выбросить, хотя «безопасно» — более вероятный класс.

Ответ: выбросить: ожидаемый вред 0,8 против бесконечного, если съесть

Задачи

1. Другое грибное приложение даёт вероятность бледной поганки всего 0,01. При тех же величинах вреда каков ожидаемый вред, если съесть, и если выбросить? О чём это говорит?

Подсказка: Любая положительная вероятность, умноженная на бесконечный вред, всё равно бесконечна.

Ответ: Съесть: 0,01 на бесконечный вред — всё равно бесконечность. Выбросить: 0,99 на 1 — это 0,99. Выбросить по-прежнему лучше: когда один исход катастрофичен, даже малый шанс на него перевешивает верную небольшую потерю.

2. Скажите, что из этого бинарная классификация, многоклассовая или регрессия: прочитатель, какая из цифр от 0 до 9 на кусочке чека; решить, кошка или собака; предсказать цену дома; прочитатель любую букву или цифру.

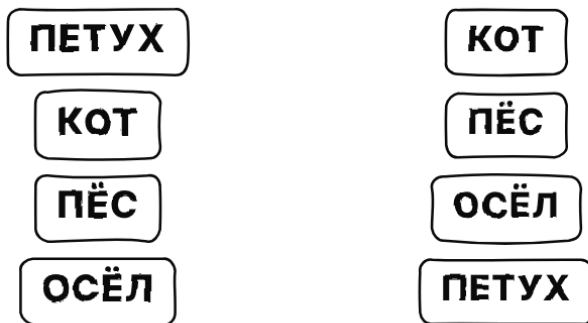
Подсказка: Посчитайте классы или спросите, является ли ответ числом.

Ответ: Цифры от 0 до 9 и буквы вместе с цифрами — многоклассовая классификация; кошка или собака — бинарная классификация; цена дома — число, значит, это регрессия.

Много меток и лучшее первым

Представьте бременских музыкантов из старой немецкой сказки: осёл, на его спине стоит собака, на собаке кошка, а на самом верху петух, за ними деревья. А теперь покажите эту картинку классификатору, обученному отвечать «кошка или собака?». Сегодня такой классификатор построить легко, и он может быть очень точным. Но здесь он обязательно ошибётся, потому что должен выбрать ровно один ответ, а верный ответ — «и то и другое, а ещё осёл и петух».

МНОГО МЕТОК



Одна картинка, несколько меток

Беда не в модели, а в вопросе. Если такие картинки ожидаются, выбор одного класса из многих — неверная постановка задачи. Модели нужно позволить сказать: кошка, и собака, и осёл, и петух. Предсказание классов, которые могут быть верны одновременно, называется классификацией с несколькими метками.

Вот один способ увидеть, насколько этот вопрос больше. С четырьмя животными и правилом «выбрать ровно одно» возможных ответов четыре. Если каждое животное может независимо быть или не быть, у каждого две возможности, так что ответов два на два на два на два, то есть шестнадцать, от ни одного животного до всех четырёх. Каждая новая метка снова удваивает число.

ВЫБРАТЬ ОДНО

КОТ

ПЁС

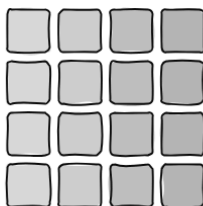
ОСЁЛ

ПЕТУХ

= 4

ЛЮБЫЕ ИЗ НИХ

$$2 \times 2 \times 2 \times 2 = 16$$



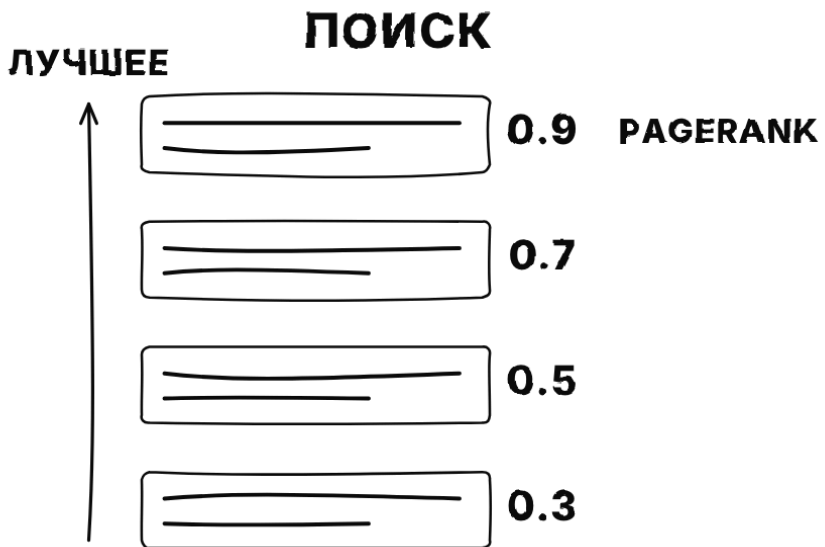
= 16

4 ответа или 16

Классификация с несколькими метками — естественное описание автоматической расстановки тегов. Вспомните теги у записей в технологическом блоге: «машинное обучение», «технологии», «гаджеты», «языки программирования», «облачные вычисления». Типичная статья может нести от пяти до десяти таких тегов. И теги ходят компаниями: записи об облачных вычислениях обычно упоминают облачный сервис Amazon, записи о машинном обучении обычно упоминают графические чипы, на которых оно работает. Хештеги, которые приложение предлагает, когда вы публикуете фото, — догадка именно такого рода: несколько тегов сразу, а не один.

Иногда набор тегов огромен. В Национальной медицинской библиотеке США обученные сотрудники помечают каждую статью, попадающую в её большой медицинский указатель, терминами из списка примерно в двадцать восемь тысяч предметных рубрик. Хорошие теги позволяют исследователям найти всё, что когда-либо написано по какому-то вопросу, но работа идёт медленно, и статья обычно ждёт тегов около года. Машинное обучение может выдавать предварительные теги, пока человек не просмотрит каждую статью, и уже несколько лет проводятся соревнования именно об этом.

Поиск снова переворачивает вопрос. При поиске в интернете подходящих страниц обычно много; настоящий вопрос — какие из них показать первыми. Один способ — дать каждой странице оценку и выдать страницы с самыми высокими оценками. PageRank, первоначальный секретный ингредиент поисковой системы Google, был ранней оценкой такого рода. Как ни странно, PageRank вовсе не зависел от слов запроса: простой фильтр сначала отбирал подходящие страницы, а PageRank затем ставил наверх более авторитетные. Сегодня поисковые системы используют машинное обучение и модели поведения людей, чтобы вычислять оценки, которые от запроса зависят. Этой одной задаче посвящены целые научные конференции.



Поиск ранжирует по оценке

Запомним

Классификатор, вынужденный выбирать один класс, проваливается на картинке, где вместе осёл, собака, кошка и петух. Классификация с несколькими метками позволяет нескольким классам быть верными одновременно, и с четырьмя метками возможных ответов уже шестнадцать. Это естественная форма автоматической расстановки тегов, от записей в блогах до двадцати восьми тысяч медицинских рубрик большой библиотеки. Поиск — это ранжирование: дать подходящим объектам оценку и показать лучшие первыми, как когда-то делал PageRank, даже не глядя на слова запроса.

Вопросы

1. Почему даже очень точный классификатор «кошка или собака» обязательно ошибётся на бременской картинке?
 - a. Потому что животные нарисованы, а не сфотографированы
 - b. Потому что он должен выбрать ровно один класс, а верный ответ — сразу несколько
 - c. Потому что на заднем плане деревья
 - d. Потому что ослов не было в его обучающей выборке
2. Если возможных меток пять и каждая может быть или не быть, сколько всего возможных ответов?

- a. 5
 - b. 10
 - c. 25
 - d. 32
3. Что поисковая система с PageRank делала со словами, которые вы набрали?
- a. Простой фильтр отбирал по ним подходящие страницы, а PageRank затем упорядочивал эти страницы, не глядя на слова
 - b. PageRank оценивал каждую страницу по тому, как часто на ней встречаются ваши слова
 - c. Она их не учитывала и показывала всем одни и те же страницы
 - d. Она использовала их, чтобы присвоить каждой странице несколько меток

Ответы

1. **b)** Он должен выбрать ровно один ответ, а верный ответ — и то и другое, а ещё осёл и петух: беда в вопросе, а не в модели.
2. **d)** Каждая новая метка удваивает число: четыре метки дают шестнадцать ответов, значит, пять дают тридцать два.
3. **a)** PageRank не зависел от слов запроса: простой фильтр отбирал подходящие страницы, а PageRank ставил наверх более авторитетные.

Разобранные примеры

Считаем возможные ответы На картинке может быть любое из четырёх животных: осёл, собака, кошка и петух. Сколько

возможных ответов, если модель должна выбрать ровно одно животное, и сколько, если каждое животное может быть или не быть?

Выбор ровно одного из четырёх животных даёт 4 возможных ответа.

Если каждое животное может быть или не быть, у каждого 2 возможности.

Возможности для четырёх животных перемножаются: 2 на 2 на 2 на 2.

Получается 16, от ни одного животного до всех четырёх вместе.

Ответ: 4 ответа при одном классе; 16 при классификации с несколькими метками

Задачи

1. В блоге 10 возможных тегов, и любая запись может нести любые из них. Сколько разных наборов тегов может быть у записи?

Подсказка: Каждый тег либо есть, либо нет, и каждый новый тег удваивает число.

Ответ: 2, умноженное само на себя 10 раз, то есть 1024, считая запись вообще без тегов.

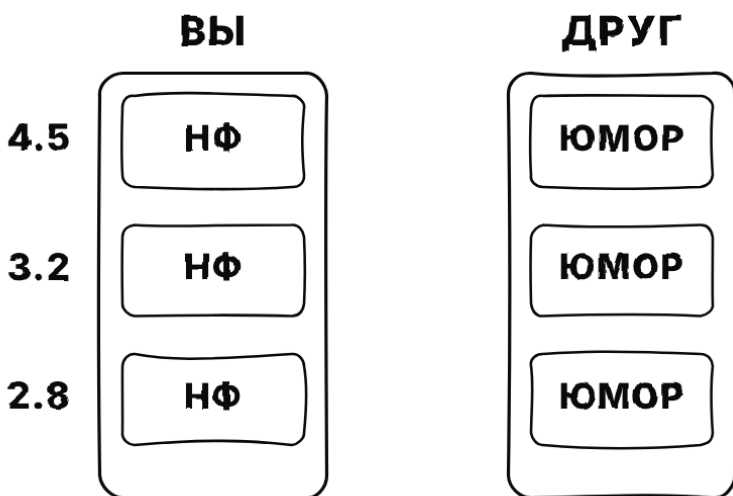
2. Объясните, почему поисковой системе нужно ранжирование, а не просто «да» или «нет» для каждой страницы.

Подсказка: Сколько подходящих страниц находит типичный поиск?

Ответ: Поиск обычно находит много подходящих страниц, так что знать, подходит ли каждая, недостаточно; система должна решить, какие показать первыми. Она даёт каждой странице оценку и показывает наверху страницы с самыми высокими оценками.

Ваша лента и то, что разворачивается во времени

Откройте на телефоне приложение с фильмами, а потом дайте его другу с совсем другими вкусами. На вашем экране верхний ряд — научная фантастика; на его — старые комедии. То же приложение, те же фильмы в библиотеке, две разные страницы. Как оно решает, что поставить первым для каждого из вас?



То же приложение, разные верхние ряды

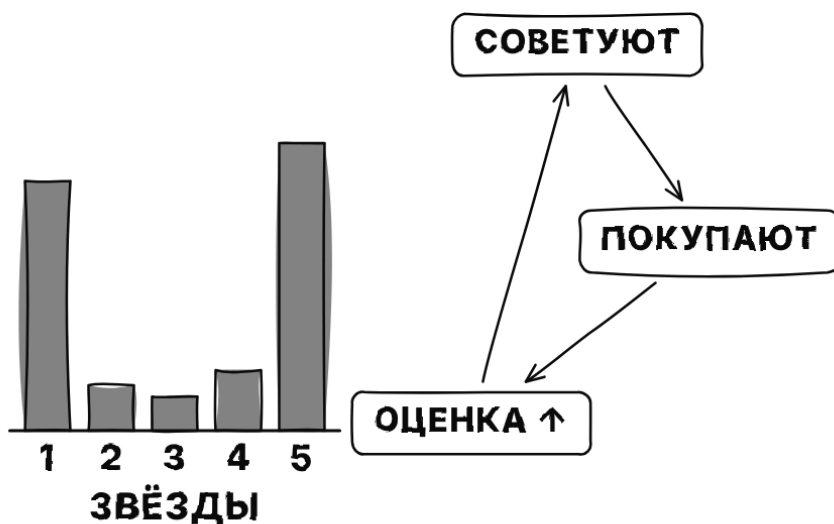
Рекомендательные системы — близкие родственники поиска и ранжирования: и те и другие выбирают набор объектов и

решают, какие показать первыми. Разница в том, что рекомендации личные. Любитель научной фантастики и поклонник комедий с Питером Селлерсом должны получать очень разные страницы результатов, и то же верно для покупок, музыки и новостей.

Чтобы узнать, что людям нравится, система прислушивается к двум видам сигналов. Иногда люди говорят это прямо — звёздными оценками и отзывами в магазине, на сайте о кино или о книгах; это явная обратная связь. Чаще они сообщают это только своими действиями, например пропуская песню в плейлисте; это неявная обратная связь. Пропуск может означать неприязнь, а может — лишь то, что песня не подошла к моменту. Когда ваша лента рекомендаций перестраивается после того, как вы пропустили несколько видео, она действует именно по такому тихому сигналу. В самом простом варианте система выучивает оценку для каждой пары из человека и объекта, например число звёзд, которое этот человек, вероятно, поставил бы, или шанс, что он купит. Затем для каждого человека она показывает объекты с самыми высокими оценками. Допустим, модель ожидает, что одному фильму вы поставите четыре с половиной звезды, другому три целых две десятых, а третьему две целых восемь десятых: она расставит их в этом порядке, четыре с половиной первым. Настоящие системы гораздо сложнее, они учитывают подробную активность и свойства каждого объекта; книга показывает книги о глубоком обучении, которые крупный интернет-магазин предложил одному из её собственных авторов.

Эти системы чрезвычайно ценны, но, построенные наивно поверх предсказательной модели, имеют серьёзные изъяны. Во-первых, обратная связь цензурирована: люди в основном

оценивают то, к чему относятся сильно. На пятизвёздочной шкале вы видите много единиц и пятёрок и странно мало троек, так что оценки не показывают, что думал средний зритель. Во-вторых, то, что люди покупают сегодня, отчасти результат того, что система рекомендовала вчера, и метод обучения может этого не учитывать. Так может возникнуть петля обратной связи: объект продвигают, поэтому больше людей его покупает, поэтому он выглядит лучше, поэтому его продвигают ещё сильнее. Как обращаться с цензурированной обратной связью, со стимулами и с такими петлями — всё ещё открытые вопросы исследований.



Перекошенные оценки, петли самоподпитки

Всё до сих пор имело фиксированную форму: один и тот же список фактов для каждого дома, одна и та же сетка пикселей

для каждого фото, один фиксированный набор классов, звёздная оценка всего лишь от человека и товара. И как только модель отвечала на пример, она его забывала, будто каждый новый случай никак не связан с предыдущим. Это допущение — что последовательные примеры независимы — часто не выполняется. В видеоролике может быть сколько угодно кадров, и то, что происходит в одном кадре, гораздо понятнее, если посмотреть на кадры до и после. С языком то же самое: при машинном переводе на вход приходит целое предложение, и на выходе должно получиться целое предложение. В отделении реанимации больницы модель могла бы наблюдать за пациентами и поднимать тревогу всякий раз, когда риск смерти в ближайшие двадцать четыре часа превышает некий порог, и было бы глупо каждый час выбрасывать историю пациента и судить только по последним показаниям.

КАДРЫ ВОКРУГ



ЛЮБАЯ ДЛИНА



ИСТОРИЯ БОЛЕЗНИ

Один кадр обманывает, кадры рассказывают

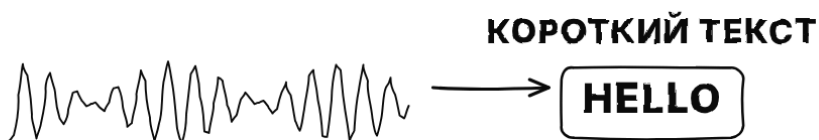
Такие задачи называются обучением на последовательностях: модель принимает последовательность, или выдаёт её, или и то и другое. Когда и вход, и выход — последовательности любой длины, это называется обучением «последовательность в последовательность», как при переводе или при превращении речи в текст. Книга называет несколько частных случаев.

Первый — разметка и синтаксический разбор, где каждое слово текста получает метку, и вход с выходом выровнены: меток столько же, сколько слов, в том же порядке. Можно пометить каждое слово его частью речи или отметить, какие слова входят в имена людей, мест или организаций. В предложении «Tom has

dinner in Washington with Sally» слова Tom, Washington и Sally были бы отмечены как имена, а остальные — нет.

TOM HAS DINNER IN
ENT — — —

WASHINGTON WITH SALLY
ENT — ENT



ТЫСЯЧИ ЧИСЕЛ

Метки по порядку; длинный звук, короткий текст

Второй — распознавание речи: на вход поступает запись, а на выходе получается письменная расшифровка. Звук обычно измеряют восемь тысяч или шестнадцать тысяч раз в секунду. При шестнадцати тысячах в секунду слово, на произнесение которого уходит полсекунды, — это восемь тысяч измерений, а на странице в нём может быть всего пять или шесть букв. Так что соответствия «один к одному» между звуком и текстом нет, и выход намного короче входа. На рисунке в книге слова «deep learning» растянуты вдоль звуковой волны, и каждая буква размазана по её отрезку. Всякий раз, когда ваш голосовой

помощник выписывает ваш вопрос на экран, он сжимает тысячи чисел в несколько слов. Люди удивительно хорошо это умеют, даже на хрипящей линии; для компьютеров это грозная задача.

Запомним

Рекомендательные системы ранжируют объекты для каждого человека, учась на явной обратной связи, например на оценках, и на неявной, например на пропусках. Самый простой вариант даёт оценку каждой паре из человека и объекта, например по ожидаемым звёздам, и показывает самые высокие оценки. Цензурированные оценки и петли обратной связи, где рекомендации подпитывают собственный успех, — серьёзные изъяны, которые остаются открытыми проблемами. Когда примеры зависят от того, что было раньше, как в видео, языке или истории болезни, нужно обучение на последовательностях, а когда обе стороны — последовательности любой длины, — обучение «последовательность в последовательность». Разметка сохраняет одну метку на слово, а распознавание речи сжимает тысячи звуковых измерений на слово в несколько букв.

Вопросы

1. Вы пропускаете песню в плейлисте. Почему рекомендательная система не может быть уверена, что она вам не нравится?

- a. Потому что пропуск — неявная обратная связь и может значить лишь то, что песня не подошла к моменту
 - b. Потому что пропуски никогда не записываются
 - c. Потому что обратной связью считаются только звёздные оценки
 - d. Потому что песню рекомендовали кому-то другому
2. Почему на пятизвёздочной шкале странно мало оценок в три звезды?
- a. Потому что система округляет их до одной или пяти
 - b. Потому что обратная связь цензурирована: люди в основном оценивают то, к чему относятся сильно
 - c. Потому что на большинстве сайтов три звезды ставить нельзя
 - d. Потому что рекомендуемые объекты всегда превосходны
3. Почему наблюдение за пациентом в реанимации — задача обучения на последовательностях?
- a. Потому что каждое показание не зависит от предыдущего
 - b. Потому что на выходе один класс
 - c. Потому что, судя только по последним показаниям, мы выбросили бы историю пациента, а она важна
 - d. Потому что показания приходят в виде фиксированного фото
4. Почему выход распознавания речи намного короче входа?
- a. Потому что распознаватель выбрасывает половину слов

- б. Потому что каждое слово — это многие тысячи звуковых измерений, но всего несколько букв
- с. Потому что расшифровки — это краткие пересказы
- д. Потому что звук измеряют лишь раз в секунду

Ответы

- 1. а) Пропуск — тихий, неявный сигнал: он может означать неприязнь, а может — лишь то, что песня не подошла к моменту.
- 2. б) Люди в основном оценивают то, к чему относятся сильно, поэтому видно много единиц и пятёрок и мало троек.
- 3. с) Было бы глупо каждый час выбрасывать историю пациента и судить только по последним показаниям.
- 4. б) При шестнадцати тысячах измерений в секунду слово длиной в полсекунды — это восемь тысяч измерений, но всего пять или шесть букв.

Разобранные примеры

Отмечаем имена В предложении «Tom has dinner in Washington with Sally» отметьте каждое слово как имя (Ent) или не имя (-), как при простой разметке именованных сущностей. Сколько всего меток и сколько из них имён?

Разметка выровнена: одна метка на слово, в том же порядке.

В предложении семь слов, значит, меток семь.

Tom, Washington и Sally называют человека, место и человека; остальные слова — нет.

Метки читаются так: Ent, -, -, -, Ent, -, Ent.

Ответ: семь меток, из них три имени: Tom, Washington и Sally

Задачи

1. На телефонной линии звук измеряют 8000 раз в секунду. Из скольких измерений состоит слово, на произнесение которого уходит четверть секунды, и почему это делает распознавание речи задачей с неравными длинами?

Подсказка: Четверть секунды — это 8000, делённое на 4.

Ответ: 8000 на одну четверть — это 2000 измерений для слова, в котором, может быть, четыре или пять букв. На входе тысячи чисел, а на выходе всего несколько букв, так что соответствия «один к одному» нет и выход намного короче входа.

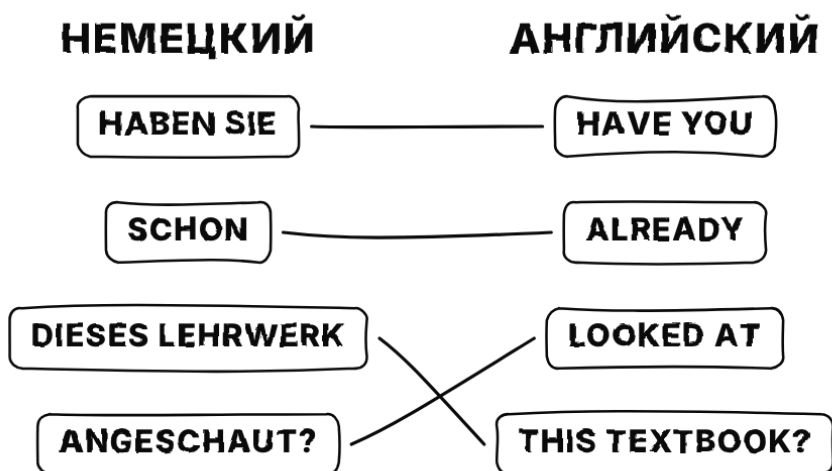
2. Модель ожидает, что трём фильмам вы поставите 3,2, 4,5 и 2,8 звезды. В каком порядке их покажет простая рекомендательная система и как петля обратной связи может потом сделать первый фильм на вид ещё лучше, чем он есть?

Подсказка: Сначала показывают самые высокие оценки; потом подумайте, что происходит с фильмами, которые показывают чаще.

Ответ: Она покажет первым фильм на 4,5 звезды, затем на 3,2 и затем на 2,8. Раз его показывают первым, верхний фильм чаще смотрят или покупают, эти покупки делают его для системы на вид ещё лучше, и его продвигают ещё сильнее: это петля обратной связи.

Всё наоборот и обучение без ответов

Наберите в приложении-переводчике короткое предложение, скажем «Have you already looked at this excellent textbook?», и нажмите кнопку, которая читает немецкий перевод вслух. Происходят две вещи, которые тянут в противоположные стороны. Слова переставляются, потому что в немецком другой порядок. А горстка букв превращается в длинный поток звука. Обе задачи — задачи о последовательностях, и обе труднее, чем кажутся.



ГЛАГОЛ В КОНЦЕ

Перевод меняет порядок

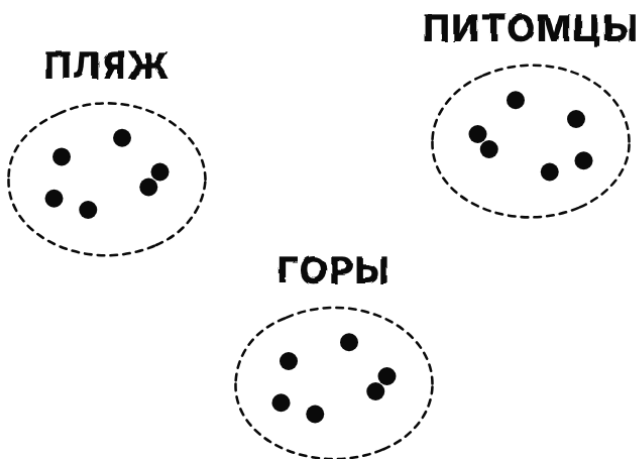
Чтение текста вслух называется синтезом речи, и это распознавание речи наоборот. Теперь на входе текст, а на выходе звукозапись, так что выход намного длиннее входа. При шестнадцати тысячах измерений в секунду предложение, на произнесение которого уходит две секунды, выходит тридцатью двумя тысячами чисел, хотя на входе было несколько десятков букв. Когда ваш голосовой помощник отвечает вам вслух, он делает именно это.

Машинный перевод добавляет новую трудность. При распознавании речи звуки и буквы хотя бы идут в одном порядке. При переводе вход и выход могут быть разной длины, а соответствующие части могут идти в другом порядке. Книга подшучивает над немецкой привычкой ставить глагол в конец. «Haben Sie sich schon dieses grossartige Lehrwerk angeschaut?» означает «Вы уже посмотрели этот замечательный учебник?» Если сохранить немецкий порядок слово в слово, получится «Вы себя уже этот замечательный учебник посмотрели?» — так по-английски никто не скажет. Книга добавляет, что похожие задачи встречаются и в других местах: определить, в каком порядке человек читает веб-страницу, — головоломка в двух измерениях, а вести разговор значит опираться на знания о мире и на всё, что было сказано задолго до этого. Чат-боту, которому вы пишете, приходится справляться именно с этим, держа в поле зрения весь разговор, чтобы решить, что сказать дальше.

Всё это было обучением с учителем: модели всегда давали правильные ответы. Книга сравнивает его с работой у начальника-диктатора, который стоит у вас за плечом и говорит, что именно делать в каждой ситуации. Скучно, но угодить легко: быстро заметить закономерность и повторять за начальником.

Противоположная работа — начальник, который понятия не имеет, чего хочет, вываливает на вас огромную кучу данных и говорит: «Займитесь-ка с ними наукой о данных!» Звучит расплывчато, потому что так и есть. Обучение на данных совсем без меток называется обучением без учителя, и вопросы, которые можно задать, ограничены только воображением.

КЛАСТЕРИЗАЦИЯ

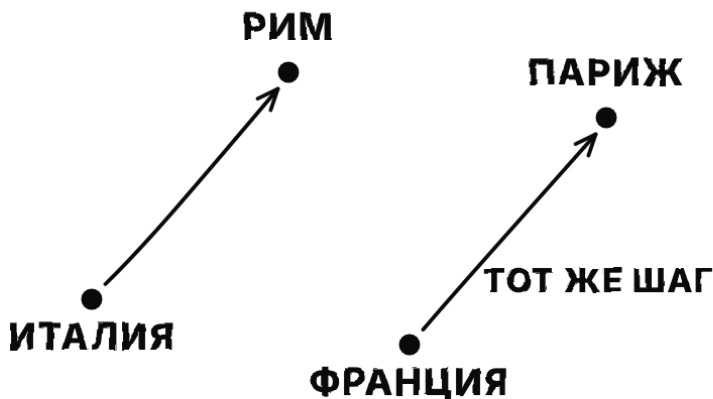


Группы, найденные без ответов

Книга даёт попробовать. Можно ли описать данные несколькими типичными примерами, чтобы фотографии разошлись на пейзажи, собак, младенцев, кошек и горные вершины, а покупатели — на группы, которые просматривают товары похоже? Это кластеризация. Могут ли несколько чисел уловить главное? Полёт мяча хорошо описывается его скоростью,

размером и массой, а портные подгоняют одежду по небольшому набору мерок; поиск таких чисел называется оценкой подпространства, а когда связь прямолинейная — методом главных компонент. Можно ли расположить объекты как точки в пространстве так, чтобы отношения стали направлениями? Тогда шаг от «Италии» к «Риму» — тот же, что от «Франции» к «Парижу»: возьмите Рим, вычтите Италию, прибавьте Францию — и окажетесь в Париже. Можно ли найти глубинные причины, например как цены на жильё, загрязнение, преступность, образование и зарплаты зависят друг от друга, по одним только данным? Это область причинности и вероятностных графических моделей.

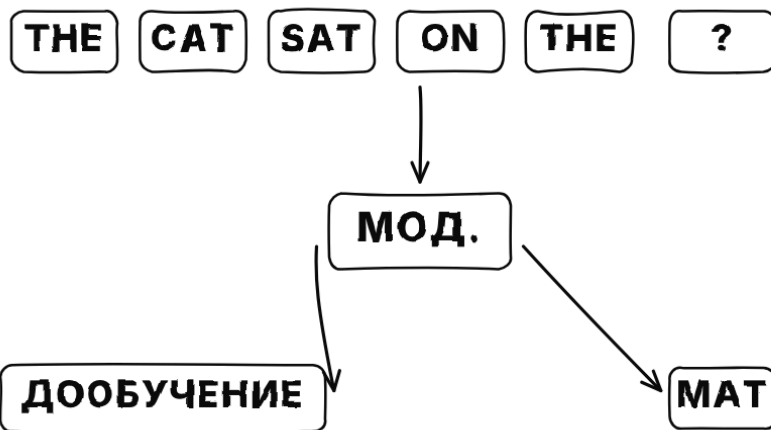
КАРТА СЛОВ



Слова как точки на карте

Недавнее и захватывающее направление — глубокие генеративные модели. Они так хорошо изучают, как распределены данные, что могут оценить, насколько вероятен пример, или создать совершенно новые собственные примеры. Первыми прорывами были вариационные автокодировщики, а затем генеративно-состязательные сети; более новые — нормализующие потоки и диффузионные модели. Генераторы изображений, которые рисуют картинку по набранному вами предложению, принадлежат к этому семейству.

Последним идёт самообучение, которое использует сами неразмеченные данные, чтобы составлять вопросы с известными ответами. Для текста: спрячьте случайное слово и учите модель угадывать его по окружающим словам. В «the cat sat on the ...» («кошка сидела на ...») пропущенное слово почти наверняка «mat», «коврике», и верный ответ всё время был в тексте, так что огромные объёмы текста можно использовать без всяких усилий по разметке. Для картинок модель можно попросить определить, где два вырезанных кусочка одного изображения находились друг относительно друга, дорисовать закрытую часть по остальному или сказать, получены ли два изменённых изображения из одного оригинала. То, чему учится такая модель, — её способ описывать данные — часто используют снова: немного дополнительной тренировки, называемой дообучением, приспособливает её к той конкретной задаче, которая на самом деле нужна. Чат-боты начинали именно так: сначала их тренировали на огромном количестве текста предсказывать пропущенные или следующие слова и только потом настраивали отвечать на вопросы.



Спрячь слово — угадай обратно

Запомним

Синтез речи превращает несколько букв в длинную звукозапись — это распознавание речи наоборот. Перевод ещё труднее, так как вход и выход различаются по длине, а их части могут идти в другом порядке, как немецкий глагол в конце. Обучение без учителя работает с данными без меток: кластеризация, поиск нескольких говорящих чисел, такое расположение слов, что Рим минус Италия плюс Франция попадает в Париж, и поиск причин. Генеративные модели изучают данные достаточно хорошо, чтобы создавать новые примеры, как это делают диффузионные модели. Самообучение составляет собственные вопросы из данных, например спрятанные слова, и результат потом дообучают для нужной задачи.

Вопросы

1. Что, согласно лекции, делает машинный перевод труднее распознавания речи?
 - a. Его вход намного длиннее
 - b. Соответствующие части входа и выхода могут идти в другом порядке
 - c. У него нет меток
 - d. Он работает только для немецкого

2. Почему самообучение может использовать огромные объёмы текста без усилий по разметке?
- a. Потому что верный ответ для спрятанного слова уже был в тексте
 - b. Потому что в тексте нет структуры, которой можно научиться
 - c. Потому что люди размечают текст бесплатно
 - d. Потому что оно не учитывает слова вокруг пропуска
3. Какой из этих вопросов — вопрос кластеризации?
- a. Какие несколько чисел описывают полёт мяча?
 - b. Ведёт ли загрязнение к росту преступности?
 - c. Можно ли разделить этих покупателей на группы, которые просматривают товары похоже?
 - d. Какое следующее слово в этом предложении?
4. Что такое дообучение?
- a. Тренировка модели с нуля на размеченных данных
 - b. Удаление ненужных частей модели
 - c. Настройка генеративной модели на более чёткие картинки
 - d. Немного дополнительной тренировки, которая приспособливает модель после самообучения к нужной задаче

Ответы

1. b) При переводе вход и выход могут быть разной длины, а соответствующие части могут идти в другом порядке.

2. а) В «the cat sat on the ...» верный ответ всё время был в тексте, так что никому не нужно ничего размечать.

3. с) Описание данных несколькими типичными примерами, например группами покупателей, которые просматривают товары похоже, — это кластеризация.

4. d) То, чему учится модель при самообучении, часто используют снова: немного дополнительной тренировки, называемой дообучением, приспособливает её к конкретной задаче.

Разобранные примеры

От букв к звуку Система синтеза речи выдаёт звук с частотой 16 000 измерений в секунду. На произнесение предложения уходит 2 секунды. Сколько чисел она выдаёт и почему синтез речи — задача, где выход намного длиннее входа?

Каждая секунда звука — это 16 000 измерений.

Две секунды дают 16 000 на 2, то есть 32 000.

На вход предложение поступило несколькими десятками букв.

Так короткий вход превращается в очень длинный выход — это распознавание речи наоборот.

Ответ: 32 000 чисел для предложения из нескольких десятков букв

Задачи

1. Используя идею, что отношения становятся направлениями, закончите: Париж минус Франция плюс Италия даёт что? А Рим минус Италия плюс Франция?

Подсказка: Если убрать одну страну и добавить другую, вы переходите от одной столицы к столице другой страны.

Ответ: Париж минус Франция плюс Италия попадает в Рим, а Рим минус Италия плюс Франция попадает в Париж: шаг от страны к её столице в обоих случаях — одно и то же направление.

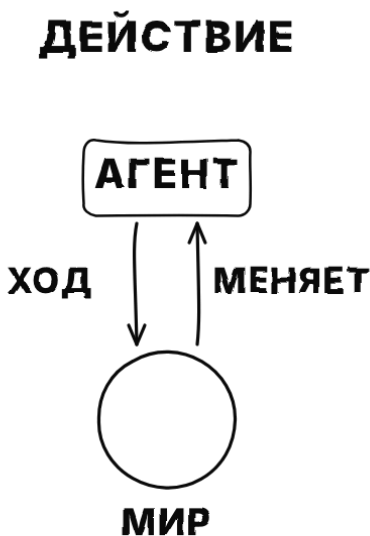
2. Объясните на примере двух начальников из книги разницу между обучением с учителем и без учителя.

Подсказка: Один начальник говорит вам всё; другой — ничего.

Ответ: При обучении с учителем начальник стоит у вас за плечом и даёт правильный ответ для каждой ситуации, так что нужно лишь повторять закономерность. При обучении без учителя начальник просто вручает грудку данных без ответов и просит найти в ней что-то полезное, так что структуру нужно открыть самому.

Когда машина действует

Посмотрите, как маленький ребёнок строит башню из кубиков. Никто не даёт ему листка с правильными ответами. Он ставит один кубик на другой, башня шатается, стоит или падает, и он пробует что-то другое. Каждый ход меняет то, что он увидит дальше, и единственный приговор всем его решениям — стоит ли башня в конце.



Учиться в отрыве или действовать

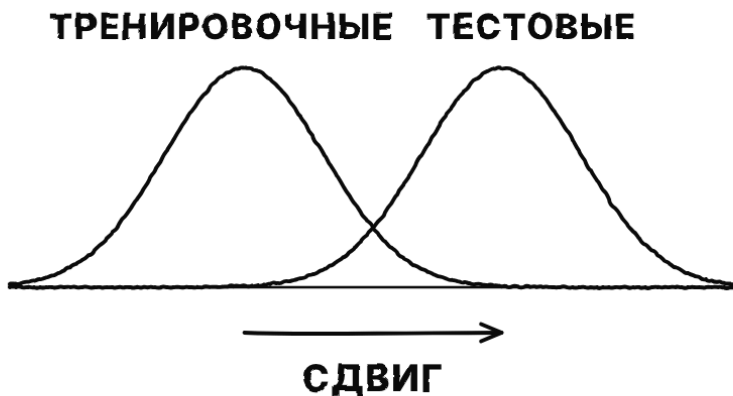
До сих пор, признаёт книга, она не спрашивала, откуда берутся данные и что происходит после того, как модель дала ответ. И

обучение с учителем, и обучение без учителя работают одинаково просто: заранее схватить большую грудку данных, а потом запустить машину, ищущую закономерности, больше никогда не касаясь мира. Поскольку всё обучение происходит после того, как машина отрезана от мира, это называется офлайн-обучением. Книга изображает обучение с учителем простой картинкой: мир предоставляет примеры, кто-то их размечает, и модель учится на готовой грудке.

В офлайн-обучении есть своя прелесть, ведь можно изучать распознавание закономерностей само по себе, без сложностей мира, который движется и отвечает. Но это ограничивает. Читатели, выросшие на романах Азимова о роботах, представляют разумные машины, которые не только предсказывают, но и действуют в мире. Книга хочет думать об агентах, а не только о предсказательных моделях, и о выборе действий, а не только о предсказаниях. Предсказание оставляет мир прежним; действие его меняет. Чтобы тренировать агента, нужно учитывать, как его собственные действия изменяют то, что он увидит дальше, а для этого офлайн-обучение — неподходящий инструмент.

Как только появляется обмен с миром, новые вопросы громоздятся один на другой. Помнит ли мир, что мы делали раньше? Хочет ли он помочь, как человек, медленно и чётко читающий текст в распознаватель речи? Хочет ли он нас обогреть? Пример книги — спамеры: они всё время переписывают свои письма, чтобы проскользнуть мимо вашего спам-фильтра, так что мир фильтра даёт отпор. Меняется ли мир со временем, сам или из-за наших же инструментов, так что будущие данные уже не похожи на прошлые? Всё это ведёт к

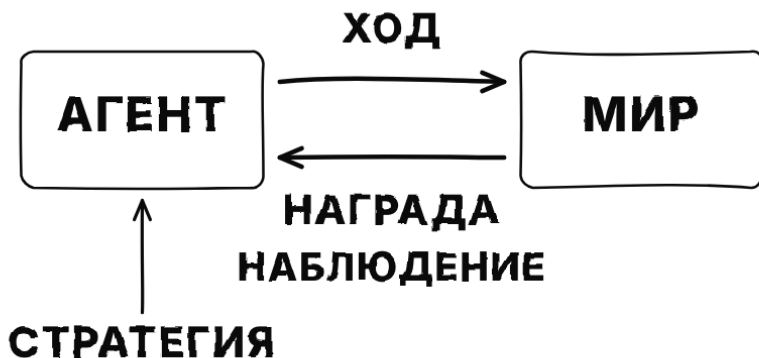
сдвигу распределения, когда данные при проверке отличаются от данных при тренировке. Пример книги знаком любому студенту: домашние задания пишут ассистенты, а экзамен — лектор.



Тестовые данные могут сдвигаться

Главная схема обучения действовать — обучение с подкреплением. Его используют в робототехнике, в системах, ведущих разговоры, и в компьютерных противниках в видеоиграх. В сочетании с глубоким обучением, как глубокое обучение с подкреплением, оно попадало в заголовки. Программа под названием «глубокая Q-сеть» научилась обыгрывать людей в старые аркадные игры Atari, не видя ничего, кроме экрана. А AlphaGo победила чемпиона мира в настольной игре го.

Книга изображает обучение с подкреплением в виде цикла. На каждом шаге агент получает наблюдение мира и выбирает действие, которое передаётся обратно в мир через какой-то механизм, иногда называемый исполнительным устройством. Затем мир даёт агенту награду и новое наблюдение, и агент выбирает снова, шаг за шагом. Поведением агента управляет его стратегия — попросту правило, которое сопоставляет тому, что он наблюдает, то, что он делает. Цель обучения с подкреплением — найти хорошие стратегии.



Цикл обучения с подкреплением

Схема необычайно общая. Даже обычную классификацию можно переодеть в обучение с подкреплением: дать агенту по одному действию на каждый класс и пусть мир возвращает награду,

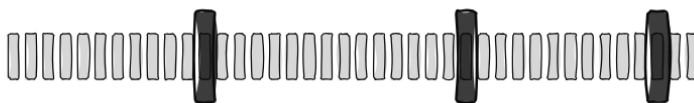
равную потере исходной задачи. Но обучение с подкреплением охватывает и задачи, с которыми обучение с учителем не справляется. При обучении с учителем каждый тренировочный вход приходит с правильной меткой. При обучении с подкреплением мир не говорит, какое действие было лучшим; он выдаёт только награду и часто даже не говорит, какое действие её заслужило.

Шахматы показывают это наглядно. Единственная настоящая награда приходит в конце: скажем, одно очко за победу и минус одно за поражение. Партия может длиться по сорок ходов у каждого игрока. Если наивно разделить победу поровну на все сорок, каждый ход получит одну сороковую очка, две с половиной сотых, будь то блестящий ход, который принёс победу, или небрежный, который чуть не привёл к поражению. Это явно неверно, и поиск того, какие действия на самом деле заслуживают заслуги или вины, называется задачей распределения заслуг. Люди тоже с ней сталкиваются: тот, кого повысили одиннадцатого октября, обязан повышением многим удачным решениям за прошедший год, и чтобы получить повышение снова, ему нужно понять, какие это были решения.

РАСПРЕДЕЛЕНИЕ ЗАСЛУГ

ПО 1/40

+1



КЛЮЧЕВЫЕ ХОДЫ

Кто заслужил победу?

Запомним

Офлайн-обучение сначала собирает данные, а потом учится, будучи отрезанным от мира; действующий агент меняет те самые данные, которые увидит. Когда мир помнит, помогает, сопротивляется или меняется, тестовые данные могут отличаться от тренировочных, и это называется сдвигом распределения. Обучение с подкреплением — это цикл: наблюдать, действовать, получать награду, снова наблюдать, а действия выбирает стратегия. Глубокое обучение с подкреплением обыграло людей в игры Atari и в го. Награды часто приходят поздно и не говорят, какие действия их заслужили, и разбор заслуг и вины — это задача распределения заслуг.

Вопросы

1. Почему офлайн-обучение — неподходящий инструмент для тренировки агента?
 - a. Потому что действия агента меняют мир, а значит, и данные, которые он увидит дальше
 - b. Потому что агентам нужно меньше данных
 - c. Потому что офлайн-обучение не умеет распознавать закономерности
 - d. Потому что агенты работают только с текстом

2. Чем главным образом обучение с подкреплением отличается от обучения с учителем по тому, что ему сообщают?
 - a. Ему сообщают лучшее действие на каждом шаге
 - b. Оно получает только награду, часто не зная, какое действие её заслужило
 - c. Оно вообще не получает обратной связи
 - d. Ему заранее дают размеченные примеры
3. Почему разделить шахматную победу поровну на все сорок ходов — плохой способ распределить заслуги?
 - a. Потому что сорок ходов — слишком мало
 - b. Потому что награда за победу равна нулю
 - c. Потому что так блестящий победный ход и небрежный получают одинаковую заслугу
 - d. Потому что в шахматах нет наград
4. Спамеры переписывают свои письма, чтобы обойти спам-фильтр. На какой вопрос о мире это ответ «да»?
 - a. Хочет ли мир нам помочь?
 - b. Нет ли у мира памяти?
 - c. Наблюдается ли мир полностью?
 - d. Хочет ли мир нас обыграть?

Ответы

1. a) Чтобы тренировать агента, нужно учитывать, как его собственные действия изменят то, что он увидит дальше, а офлайн-обучение этого не умеет.

- 2. b)** При обучении с подкреплением мир не говорит, какое действие было лучшим; он выдаёт только награду, часто не говоря, какое действие её заслужило.
- 3. c)** При равном делении каждый ход получает одну сороковую очка, блестящий он или небрежный, и это явно неверно.
- 4. d)** Спамеры всё время переписывают свои письма, чтобы проскользнуть мимо фильтра: мир фильтра даёт отпор.

Разобранные примеры

Наивный раздел победы Шахматная партия приносит награду 1 за победу. Победитель сделал 40 ходов. Если разделить награду поровну между всеми ходами, какую заслугу получит каждый ход и почему это проблема?

Единственная награда, 1, приходит в конце партии.

При равном делении на 40 ходов каждый ход получает 1, делённое на 40.

1, делённое на 40, — это 0,025.

Каждый ход получает одинаковые 0,025, принёс он победу или чуть её не погубил, так что равное деление не отличает хорошие ходы от плохих: это задача распределения заслуг.

Ответ: 0,025 на ход, одинаково для хороших и плохих ходов

Задачи

1. Опишите цикл обучения с подкреплением для компьютерного противника в видеоигре: что он наблюдает, каковы его действия, какой может быть его награда и какова его стратегия?

Подсказка: Наблюдать, действовать, получать награду, снова наблюдать.

Ответ: Он наблюдает экран или состояние игры, его действия — это ходы или нажатия кнопок, которые он может сделать, а наградой могут быть набранные очки или прохождение уровня. Его стратегия — правило, выбирающее действие по тому, что он сейчас наблюдает, и обучение стремится сделать это правило хорошим.

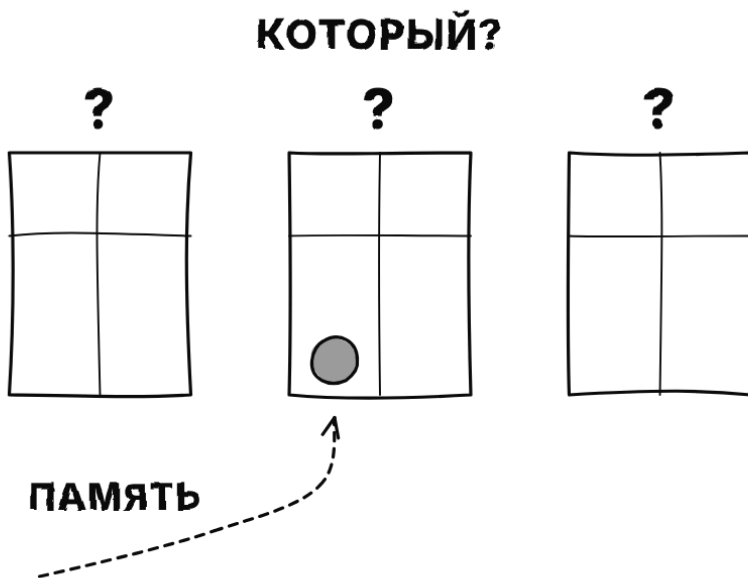
2. Объясните, как классификацию можно переформулировать как обучение с подкреплением, как это описывает книга.

Подсказка: Какими были бы действия и награда?

Ответ: Дайте агенту по одному действию на каждый класс, так что выбрать действие значит выбрать класс, и пусть мир возвращает награду, равную потере исходной задачи классификации. Тогда агент учится выбирать классы, приносящие лучшую награду, а это то же самое, что хорошо классифицировать.

Слепые пятна, риск и старые корни

Представьте, что ваш робот-пылесос забрёл в один из нескольких одинаковых чуланов в вашем доме и застрял. Изнутри все чуланы выглядят совершенно одинаково: четыре стены, дверь, полка. То, что он видит прямо сейчас, не может сказать ему, в каком он чулане. Только вспомнив, что он видел, прежде чем зайти, он — или вы — сможет понять, где он.

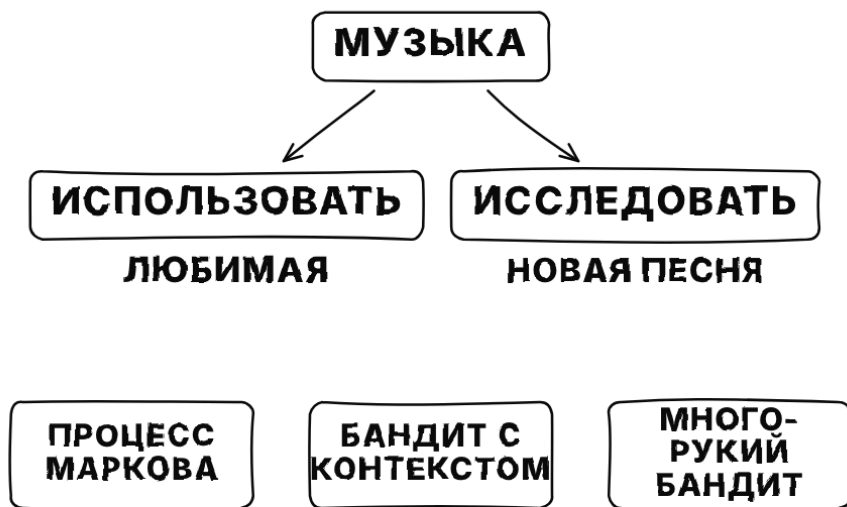


Одинаковые чуланы

Это задача частичной наблюдаемости. В обучении с подкреплением текущее наблюдение агента может не сообщать

ему всего о состоянии, в котором он находится. Чтобы действовать хорошо, ему, возможно, придётся собирать настоящее по кусочкам из того, что он видел раньше.

Есть и другая трудность. В любой момент агент может знать одну стратегию, которая работает вполне прилично, а лучших, которых он никогда не пробовал, может быть много. Ему всё время приходится выбирать между использованием лучшей известной стратегии и исследованием других, возможно, жертвуя частью награды сейчас в обмен на знание. Ваше музыкальное приложение сталкивается с этим каждый день: поставить песню, которая вам точно нравится, или рискнуть новой, которую вы можете полюбить, а можете пропустить.

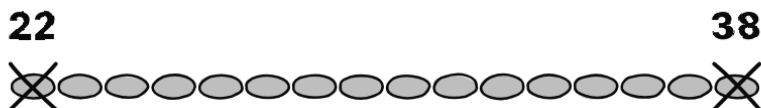


Использовать или исследовать; более простые случаи

Если сложить всё вместе, общая задача очень широка: действия меняют то, что агент увидит дальше, награды видны только для действительно совершённых действий, а мир может быть виден полностью или лишь отчасти. Браться за всё это сразу, возможно, слишком много, да и во многих реальных задачах всего этого нет, поэтому исследователи изучают частные случаи. Когда агент видит всё состояние мира, задача называется марковским процессом принятия решений. Когда состояние не зависит от прежних действий агента, это задача контекстного бандита. А когда состояния нет вовсе, а есть только набор возможных действий с поначалу неизвестными наградами, это классическая задача многорукого бандита, названная по ряду игровых автоматов — «одноруких бандитов».

Закончив этот обзор задач, книга обращается к истории. Многие методы глубокого обучения появились недавно, но идеи обучения на данных изучаются веками. Желание анализировать данные и предвидеть, что будет дальше, лежит в основе значительной части науки и математики. Две закономерности случая носят имена великих математиков: распределение Бернулли, по имени Якоба Бернулли, умершего в 1705 году, и гауссово распределение, по имени Карла Фридриха Гаусса. Гаусс также придумал метод наименьших квадратов, который подгоняет правило к измерениям, делая квадраты промахов как можно меньше, и который до сих пор применяют от страхования до медицинской диагностики. Такие инструменты укрепили экспериментальные науки; закон Ома, связывающий ток через резистор с напряжением на нём, идеально описывается прямолинейной моделью.

ОБЫЧНЫЙ ФУТ



$$480 \div 16 = 30$$

$$420 \div 14 = 30$$

УСЕЧЁННОЕ СРЕДНЕЕ

Шестнадцать ступней, усечённое среднее

Ещё раньше люди хорошо чувствовали оценки. Книга по геометрии шестнадцатого века Якоба Кёбеля показывает, как найти длину типичной ступни. Когда люди выходили из церкви, шестнадцать взрослых мужчин выстраивались в ряд пятка к носку, общую длину их ступней измеряли и делили на шестнадцать. Допустим, шестнадцать ступней дали четыреста восемьдесят сантиметров: делим на шестнадцать — получаем тридцать сантиметров. Позже метод улучшили, чтобы справляться со странными ступнями: мужчину с самыми короткими ступнями и мужчину с самыми длинными отправляли прочь, а усредняли только остальных четырнадцать. Если у тех двоих было двадцать два и тридцать восемь сантиметров, оставшиеся четырнадцать дают в сумме четыреста

двадцать, а четыреста двадцать, делённое на четырнадцать, — снова тридцать. Это один из самых ранних примеров усечённого среднего — среднего, которое выбрасывает крайние значения, чтобы одно странное значение не могло его испортить.

Статистика по-настоящему взлетела, когда данные стали собирать в больших количествах. Рональд Фишер, один из её первопроходцев, сформировал и её теорию, и её применение в генетике, и некоторые его методы и идеи до сих пор лежат в основе современной статистики. Даже его данные оказались долговечными: измерения цветков ириса, опубликованные им в 1936 году, до сих пор иногда используют, чтобы испытывать методы обучения. Фишер был также пропагандистом евгеники — напоминание о том, что у морально сомнительного использования науки о данных такая же долгая история, как и у полезного.

Другие влияния пришли из теории информации Клода Шеннона и теории вычислений Алана Тьюринга. В знаменитой статье Тьюринг спросил: «Могут ли машины мыслить?» — и предложил то, что теперь называют тестом Тьюринга: машина считается разумной, если судье, общающемуся с ней только письменными сообщениями, трудно отличить её ответы от человеческих. Пообщавшись с современным чат-ботом, многие люди обнаружили, что этот тест уже не так гипотетичен, как казался когда-то. Нейронаука и психология добавили ещё больше, ведь люди явно ведут себя разумно, и можно надеяться понять и скопировать, как именно. Дональд Хебб в своей книге об организации поведения предположил, что нервные клетки учатся через положительное подкрепление, — эту идею теперь называют правилом обучения Хебба. Она вдохновила

перцептрон Розенблатта и заложила основу для градиентных методов, которые используются в глубоком обучении сегодня: усиливать то, что ведёт к хорошему поведению, ослаблять то, что ведёт к плохому, и так прийти к хорошим положениям ручек. Из этого биологического вдохновения нейронные сети и получили своё название; больше века, со времён моделей Бейна и Шеррингтона, исследователи пытаются строить схемы, похожие на сети нервных клеток.

Запомним

Агент может видеть лишь часть своей ситуации, как робот, застрявший в одном из многих одинаковых чуланов, и должен опираться на то, что видел раньше. Он также должен балансировать между использованием лучшей известной стратегии и исследованием других, которые могут оказаться лучше. Частные случаи обучения с подкреплением — марковский процесс принятия решений, контекстный бандит и многорукий бандит. Обучению на данных века, от Бернулли и наименьших квадратов Гаусса до ступней шестнадцати мужчин, усреднённых у церкви, позже в виде усечённого среднего. Статистика, теория информации, теория вычислений и изучение мозга — всё это питало машинное обучение, а нейронные сети получили своё название от нервных клеток.

Вопросы

1. Почему робот, застрявший в одном из нескольких одинаковых чуланов, не может понять, где он, по тому, что видит сейчас?
 - a. Потому что его датчики сломаны
 - b. Потому что все чуланы выглядят одинаково, так что текущее наблюдение не раскрывает всю его ситуацию
 - c. Потому что у него нет стратегии
 - d. Потому что он исследует
2. В каком частном случае обучения с подкреплением нет состояния вовсе, а есть только действия с неизвестными наградами?
 - a. Марковский процесс принятия решений
 - b. Задача контекстного бандита
 - c. Задача многорукого бандита
 - d. Офлайн-обучение
3. Почему более поздние измерители ступней отправляли прочь мужчин с самыми короткими и самыми длинными ступнями?
 - a. Чтобы одно странное значение не могло испортить среднее
 - b. Чтобы фут получился длиннее
 - c. Потому что шестнадцать мужчин — слишком много, чтобы измерить
 - d. Чтобы следовать методу наименьших квадратов Гаусса
4. Какой принцип из идеи Хебба до сих пор направляет то, как глубокое обучение настраивает свои ручки?

- a. Измерять каждую нервную клетку напрямую
- b. Точно копировать проводку мозга
- c. Не трогать ручки после первой попытки
- d. Усиливать то, что ведёт к хорошему поведению, и ослаблять то, что ведёт к плохому

Ответы

- 1. **b)** Это частичная наблюдаемость: текущее наблюдение агента может не сообщать ему всего о состоянии, в котором он находится.
- 2. **c)** Когда состояния нет, а есть только набор действий с поначалу неизвестными наградами, это классическая задача многорукого бандита.
- 3. **a)** Усечённое среднее выбрасывает крайние значения, чтобы одно странное значение не могло его испортить.
- 4. **d)** Идея Хебба заложила основу для сегодняшних градиентных методов: усиливать то, что ведёт к хорошему поведению, ослаблять то, что ведёт к плохому.

Разобранные примеры

Фут Кёбеля и его усечённый вариант Ступни шестнадцати мужчин, выстроенные пятка к носку, дают в сумме 480 сантиметров. Самая короткая ступня — 22 сантиметра, самая длинная — 38. Найдите обычное среднее и усечённое среднее после того, как этих двоих отправили прочь.

Обычное среднее делит сумму на число мужчин: $480, \text{ делённое на } 16,$ — это 30 сантиметров.

Для усечённого среднего уберите самую короткую и самую длинную ступни: $480 \text{ минус } 22 \text{ минус } 38$ — это 420 сантиметров.

Остаётся четырнадцать мужчин, так что делим на 14.

420, делённое на 14, — 30 сантиметров; здесь две крайности уравнивали друг друга, но одна очень странная ступня уже не смогла бы утянуть ответ.

Ответ: 30 сантиметров в обоих случаях

Задачи

1. Ступни десяти мужчин дают в сумме 300 сантиметров, но один из них, с результатом 50 сантиметров, по ошибке был в огромных сапогах, а самая короткая ступня — 25 сантиметров. Сравните обычное среднее с усечённым, без самой короткой и самой длинной.

Подсказка: Для усечённого среднего уберите обе крайности и разделите на восемь.

Ответ: Обычное среднее — 300, делённое на 10, то есть 30 сантиметров. Если убрать 25 и 50, остаётся 225 сантиметров на 8 мужчин, около 28,1 сантиметра. Сапог потянул обычное среднее вверх; усечённое среднее он не обманывает.

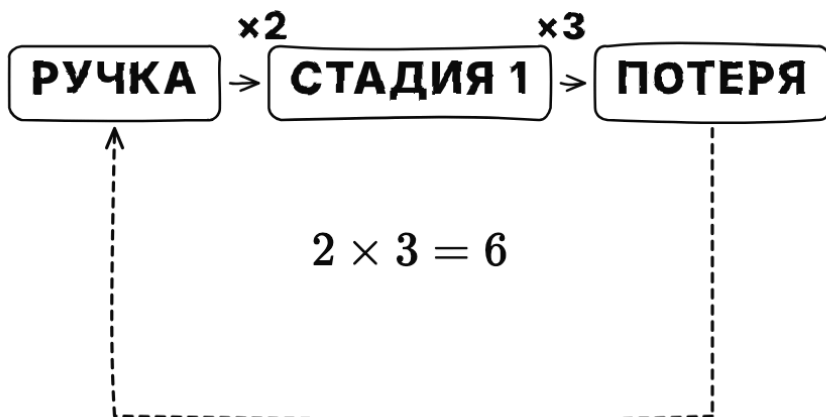
2. Объясните на собственном примере выбор между исследованием и использованием.

Подсказка: Подумайте о ресторане, дороге на работу или плейлисте.

Ответ: Пойти в любимый ресторан — это использование: вы знаете, что хорошо поедите. Попробовать новый — это исследование: он может оказаться хуже, а может и лучше, и узнать это можно, только попробовав. Ученик, который никогда не исследует, может застрять на просто неплохом варианте; тот, кто всё время исследует, никогда не пользуется тем, что узнал.

Из тупика — к взрыву

Вспомните первую компьютерную игру, в которую вы когда-либо играли, а потом — игру на сегодняшнем телефоне. Скачок в том, что маленькая машина может нарисовать и как быстро, огромен. И вот поворот: чипы, созданные, чтобы игры хорошо выглядели, оказались именно тем, чего ждали нейронные сети. Чтобы понять почему, нужны две идеи, лежащие в основе этих сетей, и история того, почему они когда-то застопорились.



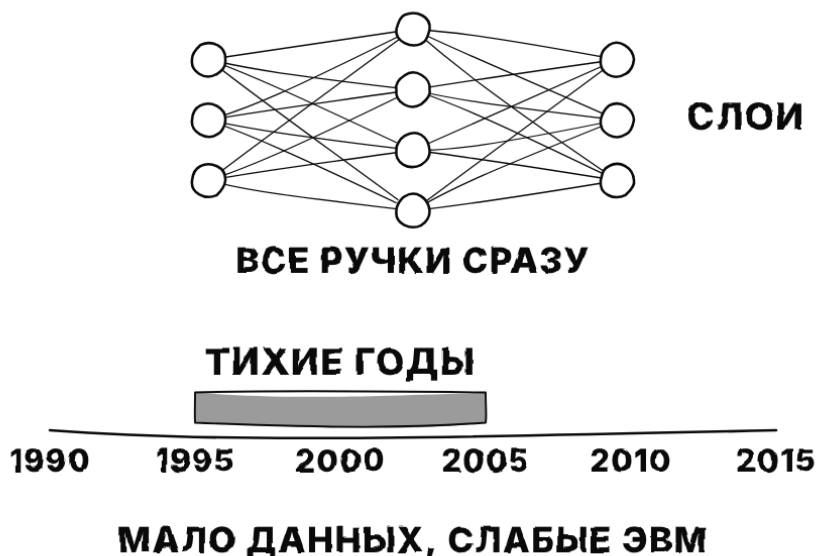
ОБРАТНОЕ РАСПРОСТРАНЕНИЕ

Цепное правило в числах

Почти любая нейронная сеть сегодня опирается на два принципа. Первый — слои: стадии обработки, в которых чередуются простые прямолинейные комбинации чисел и простые изгибы, так что каждая стадия немного преобразует полученное, прежде чем передать дальше. Второй — способ подстраивать ручки всей сети сразу с помощью цепного правила из математического анализа; этот метод называется обратным распространением ошибки. Цепное правило гласит, если сказать словами: если одна стадия питает следующую, влияние ранней ручки на итоговую потерю равно произведению влияний по пути. Например, если небольшой поворот ручки повышает выход первой стадии на две единицы на единицу поворота, а каждая единица этого выхода повышает потерю на три, то ручка повышает потерю на два, умноженное на три, — шесть единиц на единицу поворота. Обратное распространение ошибки применяет это от конца сети к началу и так за один проход узнаёт для каждой ручки, в какую сторону её повернуть.

После раннего быстрого прогресса исследования нейронных сетей зачихали примерно с 1995 по 2005 год, по двум причинам. Тренировка сети требует очень много вычислений, и хотя к концу века памяти было вдоволь, вычислительной мощности не хватало. А наборы данных были маленькими: ирисы Фишера 1936 года всё ещё были популярной проверкой, а собрание из шестидесяти тысяч рукописных цифр считалось огромным. При таком малом количестве данных и вычислений другие сильные статистические инструменты на практике справлялись лучше, среди них ядерные методы, деревья решений и графические модели. В отличие от нейронных сетей, им не нужны были

недели на тренировку, и они давали предсказуемые результаты с надёжными математическими гарантиями.



Слои и тихие годы

Изменились данные. Интернет, компании, обслуживающие онлайн сотни миллионов людей, дешёвые и хорошие датчики повсюду и всё дешевеющие хранилища вместе дали данные в огромных масштабах. Вычисления тоже продолжали дешеветь, и прежде всего графические чипы, созданные для компьютерных игр, — графические процессоры, GPU, — преобразили то, что могло вычислять глубокое обучение. Внезапно алгоритмы и модели, казавшиеся безнадёжными, стали доступны.

Книга излагает это в таблице, десятилетие за десятилетием. В 1970 году типичным набором данных были цветки ириса, около сотни примеров; у компьютера был килобайт памяти, и он делал около ста тысяч вычислений в секунду. К 1990 году типичным набором данных были десять тысяч отсканированных символов, при десяти мегабайтах памяти и десяти миллионах вычислений в секунду. К 2020 году набором данных могла быть целая социальная сеть, около триллиона объектов; память дошла до ста гигабайт, а одна машина, набитая графическими чипами, делала около миллиона миллиардов вычислений в секунду. Сравните рост за эти пятьдесят лет. Данные выросли со ста до триллиона, в десять миллиардов раз; память выросла с килобайта до ста гигабайт, в сто миллионов раз. Значит, память отстала от данных в сто раз. Вычислительная мощность, говорит книга, обогнала даже данные. Это подтолкнуло лучшие методы к моделям, которые бережливы с памятью, но могут щедро тратить вычисления на настройку своих ручек, и золотая середина сместилась от линейных моделей и ядерных методов к глубоким нейронным сетям. Это объясняет и то, почему многие опоры глубокого обучения — многослойные перцептроны, свёрточные сети, сети с долгой краткосрочной памятью и Q-обучение — по сути были заново открыты в последнее десятилетие, пролежав годы без дела.

РОСТ С 1970 ГОДА

ДАННЫЕ ×10 МЛРД



ВЫЧИСЛЕНИЯ: ЕЩЁ БОЛЬШЕ



ПАМЯТЬ ×100 МЛН



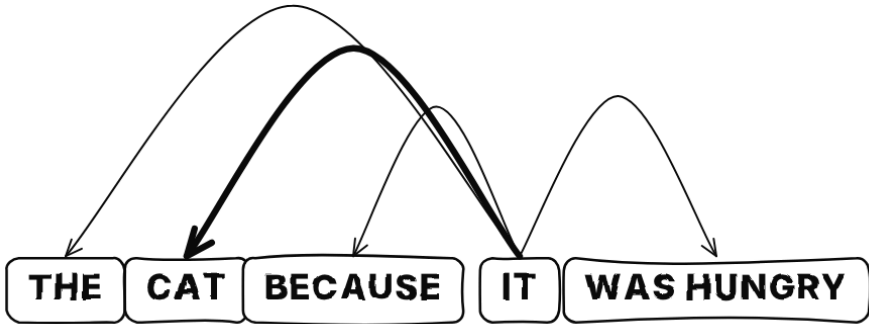
| | | | | | | | | |

ДЕЛЕНИЕ: ×10

Что выросло и насколько

И всё же прогресс был больше, чем старые рецепты на более быстрых машинах. Книга сравнивает его с кембрийским взрывом — всплеском, когда животная жизнь вдруг стала разнообразной, — и перечисляет несколько новых идей. Дропаут борется с переобучением, впрыскивая случайный шум по всей сети во время тренировки. Внимание решило старую задачу: как дать системе больше памяти и сложности, не добавляя ручек. Вместо того чтобы втискивать целое предложение в сводку фиксированного размера перед переводом, модель с вниманием хранит своего рода обучаемый указатель назад на нужную часть, и длинные предложения стали переводиться гораздо точнее.

ВНИМАНИЕ



ТРАНСФОРМЕР: ТОЛЬКО ВНИМАНИЕ

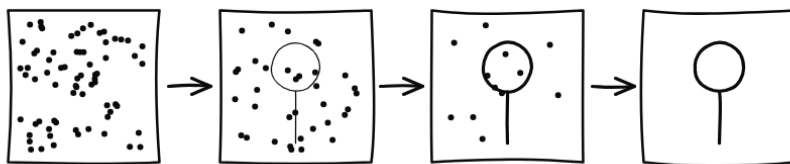
Внимание оглядывается назад

Трансформер, построенный целиком на внимании, становится всё лучше по мере роста данных, модели и вычислений. Он добился успеха в языке, зрении, речи, обучении с подкреплением и не только; один Трансформер, обученный на тексте, изображениях, движениях суставов робота и нажатиях кнопок, может играть в Atari, подписывать изображения, разговаривать и управлять роботом. Языковые модели предсказывают текст по другому тексту и по мере увеличения обрели всё новые способности. Чат-бот, которому вы пишете, — собственный пример книги: ChatGPT, языковая модель, согласованная с тем, чего хотят люди, которая может помочь отладить программный код или творчески писать в разговоре. Многоступенчатые архитектуры, например сети с памятью, позволяют сети менять

своё внутреннее состояние шаг за шагом, проводя цепочку рассуждений примерно так, как процессор обновляет свою память во время вычисления.

Генеративные модели тоже совершили скачки. Генеративно-состязательные сети ставят генератор, который делает поддельные примеры, против дискриминатора, который пытается отличить подделку от настоящего; ручки генератора настраивают, пока дискриминатор не перестанет их различать. Среди результатов — скачущие зебры, сделанные из лошадей, и убедительные поддельные лица знаменитостей, и даже рисующие любители могут превратить грубый набросок в почти фотографическую картинку. Позже диффузионные модели научились обращать постепенное добавление шума, выстраивая картинку шаг за шагом из помех, и на них работают системы «текст в изображение», такие как DALL-E 2 и Imagen. Генератор изображений, который рисует всё, что вы опишете, — это та самая идея в действии.

ДИФФУЗИЯ



УБРАТЬ ШУМ

Картинка из шума

Инженерия тоже имела значение. Рабочая лошадка оптимизации, стохастический градиентный спуск, работает с небольшими пачками примеров за раз, но маленькие пачки недогружают графические чипы. Распределите тренировку на тысячу двадцать четыре чипа, каждый из которых берёт по тридцать два изображения, и каждый шаг обрабатывает тридцать две тысячи семьсот шестьдесят восемь изображений, около тридцати двух тысяч. Более поздние работы довели это до шестидесяти четырёх тысяч и сократили тренировку стандартной сети распознавания изображений, ResNet-50 на коллекции ImageNet, до менее чем семи минут, тогда как поначалу она занимала дни. Параллельные вычисления помогли и обучению с подкреплением достичь сверхчеловеческой игры в го, в игры

Atari и в StarCraft — везде, где симулятор может дать вдоволь состояний, действий и наград для обучения.

Наконец, фреймворки — бесплатные инструментарии для построения сетей — распространили идеи. Первое поколение, Caffe, Torch и Theano, вынесло на себе много знаковых статей. Затем появился TensorFlow, которым часто пользовались через Keras, а также CNTK, Caffe 2 и MXNet. Третье поколение — императивное: модель пишут так же, как обычные численные программы на Python; эту идею запустил Chainer, а подхватили PyTorch, Glueon из MXNet и JAX.

Запомним

В нейронных сетях чередуются прямолинейные и изгибающие слои, а обратное распространение ошибки использует цепное правило, перемножая влияния по пути, чтобы подстроить все ручки сразу. Они застопорились примерно с 1995 по 2005 год, потому что вычислений не хватало, а наборы данных были малы, так что побеждали ядерные методы и деревья решений. С 1970 по 2020 год данные выросли в десять миллиардов раз, память — в сто миллионов раз, а вычисления — ещё быстрее, и это дало преимущество глубоким сетям. Дропаут, внимание, Трансформер, языковые модели, состязательные сети и диффузионные модели вызвали всплеск прогресса. Тренировка на тысячах чипов и фреймворки вроде PyTorch сделали всё это быстрым и широко доступным.

Вопросы

1. Что использует обратное распространение ошибки, чтобы сразу для каждой ручки найти, в какую сторону её повернуть?
 - a. Случайные пробы каждой ручки
 - b. Цепное правило, перемножая влияния по стадиям от конца к началу
 - c. Таблицу найденных ранее лучших положений
 - d. Суждение человека о каждом слое
2. Почему ядерные методы и деревья решений обыгрывали нейронные сети примерно в 1995–2005 годах?
 - a. Потому что было доказано, что нейронные сети ошибочны
 - b. Потому что для них было слишком мало памяти
 - c. Потому что вычислений не хватало, а наборы данных были малы, и эти методы тренировались быстрее и давали надёжные гарантии
 - d. Потому что они использовали графические чипы
3. С 1970 по 2020 год данные выросли гораздо сильнее, чем память. К чему это подтолкнуло лучшие методы?
 - a. К моделям, бережливым с памятью, которые тратят много вычислений на настройку ручек
 - b. К моделям, которые хранят каждый пример
 - c. К моделям вообще без ручек
 - d. К меньшим наборам данных

4. Какую проблему внимание решило при переводе?
- a. Сделало ненужными тренировочные данные
 - b. Убрало шум с изображений
 - c. Заменяло графический чип
 - d. Позволило модели указывать назад на нужную часть предложения вместо того, чтобы втискивать всё предложение в фиксированную сводку

Ответы

1. **b)** Обратное распространение ошибки применяет цепное правило от конца сети к началу и так за один проход узнаёт для каждой ручки, в какую сторону её повернуть.
2. **c)** При таком малом количестве данных и вычислений другие инструменты справлялись лучше; им не нужны были недели на тренировку, и они давали предсказуемые результаты с гарантиями.
3. **a)** Это подтолкнуло лучшие методы к моделям, бережливым с памятью, но способным щедро тратить вычисления на настройку своих ручек.
4. **d)** Вместо того чтобы втискивать целое предложение в сводку фиксированного размера, модель с вниманием хранит обучаемый указатель назад на нужную часть.

Разобранные примеры

Один огромный шаг на многих чипах Тренировка распределена на 1024 графических чипа, и каждый чип на каждом шаге берёт пачку из 32 изображений. Сколько всего изображений обрабатывает один шаг?

Каждый чип обрабатывает 32 изображения за шаг.

Все 1024 чипа работают одновременно, так что умножаем.

1024 на 32 — это 32 768.

Это около 32 000 изображений за шаг, и каждый чип загружен работой.

Ответ: 32 768 изображений, около 32 000

Задачи

1. Используя цепное правило так, как его излагает лекция: поворот ручки повышает выход слоя на 4 единицы на единицу поворота, а каждая единица этого выхода понижает потерю на 0,5. Каково влияние ручки на потерю и в какую сторону её должен повернуть градиентный спуск?

Подсказка: Перемножьте влияния по пути, следя за повышениями и понижениями.

Ответ: 4 на минус 0,5 даёт минус 2: каждая единица поворота понижает потерю на 2. Поскольку поворот ручки вверх понижает потерю, градиентный спуск поворачивает её вверх.

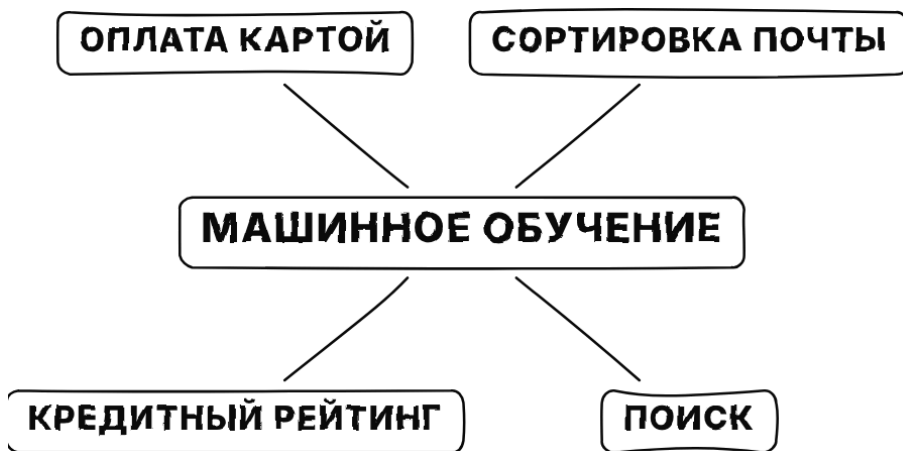
2. Память выросла с 1 килобайта в 1970 году до 100 гигабайт в 2020-м, а наборы данных — примерно со 100 примеров примерно до триллиона. Во сколько раз выросло каждое и что это говорит о хранении целых наборов данных в памяти?

Подсказка: Гигабайт — это миллион килобайт.

Ответ: 100 гигабайт — это 100 миллионов килобайт, так что память выросла в 100 миллионов раз; данные выросли со 100 до триллиона, в 10 миллиардов раз. Данные росли в сто раз быстрее, так что целые наборы данных всё хуже помещаются в память, и методам пришлось стать бережливыми с памятью.

Тихие успехи и настоящие тревоги

Заплатите картой за что-нибудь в интернете, и в то мгновение, пока платёж не прошёл, программа уже решит, не похоже ли это на мошенничество. Отправьте письмо, и машина, весьма вероятно, прочитает рукописный адрес. Ни то ни другое не ощущается как искусственный интеллект. Но и то и другое — машинное обучение, и оба работают уже много лет, в основном незаметно.



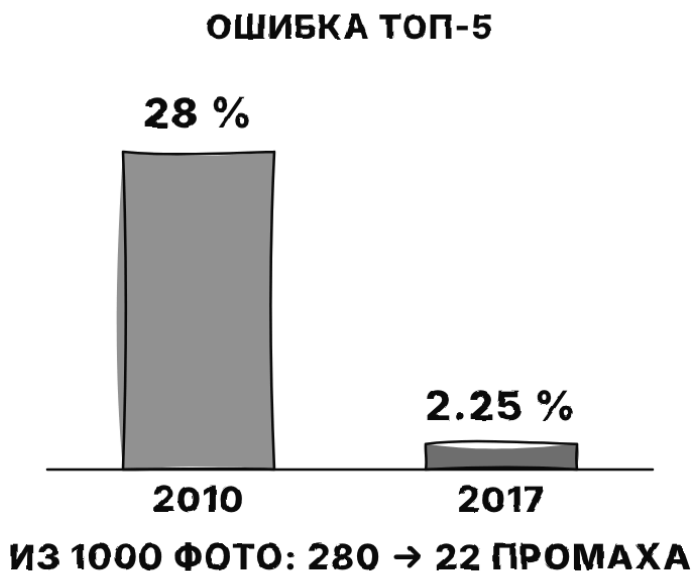
Машинное обучение вне поля зрения

Одна из причин, почему область так быстро движется вперёд, — разделение труда. Одни исследователи делают лучшие инструменты, другие — лучшие модели, и каждая сторона выигрывает от другой. Книга приводит показательный пример. В 2014 году тренировка линейной логистической регрессии, одного из простейших классификаторов, всё ещё была домашним заданием, которое стоило давать новым аспирантам по машинному обучению в Университете Карнеги — Меллона. Сегодня это занимает меньше десяти строк кода и вполне по силам любому программисту.

Некоторые успехи давние. Машины для сортировки почты, читающие рукописные адреса, используются с 1990-х годов, и именно оттуда взялось знаменитое собрание рукописных цифр. Машины читают чеки при зачислении в банк и оценивают, кредитоспособны ли заявители на кредит. Платежи автоматически проверяются на мошенничество, и это основа платёжных систем вроде PayPal, Stripe, Visa и Mastercard. Шахматные программы конкурентоспособны уже десятилетиями. А машинное обучение питает поиск, рекомендации, персонализацию и ранжирование по всему интернету. Порядок видео в вашей ленте рекомендаций — один из этих тихих результатов. Короче говоря, машинное обучение повсюду, хотя часто скрыто.

Искусственный интеллект вышел в центр внимания лишь недавно, потому что начал решать задачи, когда-то считавшиеся неразрешимыми, — задачи, с которыми обычные потребители сталкиваются напрямую, и многими из этих успехов мы обязаны глубокому обучению. Голосовые помощники вроде Siri, Alexa и помощника Google с приемлемой точностью отвечают на устные

просьбы, от включения света до записи к парикмахеру, и это, пожалуй, самый заметный признак искусственного интеллекта в повседневной жизни. Их ключевой ингредиент, распознавание речи, улучшился настолько, что в некоторых применениях сравнялся с людьми.



Ошибки на ImageNet с 2010 по 2017 год

Распознавание объектов продвинулось не меньше. На проверке ImageNet модель считается ошибившейся на фото, если правильного ответа нет среди её пяти лучших догадок, а доля таких промахов — её ошибка по пяти лучшим. В 2010 году команда из NEC Labs и Иллинойского университета достигла ошибки по пяти лучшим в двадцать восемь процентов. К 2017 году она упала до двух целых двадцати пяти сотых процента. Из

тысячи фотографий это падение с двухсот восьмидесяти промахов примерно до двадцати двух, более чем в двенадцать раз меньше. Похожие поразительные результаты получены в определении птиц по пению и в диагностике рака кожи.

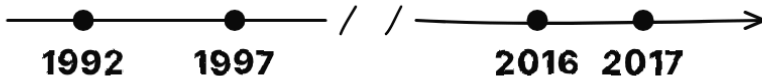
Игры когда-то были мерилom человеческих способностей, и машины покорили эту шкалу. TD-Gammon научилась нардам с помощью одной из форм обучения с подкреплением. В шахматах позиций и ходов гораздо больше, и Deep Blue обыграл Гарри Каспарова благодаря массивным параллельным вычислениям, специальному оборудованию и эффективному поиску по дереву возможных партий. Го ещё труднее, с ещё более огромным числом позиций, и AlphaGo достигла человеческого уровня в 2015 году, соединив глубокое обучение со способом выборки вероятных продолжений партии, называемым выборкой по дереву методом Монте-Карло. Покер добавляет другую трудность: состояние велико и видно лишь отчасти, ведь карт других игроков не видно, и всё же программа под названием Libratus превзошла людей благодаря тщательно выстроенным стратегиям.

ИГРЫ, ВЫИГРАННЫЕ МАШИНАМИ

ШАХМАТЫ

НАРДЫ

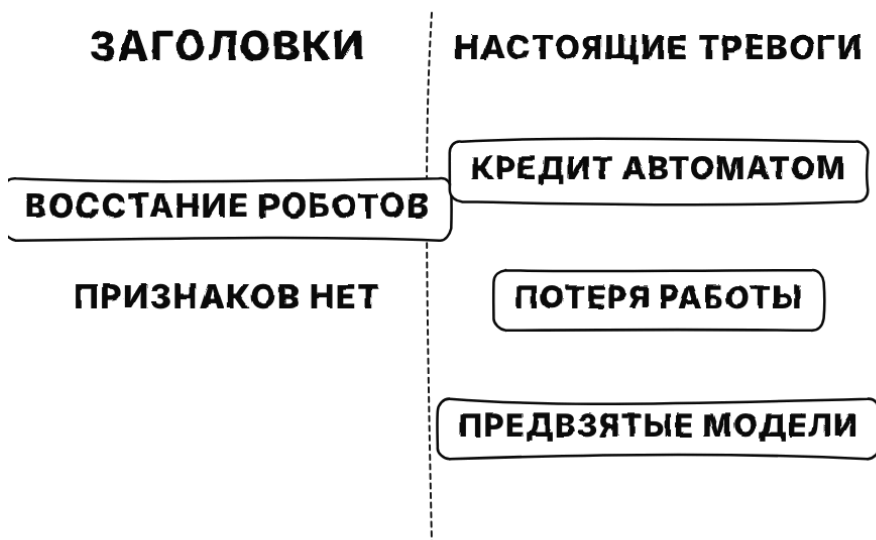
ГО ПОКЕР



Игры, в которых машины научились побеждать

Беспилотные машины — ещё один признак прогресса. Полная автономность пока недостижима, но компании вроде Tesla, NVIDIA и Waymo выпускают продукты, которые часть времени едут сами. Полная автономность трудна, потому что хорошее вождение требует воспринимать, рассуждать и следовать правилам; сейчас глубокое обучение в основном отвечает за восприятие, а остальное инженеры настраивают вручную. Кроме того, робототехника, логистика, вычислительная биология, физика частиц и астрономия обязаны некоторыми из своих самых впечатляющих недавних успехов хотя бы отчасти машинному обучению, которое становится повседневным инструментом инженеров и учёных.

Популярные статьи часто пугают апокалипсисом искусственного интеллекта, так называемой сингулярностью, когда машины обретут сознание и станут решать нашу судьбу без своих программистов. В какой-то мере искусственный интеллект уже напрямую влияет на жизнь людей: кредитоспособность оценивают автоматически, автопилоты выполняют большую часть навигации многих транспортных средств, а решения об освобождении под залог опираются на статистические данные. Если же о пустяках — можно попросить голосового помощника включить кофеварку.



Страхи из заголовков и настоящие тревоги

Насчёт апокалипсиса авторы успокаивают. Во-первых, сегодняшние системы проектируют, тренируют и развёртывают

под конкретные цели; каким бы общим ни казалось их поведение, внутри — сочетание правил, эмпирических приёмов и статистических моделей. Во-вторых, сейчас просто нет инструментов для сильного искусственного интеллекта, который мог бы совершенствовать себя, рассуждать о себе и перестраивать собственное устройство, решая общие задачи.

Насущный вопрос, говорят они, — как искусственный интеллект используется в повседневной жизни. Многие рутинные задачи, которые сейчас выполняют люди, можно автоматизировать и автоматизируют. Сельскохозяйственные роботы, вероятно, снизят затраты органических фермеров, но они же автоматизируют уборку урожая, а поскольку такой ручной труд даёт значительную часть занятости во многих странах, этот этап промышленной революции может иметь глубокие последствия для больших слоёв общества. А статистические модели, применяемые без должной осторожности, могут быть предвзятыми по признаку расы, пола или возраста, что вызывает настоящие опасения насчёт справедливости, когда от них зависят решения с серьёзными последствиями. Решение о кредите или работе, принятое моделью, обученной на вчерашних несправедливых решениях, может автоматически их повторить, как мы видели на примере с наймом. Пользоваться этими алгоритмами нужно осторожно, и авторы считают это гораздо более срочным, чем любой злонамеренный сверхинтеллект.

Запомним

Лучшие инструменты превратили то, что в 2014 году было домашним заданием для аспирантов, в меньше чем десять строк кода. Машинное обучение давно тихо работает в сортировке почты, чтении чеков, выявлении мошенничества и ранжировании в интернете. Глубокое обучение принесло заметные скачки: голосовые помощники, распознавание речи, распознавание изображений с ошибками, сокращёнными с двадцати восьми процентов примерно до двух, и победы в нардах, шахматах, го и покере. Беспилотные машины и многие науки продвигаются вместе с ним, хотя полной автономности пока нет. Настоящие тревоги — не сознательная машина, а потерянные рабочие места и предвзятые автоматические решения, которые требуют осторожного применения.

Вопросы

1. Что означает ошибка по пяти лучшим в двадцать восемь процентов?
 - a. Первая догадка модели неверна для 28 фото из 100
 - b. Для 28 фото из 100 правильного ответа нет среди пяти лучших догадок модели
 - c. Модели нужно пять попыток, чтобы достичь точности 28 процентов
 - d. На 28 процентах фото по пять объектов

2. Почему, согласно лекции, полная автономность машин пока недостижима?
 - a. Потому что вождение требует воспринимать, рассуждать и следовать правилам, а глубокое обучение в основном отвечает только за восприятие
 - b. Потому что машины не могут везти графические чипы
 - c. Потому что нет данных о дорогах
 - d. Потому что законы это везде запрещают
3. Почему авторы сегодня не тревожатся о сознательном, строящем козни ИИ?
 - a. Потому что системы ИИ слишком медленные
 - b. Потому что ИИ запрещено участвовать в важных решениях
 - c. Потому что сегодняшние системы строят под конкретные цели, а инструментов для самосовершенствующегося общего интеллекта нет
 - d. Потому что ИИ не умеет играть в игры
4. Какую тревогу авторы считают самой насущной?
 - a. Сингулярность
 - b. Нехватку памяти у компьютеров
 - c. Слишком хорошую игру роботов в шахматы
 - d. Автоматизацию рутинной работы и предвзятые решения небрежно применяемых моделей

Ответы

1. b) Модель считается ошибившейся на фото, если правильного ответа нет среди её пяти лучших догадок, а доля таких промахов —

ошибка по пяти лучшим.

2. а) Хорошее вождение требует воспринимать, рассуждать и следовать правилам; глубокое обучение в основном отвечает за восприятие, а остальное инженеры настраивают вручную.

3. с) Системы проектируют под конкретные цели, и сейчас просто нет инструментов для сильного искусственного интеллекта, который мог бы совершенствоваться и перестраивать себя.

4. d) Насущный вопрос — как ИИ используется в повседневной жизни: автоматизация рутинной работы и модели, применяемые без осторожности, предвзятые по признаку расы, пола или возраста.

Разобранные примеры

Насколько улучшилось распознавание изображений На проверке ImageNet ошибка по пяти лучшим составляла 28 процентов в 2010 году и 2,25 процента в 2017-м. Сколько это промахов из 1000 фото в каждом году и примерно во сколько раз меньше?

28 процентов от 1000 фото — это 280 промахов.

2,25 процента от 1000 фото — это 22,5, около 22 промахов.

Разделите старое число на новое: 280, делённое на 22,5, — около 12,4.

Значит, промахов стало примерно в двенадцать раз меньше.

Ответ: 280 промахов против примерно 22; более чем в двенадцать раз меньше

Задачи

1. У модели ошибка по пяти лучшим 5 процентов на 2000 тестовых фото. Для скольких фото правильного ответа нет среди

её пяти лучших догадок?

Подсказка: Возьмите 5 процентов от 2000.

Ответ: 5 процентов от 2000 — это 100 фото.

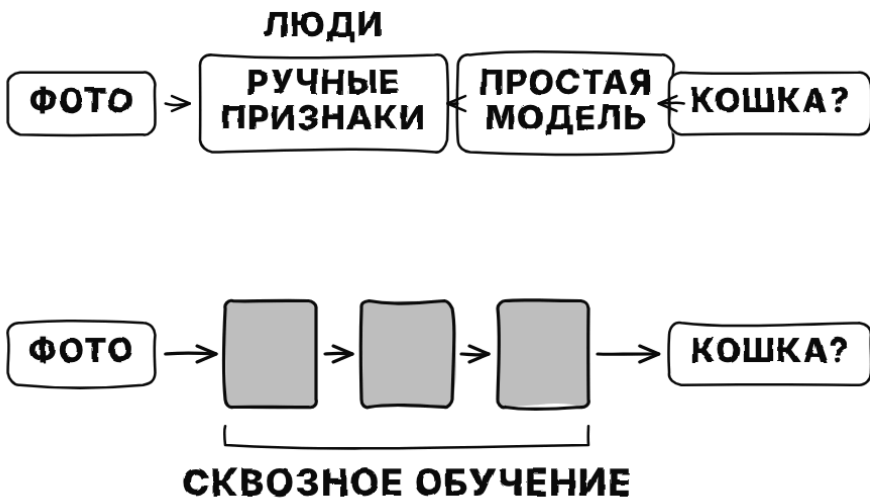
2. Объясните на примере логистической регрессии из книги, как лучшие инструменты изменили круг людей, которые могут пользоваться машинным обучением.

Подсказка: Сравните 2014 год с сегодняшним днём.

Ответ: В 2014 году тренировка логистической регрессии была домашним заданием, достойным новых аспирантов; сегодня, благодаря инструментам, созданным системными исследователями, это занимает меньше десяти строк кода, так что справится любой программист. Разделение труда между создателями инструментов и создателями моделей открыло область гораздо большему числу людей.

Что делает его глубоким

Представьте, что учите компьютер замечать кошек так, как это делали пятнадцать лет назад. Сначала умные люди пишут программу, которая находит на фото края, углы и маленькие участки текстуры. Её находки, список чисел, поступают в простую обучающуюся модель, которой остаётся лишь решить: «кошка» или «не кошка». Если сделанный вручную первый шаг упустит что-то важное, ничто из того, чему модель научится потом, этого не вернёт. Глубокое обучение убирает это разделение на ручную переднюю часть и выученную заднюю, и именно к этому, в конечном счёте, вело всё это введение.

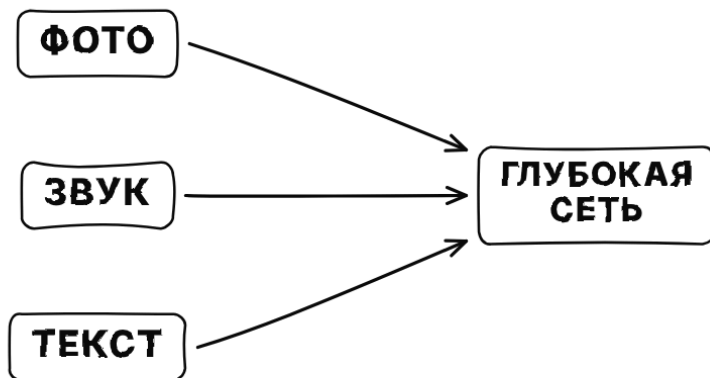


Ручная обработка или сквозное обучение

Глубокое обучение — это часть машинного обучения, занимающаяся моделями, построенными из нейронных сетей со многими слоями. Оно глубокое именно в этом смысле: его модели выучивают много слоёв преобразований, каждое из которых перестраивает полученное и передаёт дальше. Это может звучать как узкое определение, но оно породило головокружительное разнообразие моделей, приёмов, типов задач и применений. В каком-то смысле у любого машинного обучения несколько слоёв вычислений, и первый из них — какая-то обработка признаков. Глубокое обучение выделяется тем, что операции на каждом из его многих слоёв представления выучиваются совместно, все вместе, из данных.

Поэтому глубокое обучение превосходит именно там, где традиционные методы давали сбой: при обучении на сыром звуке, на сырых значениях пикселей изображений или при переводе предложений любой длины на другой язык. Эти многослойные модели справляются с низкоуровневыми данными восприятия так, как прежние инструменты не могли. Самое важное, что объединяет методы глубокого обучения, — сквозная тренировка: вместо того чтобы собирать систему из частей, каждую из которых настраивают отдельно, вы строите всю систему и настраиваете её целиком.

Компьютерное зрение показывает эту переменную. Годами учёные отделяли конструирование признаков — ручную разработку способов превращать изображения в числа — от построения обучающейся модели. Два спроектированных вручную метода, детектор краёв Кэнни и извлекатель признаков SIFT Лоу, больше десяти лет царили как стандартные способы превращать изображение в вектор признаков. В те времена решающей частью применения машинного обучения было придумывать хитрые ручные преобразования, с которыми могла бы справиться неглубокая модель. Но человеческая изобретательность может зайти лишь так далеко против алгоритма, который постоянно автоматически оценивает миллионы вариантов. Когда глубокое обучение взяло верх, ручные извлекатели признаков сменились автоматически настраиваемыми фильтрами, и точность выросла.

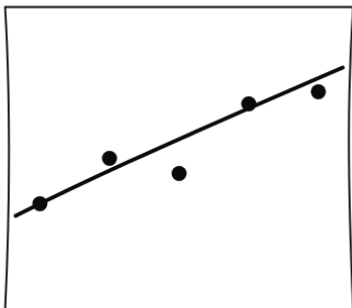


Один вид сети для всех областей

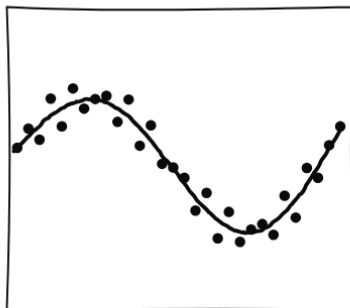
Так что глубокое обучение заменяет не только неглубокую модель в конце старого конвейера, но и трудоёмкое конструирование признаков перед ней. А поскольку большая часть этой предобработки была свойственна одной области, глубокое обучение убрало многие границы между компьютерным зрением, распознаванием речи, обработкой языка, медицинской информатикой и другими областями, предложив один общий набор инструментов для очень разных задач. Поэтому голосового помощника, который вас слышит, и чат-бота, который вам отвечает, можно построить из одного и того же семейства моделей.

Есть и ещё один сдвиг — от параметрических моделей к полностью непараметрическим. Когда данных мало, полезная модель должна опираться на упрощающие допущения о действительности — несколько параметров, описывающих фиксированную форму. Когда данных много, такие допущения могут уступить место моделям, которые свободнее следуют данным. Книга сравнивает это с физикой середины прошлого века: когда появились компьютеры, физики смогли численно моделировать уравнения поведения электронов, вместо того чтобы решать упрощённые приближения вручную. Модели стали гораздо точнее, хотя часто их стало труднее истолковать.

МАЛО: ДОПУСКАТЬ



МНОГО: СЛЕДОВАТЬ



Меньше допущений при обилии данных

Глубокое обучение отличается от прежней работы и своим отношением. Оно принимает решения хорошие, а не доказуемо лучшие, берётся за задачи оптимизации со множеством холмов и долин и готово пробовать, прежде чем доказывать. Этот эмпиризм вместе с быстрым притоком талантов дал быстрый прогресс практических методов, хотя иногда ценой повторного изобретения инструментов, существовавших десятилетиями. А сообщество гордится тем, что делится инструментами между университетами и компаниями, свободно выпуская библиотеки, модели и обученные сети; в том же духе собственные блокноты книги свободны для использования и распространения.

Итог книги связывает все нити. Машинное обучение изучает, как компьютерные системы используют опыт, обычно данные, чтобы становиться лучше в конкретных задачах, соединяя идеи статистики, интеллектуального анализа данных и оптимизации, и часто служит средством построения искусственного интеллекта. Обучение представлений, один из видов машинного обучения, — это автоматический поиск правильного способа представлять данные, а глубокое обучение — это многоуровневое обучение представлений, через много выученных слоёв. Его недавний прогресс запустили обилие данных от дешёвых датчиков и приложений масштаба интернета, огромные успехи в вычислениях, в основном благодаря графическим чипам, и эффективные фреймворки, которые сильно облегчают проектирование и оптимизацию целых систем. На этом введение завершается: от программ, следующих фиксированным правилам, к системам, которые выучивают собственные слои описания из данных.

Запомним

Глубокое обучение — это машинное обучение с многослойными нейронными сетями, в которых каждый слой представления выучивается совместно из данных. Его отличительная черта — сквозная тренировка: вся система настраивается вместе, а не по отдельности настраиваемым частям. Оно заменило и неглубокие модели, и ручное конструирование признаков, вроде детекторов краёв, и объединило зрение, речь и язык одним набором инструментов. При обилии данных оно может отказаться от упрощающих допущений и предпочитает пробовать на практике, а не сначала доказывать. Его подъём опирается на обилие данных, быстрые графические чипы и бесплатные эффективные фреймворки.

Вопросы

1. У многих систем машинного обучения несколько стадий. Чем отличается глубокое обучение?
 - a. Его стадии работают на графических чипах
 - b. Операции на всех его многих слоях выучиваются совместно из данных
 - c. У него ровно три слоя
 - d. Ему не нужна никакая подготовка данных

2. Почему автоматически настраиваемые фильтры заменили спроектированные вручную извлекатели вроде SIFT?
 - a. Потому что ручные извлекатели работали слишком медленно
 - b. Потому что алгоритм, оценивающий миллионы вариантов, может превзойти человеческую изобретательность, и точность выросла
 - c. Потому что SIFT перестал работать на новых камерах
 - d. Потому что фильтрам не нужна тренировка
3. Когда модель может отказаться от упрощающих допущений и свободнее следовать данным?
 - a. Когда данных мало
 - b. Когда модель маленькая
 - c. Когда данных много
 - d. Когда задача — о физике
4. Что, по сравнению в книге, физика выиграла и потеряла, перейдя к компьютерному моделированию?
 - a. Выиграла в скорости и потеряла в точности
 - b. Выиграла в точности, но часто потеряла в простоте истолкования
 - c. Выиграла в истолковании, но потеряла в точности
 - d. Потеряла и в точности, и в истолковании

Ответы

1. b) Глубокое обучение выделяется тем, что операции на каждом из его многих слоёв представления выучиваются совместно, все вместе, из данных.

2. b) Человеческая изобретательность может пойти лишь так далеко против алгоритма, автоматически оценивающего миллионы вариантов; настроенные фильтры дали лучшую точность.

3. c) Когда данных мало, модели опираются на упрощающие допущения; когда данных много, эти допущения могут уступить место моделям, которые следуют данным.

4. b) Модели стали гораздо точнее, хотя часто их стало труднее истолковать.

Задачи

1. Объясните своими словами, что такое сквозная тренировка и почему она может обыграть конвейер из отдельно настроенных частей.

Подсказка: Вспомните детектор кошек с ручным искателем краёв впереди.

Ответ: Сквозная тренировка строит всю систему и настраивает её целиком на данных, вместо того чтобы настраивать каждую часть отдельно. В конвейере ручной первый шаг может выбросить что-то важное, и никакое последующее обучение этого не вернёт; когда все слои выучиваются вместе, ранние слои учатся сохранять именно то, что нужно поздним.

2. Подведите в трёх предложениях итог трём условиям, которым книга приписывает недавний прогресс глубокого обучения.

Подсказка: Данные, железо и программы.

Ответ: Обилие данных от дешёвых датчиков и приложений масштаба интернета дало моделям достаточно примеров. Успехи в вычислениях, в основном графические чипы, сделали тренировку больших сетей возможной. Эффективные фреймворки сильно

облегчили проектирование, реализацию и оптимизацию целых систем.