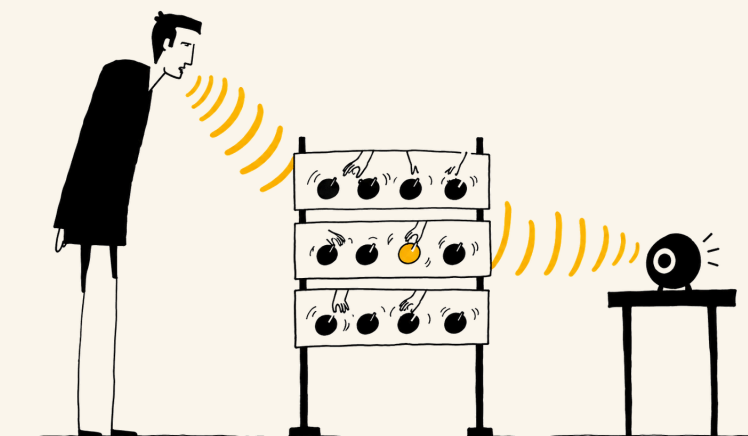


Погружение в глубокое обучение



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

Магазин, который никогда не учится

Программирование примерами

Хорошие данные, плохие данные

Пробные варианты и настоящий экзамен

Угадать, сколько

Что это?

Теги, рейтинги и ваша лента

Когда важен порядок

Учиться без ответов

Учиться на деле

Старые корни

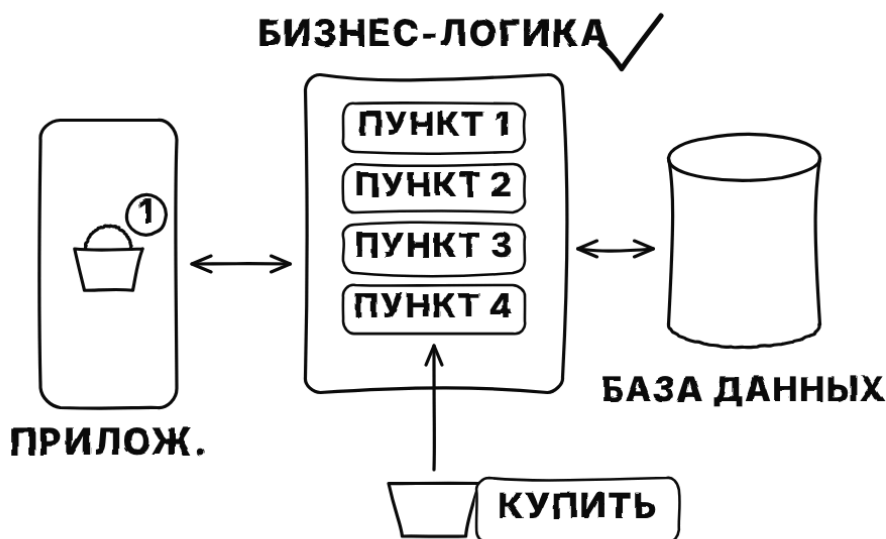
Великое ускорение

Повсюду, но почти незаметно

Что делает его глубоким

Магазин, который никогда не учится

Откройте приложение магазина, нажмите «в корзину» у пары носков и посмотрите, как маленькая цифра на корзине прыгает до единицы. Повторите это завтра или тысячу раз — она прыгнет точно так же. Кто-то давным-давно написал правило для этого нажатия, и от того, что вы им пользуетесь, правило не меняется.



Магазин, построенный из правил

Большинство программ, с которыми вы встречаетесь за день, устроены именно так: как правила, написанные людьми. Книга представляет команду у маркерной доски, которая проектирует

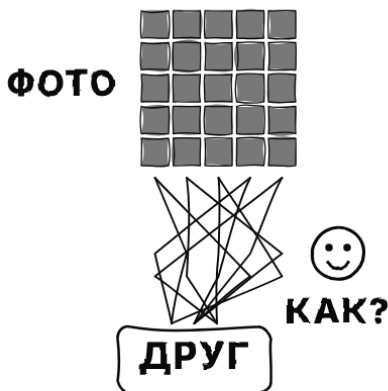
интернет-магазин. Покупатели будут видеть его в браузере или в приложении на телефоне; база данных будет помнить каждого покупателя и каждую покупку; а посередине находится то, что авторы называют бизнес-логикой, — мозг магазина, набор правил о том, что делать в каждой ситуации. Затем команда выискивает неудобные случаи, например когда кто-то нажимает «купить» с пустой корзиной, и пишет правило на каждый. После некоторого тестирования всё работает, и магазин может открыться раньше, чем придёт хоть один настоящий покупатель. Авторы этим восхищаются и добавляют прямой совет: если вы можете написать правила, которые верны каждый раз без исключения, машинное обучение вам не нужно.

А теперь дайте той же команде задачи потруднее. Предсказать завтрашнюю погоду по спутниковым снимкам и измерениям за прошлые недели. Ответить на вопрос, набранный обычными словами. Найти на фото каждого человека и обвести каждого. Предложить покупателю вещи, которые ему очень понравятся, но которые он никогда не нашёл бы сам, листая каталог. Именно эту последнюю задачу пытается решить ваша лента рекомендаций каждый раз, когда вы её открываете. Даже лучшим программистам было бы трудно написать правила для любой из этих задач.

ЦЕЛЬ ДВИЖЕТСЯ



ЗАПУТАНО

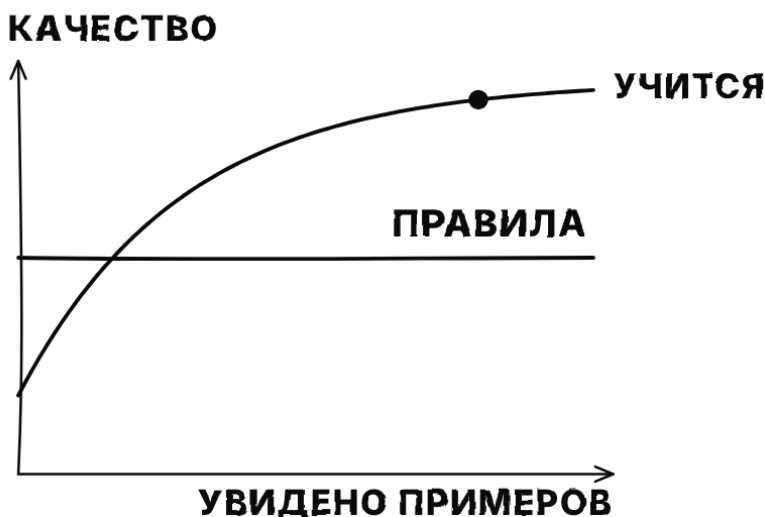


Две причины, почему правила не работают

Причин две. Иногда правильный ответ всё время смещается, так что застывшие правила устаревают, и хорошее решение должно подстраиваться, когда меняется мир. Спам-фильтр живёт с этим постоянно: стоит ему поймать один вид мусорных писем, как отправители меняют формулировки. А иногда связь между тем, что поступает на вход, и тем, что должно выйти, слишком запутана, чтобы её записать. Вы мгновенно узнаете лицо друга в толпе, но никогда не смогли бы перечислить шаги, которые для этого проделал ваш мозг.

Машинное обучение — это наука о методах, которые учатся на опыте, обычно на множестве записанных примеров, и становятся лучше, чем больше видят. Правила магазина остаются

прежними, сколько бы покупателей через него ни прошло, пока программист не решит их переписать. Обучающаяся система улучшается на тех самых примерах, которые видит. Когда голосовой помощник начинает лучше понимать, как люди говорят на самом деле, никто не писал правила для каждого голоса; он научился на большом объёме записанной речи. Книга посвящена одному мощному семейству этих методов, которое называется глубоким обучением; за ним стоит большая часть недавних успехов в компьютерном зрении, понимании языка, медицине и изучении генов. Чат-бот, которому вы пишете, — один из результатов: никто не писал его ответы в виде правил; он усвоил закономерности языка из огромного количества текста.



Застывшие правила против обучения

Запомним

Обычные программы — это правила, написанные людьми, и ведут они себя одинаково, сколько бы ими ни пользовались. Когда люди могут предвидеть каждый случай и каждый раз решать его верно, правил достаточно. Некоторые задачи не поддаются правилам, потому что ответ всё время меняется или потому что путь к нему слишком запутан, чтобы его записать. Машинное обучение — о методах, которые улучшаются с опытом, а глубокое обучение — то семейство таких методов, которому учит эта книга.

Вопросы

1. Почему цифра на корзине прыгает одинаково, сколько бы раз вы ни нажимали?
 - a. Приложение ещё собирает примеры, чтобы учиться
 - b. Оно следует заранее написанному правилу, которое не меняется от использования
 - c. Телефон сразу забывает каждое нажатие
 - d. Привычки покупателей меняются слишком быстро, чтобы им научиться
2. Когда, по словам книги, машинное обучение вам не нужно?
 - a. Когда задача связана с приложением на телефоне
 - b. Когда данных пока мало

- c. Когда вы можете написать правила, верные каждый раз без исключения
 - d. Когда покупатели ещё не пришли
- 3. Спам-фильтр всё время пропускает новые мусорные письма. Какую из двух причин неудачи правил это показывает?
 - a. Правильный ответ всё время смещается, потому что отправители меняют формулировки
 - b. Связи между письмом и ответом невозможно научиться
 - c. Программисты забыли проверить случай с пустой корзиной
 - d. Фильтр увидел слишком много примеров
- 4. Чем обучающаяся система отличается от правил магазина?
 - a. Она никогда не ошибается
 - b. Её пишут более сильные программисты
 - c. Ей вообще не нужны примеры
 - d. Она улучшается на примерах, которые видит

Ответы

- 1. **b)** Кто-то давным-давно написал правило для этого нажатия, и от того, что вы им пользуетесь, правило не меняется.
- 2. **c)** Если вы можете написать правила, которые верны каждый раз без исключения, машинное обучение вам не нужно.
- 3. **a)** Иногда правильный ответ всё время смещается: стоит фильтру поймать один вид мусорных писем, как отправители меняют формулировки.

4. d) Правила магазина остаются прежними, пока программист их не перепишет, а обучающаяся система улучшается на тех самых примерах, которые видит.

Задачи

1. Для каждой из четырёх трудных задач (завтрашняя погода, ответ на набранный вопрос, контур вокруг каждого человека на фото, подбор вещей, которые покупатель сам никогда не нашёл бы) скажите, какая причина мешает правилам: ответ всё время меняется или путь к нему слишком запутан, чтобы его записать.

Подсказка: Спросите, смещается ли правильный ответ со временем или вы просто не смогли бы перечислить шаги.

Ответ: Погода и вкусы покупателей всё время меняются, так что застывшие правила устаревают, а погода к тому же запутана. Ответы на обычные вопросы и поиск людей на фото — запутанный случай: мы делаем это легко, но не можем перечислить шаги. У нескольких задач есть обе трудности.

2. Объясните своими словами, почему для покупателя, нажавшего «купить» с пустой корзиной, правило вполне подходит.

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

Ответ: Команда может предвидеть этот случай ещё до прихода покупателей и написать реакцию, верную каждый раз, например сообщение о том, что корзина пуста. Когда правила могут быть верны каждый раз, машинное обучение не нужно.

Программирование примерами

Скажите «Hey Siri» или «OK Google» через шумную комнату и посмотрите, как просыпается телефон. Среди звуков телевизора, чайника и всех разговоров он уловил одну фразу. А теперь представьте, что пишете такую программу сами, когда у вас есть только компьютер и пустой экран. С чего вообще начать?



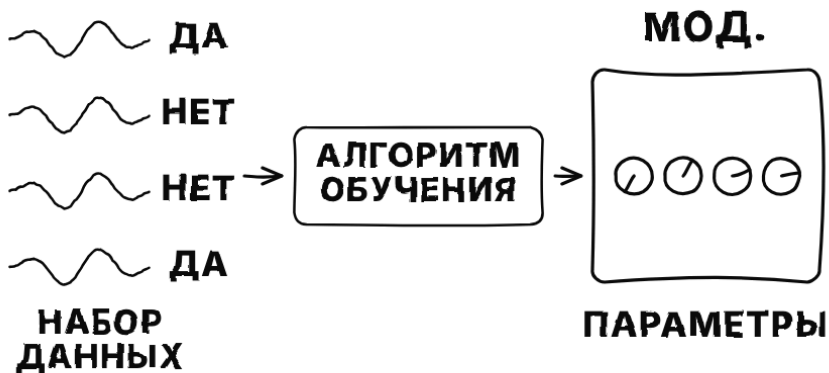
На входе звук, на выходе «да» или «нет»

Авторы рассказывают небольшую сценку, придуманную для наглядности. По дороге за кофе один из них будит телефон словами «Hey Siri» и спрашивает дорогу до кофейни. Телефон

записывает его слова, понимает, что ему нужен маршрут, открывает карты и показывает несколько путей с предсказанным временем в пути рядом с каждым. За несколько секунд поработали несколько отдельных обученных программ. Когда вы просите голосового помощника поставить таймер, работает та же цепочка: одна программа слушает фразу пробуждения, другая записывает ваши слова, третья понимает, что вы имели в виду.

Возьмём только первую задачу — заметить слово пробуждения. Микрофон превращает звук в длинный поток чисел, десятки тысяч в секунду. По этому потоку программа должна ответить только «да» или «нет»: была ли произнесена фраза? Никто не знает правила, которое делало бы это надёжно, и книга признаёт, что её авторы тоже такого не знают. Именно поэтому и применяют машинное обучение.

Хитрость вот в чём: мы не можем объяснить, как распознать «Алеха», но сами её распознаём. Поэтому люди прослушивают очень много коротких записей и помечают каждую: «да» или «нет». Такое собрание называют набором данных, а пометки — метками. Вместо того чтобы писать распознаватель, мы пишем гибкую программу с множеством настраиваемых величин, которые называются параметрами, и даём набору данных решить, какими они должны быть. Книга изображает параметры как ручки на машине. Когда ручки выставлены и больше не трогаются, программу называют моделью, а программа, которая по данным выбирает положения ручек, — это алгоритм обучения.



Метки, ручки и ученик

Одна и та же машина с ручками в одном положении говорит «да» на «Alexa»; в другом положении она могла бы говорить «да» только на «Apricot», потому что эти задачи похожи. Превращать фотографии в подписи или переводить с английского на китайский — совсем другая задача, и для неё может понадобиться машина другого рода.

Опыт

Попросите друга сказать слово шёпотом, невнятно и криком из соседней комнаты. Вы узнаете его каждый раз, но так и не сумеете записать, что делает его именно этим словом, а не похожим. Этот разрыв между умением что-то делать и умением сформулировать правило — причина, по которой мы учимся на примерах.

Когда ручки выставлены случайно, модель ничего не распознаёт. Нужное положение находят с помощью тренировки, и книга изображает её как цикл. Начните с модели, которая не делает ничего полезного, покажите ей несколько размеченных записей, чуть подкрутите ручки, чтобы на них она работала немного лучше, и повторяйте на новых примерах, пока она не станет очень хорошей. Авторы называют это программированием с помощью данных. Детектор кошек можно «запрограммировать» тем же способом, показывая ему много картинок с кошками и собаками, пока он не начнёт давать уверенное «да» для кошек, уверенное «нет» для собак и что-то посередине, когда не уверен. Ваш спам-фильтр научился так же: ему показали много писем с пометками «спам» или «не спам», и его ручки встали там, где эти два вида разделяются. Глубокое обучение, предмет этой книги, — один из нескольких популярных способов это делать.

Запомним

Несколько секунд с телефоном могут задействовать сразу несколько обученных программ. Никто не может написать правила, чтобы найти слово пробуждения в сыром звуке, но люди способны его узнать, поэтому они размечают много примеров. Затем гибкую программу с настраиваемыми ручками, её параметрами, тренируют на этих примерах, круг за кругом, пока она не начнёт справляться с задачей. Это программирование с помощью данных, а глубокое обучение — один из способов его делать.

Вопросы

1. Почему люди могут помочь компьютеру выучить слово пробуждения, для которого сами никогда не написали бы правила?
 - a. Потому что они сами узнают слово и могут пометить записи «да» или «нет»
 - b. Потому что они могут записать, как выглядит звук в виде чисел
 - c. Потому что телефон уже знает это слово с момента выпуска
 - d. Потому что слово всегда произносят одинаково
2. Что обозначают ручки на картинке из книги?
 - a. Метки на записях

- b. Настраиваемые величины — параметры, которые определяют, как ведёт себя программа
 - c. Кнопки, которые пользователь нажимает на телефоне
 - d. Число записей в наборе данных
3. Почему машина одного рода может справиться и с «Аlexа», и с «Apricot», но не с подписями к фотографиям?
- a. Для подписей нужен больший набор данных, чем может использовать любая машина
 - b. Фотографии нельзя разметить
 - c. Два слова пробуждения — похожие задачи, а подписи к фотографиям — совсем другая задача
 - d. В «Apricot» больше букв, чем в «Аlexа»
4. Что происходит на каждом круте тренировки?
- a. Все ручки снова выставляют случайно
 - b. Программист пишет ещё одно правило
 - c. Набор данных выбрасывают и собирают заново
 - d. Ручки чуть подкручивают, чтобы модель немного лучше работала на нескольких размеченных примерах

Ответы

1. **a)** Мы не можем объяснить, как узнавать слово, но сами его узнаём, поэтому помечаем много записей, на которых программа учится.
2. **b)** Книга изображает параметры, настраиваемые величины программы, как ручки на машине.
3. **c)** Задачи с двумя словами пробуждения похожи, так что хватит другого поворота тех же ручек; подписи к фото — совсем другая задача, и для неё может понадобиться другая машина.

4. d) Покажите модели несколько размеченных записей, чуть подкрутите ручки, чтобы на них она работала немного лучше, и повторите.

Задачи

1. Расставьте шаги тренировки по порядку: повторять, пока модель не станет хорошей; начать с модели, которая не делает ничего полезного; подкрутить ручки; показать ей несколько размеченных примеров. Почему первая модель бесполезна?

Подсказка: Чему научилась модель до того, как увидела хоть один пример?

Ответ: Начать с модели, которая не делает ничего полезного, показать ей несколько размеченных примеров, подкрутить ручки, чтобы на них она работала лучше, и повторять, пока она не станет хорошей. Первая модель бесполезна, потому что её ручки выставлены случайно и она ещё ничему не научилась.

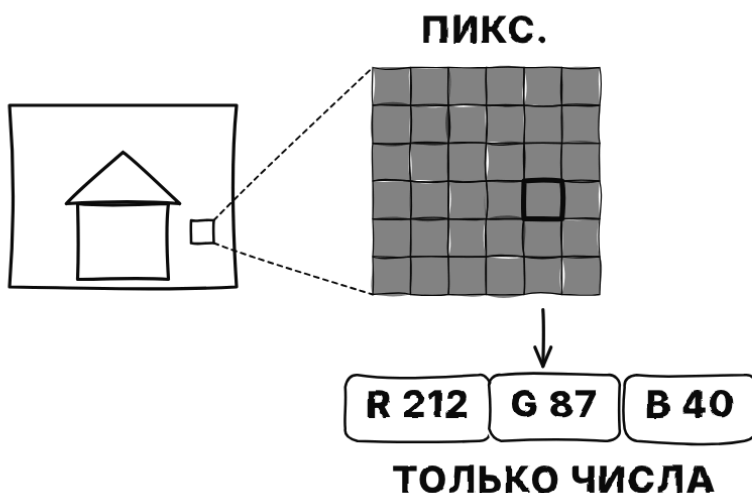
2. Объясните своими словами, почему авторы называют тренировку «программированием с помощью данных».

Подсказка: Кто определяет, как ведёт себя готовая программа?

Ответ: Поведение программы никто не пишет правило за правилом. Его задают, показывая программе много размеченных примеров и позволяя ручкам встать туда, где программа работает хорошо, так что примеры делают ту работу, которую сделали бы написанные вручную правила.

Хорошие данные, плохие данные

Увеличьте фото на телефоне до предела, и картинка распадётся на крошечные цветные квадратики. Каждый квадратик — всего три показания яркости: одно для красного, одно для зелёного, одно для синего. Для обучающейся программы ваш снимок из отпуска — не более чем длинный список таких чисел. Это и есть сырьё, с которым работает любая обучающаяся машина.



Фотография — это только числа

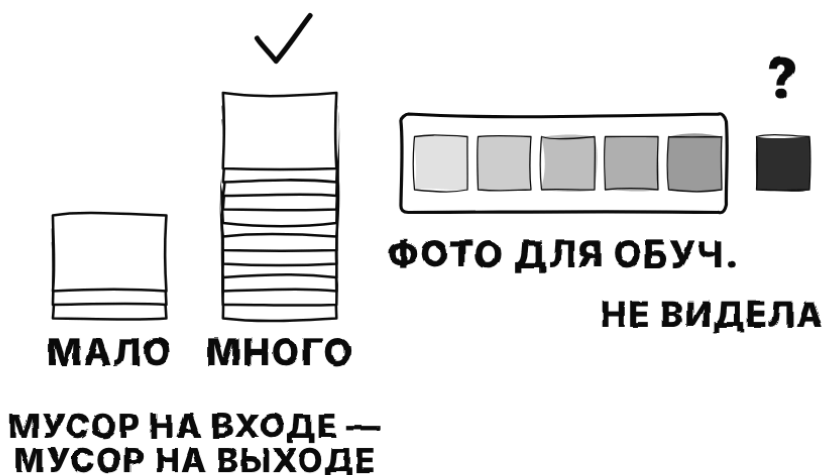
Книга называет четыре составляющие, которые есть у каждой задачи машинного обучения, какой бы она ни была. Первая —

данные, на которых учится машина. Вторая — модель, программа, усеянная ручками, которая превращает эти данные в ответ. Третья — способ оценить, насколько хорошо или плохо работает модель. Четвёртая — метод, который крутит ручки так, чтобы оценка улучшалась. Следующие части посвящены последним трём; здесь речь о данных.

Данные — это груда примеров, и каждый пример нужно превратить в числа, прежде чем машина сможет им воспользоваться. Фото превращается в показания своих пикселей, и даже маленькая квадратная фотография даёт уже намного больше ста тысяч чисел. Больничная карта превращается в возраст пациента, пульс, другие болезни, лекарства и недавние операции, всё записанное числами, а предсказать нужно, проживёт ли пациент следующий месяц. Ваш спам-фильтр видит каждое письмо так же — как набор чисел, который его описывает, а ответ, который он должен дать, — «спам» или «не спам».

Проще всего, когда каждый пример приходит в виде одинакового числа показаний, как фотографии, снятые одним и тем же микроскопом. В жизни всё запутаннее. Картинки из интернета бывают любого размера и формы; можно обрезать их все до одного размера, но всё отрезанное теряется. С текстом ещё хуже: один отзыв в магазине говорит только «полный отстой!», другой тянется на страницы. Книга считает одной из больших сил глубокого обучения то, что оно справляется с примерами разной длины гораздо изящнее старых методов. Поэтому чат-бот, которому вы пишете, так же охотно принимает вопрос в три слова, как и в три страницы.

Больше данных обычно всё упрощает. На большем числе примеров можно обучить более мощную модель и заранее закладывать меньше догадок. Книга говорит, что переход от маленьких собраний примеров к огромным — одна из главных причин взлёта глубокого обучения; многие из лучших его моделей просто не работают без огромного количества данных.



Больше данных, правильные данные

Но гора данных бесполезна, если это не те данные. Если примеры полны ошибок или описывают то, что не связано с предсказываемым, обучение проваливается: мусор на входе — мусор на выходе. А там, где машины помогают решать судьбу людей, — в полиции, при найме или выдаче кредитов, — плохие данные наносят настоящий вред. Одна опасность — каких-то

людей среди примеров нет вовсе, как в детекторе рака кожи, которому ни разу не показали тёмную кожу. Другая — примеры несут старые предрассудки: программа, обученная на прошлых решениях фирмы о найме, может научиться автоматически повторять их несправедливость. Авторы подчёркивают, что это может случиться, хотя никто этого не хотел и даже не заметил.

Запомним

У каждой задачи обучения четыре составляющие: данные, модель, оценка того, насколько хорошо она работает, и метод подкрутки её ручек. Данные должны стать числами, и одна маленькая фотография превращается уже в сто с лишним тысяч чисел. Примеры разного размера, вроде отзывов и картинок из интернета, неудобны, а глубокое обучение справляется с ними сравнительно хорошо. Больше данных — лучше, но только правильных: ошибки, пропущенные группы или старые предрассудки в примерах превращаются в провалы и несправедливость машины.

Вопросы

1. Почему фотографию нужно превратить в числа, прежде чем машина сможет на ней учиться?
 - а. Потому что обучающаяся машина работает только с числами, например с показаниями яркости каждого крошечного квадрата

- b. Потому что числа занимают меньше места, чем картинки
 - c. Потому что метки можно приписать только к числам
 - d. Потому что иначе фото было бы слишком большим, чтобы его обрезать
2. Почему отзывы в магазинах — неудобные данные?
- a. Их пишет слишком мало людей
 - b. Они бывают очень разной длины — от двух слов до нескольких страниц
 - c. В них нет полезной информации
 - d. Их нельзя хранить на компьютере
3. Программа, обученная на прошлых решениях фирмы о найме, несправедливо обходится с частью соискателей. Какова вероятная причина?
- a. Ей дали слишком много данных
 - b. Её ручки ни разу не крутили
 - c. Примеры несли старые предрассудки, и она научилась их повторять
 - d. Заявления были слишком короткими
4. Почему переход к огромным собраниям примеров так много значил?
- a. Потому что большим данным не нужна оценка
 - b. Потому что старые методы не умели читать текст
 - c. Потому что большие данные устраняют все предрассудки
 - d. Потому что многие из лучших моделей глубокого обучения просто не работают без огромного количества данных

Ответы

1. **a)** Для обучающейся программы фото — не более чем длинный список показаний яркости; каждый пример должен стать числами.
2. **b)** Один отзыв говорит только «полный отстой!», другой тянется на страницы, так что они приходят не в виде одинакового числа показаний.
3. **c)** Если примеры несут старые предрассудки, обученная на них программа может научиться автоматически повторять эту несправедливость.
4. **d)** Переход от маленьких собраний к огромным — одна из главных причин взлёта глубокого обучения; многим из лучших его моделей нужны огромные объёмы данных.

Задачи

1. Детектор рака кожи отлично работает на проверках у своих создателей, но часто ошибается в клинике, где много пациентов с тёмной кожей. Объясните, что, вероятно, пошло не так и как этого можно было избежать.

Подсказка: Подумайте, кто был среди примеров, на которых учился детектор.

Ответ: Людей с тёмной кожей среди тренировочных примеров, вероятно, не было или почти не было, так что детектор так и не узнал, как болезнь выглядит на тёмной коже. Решение — собрать примеры, охватывающие всех, на ком детектор будут применять.

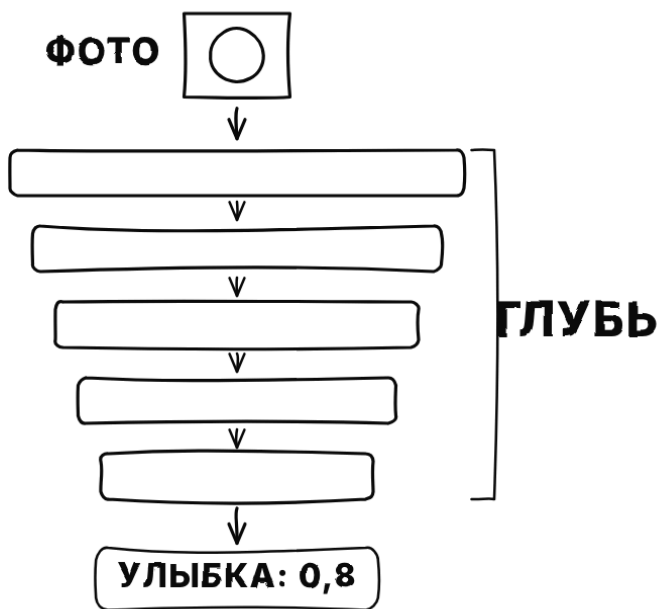
2. Назовите четыре составляющие любой задачи обучения и своими словами скажите, для чего каждая.

Подсказка: На чём учиться, что даёт ответы, что оценивает, что подстраивает.

Ответ: Данные — то, на чём учится машина; модель — программа с ручками, которая даёт ответы; оценка говорит, насколько хорошо или плохо она работает; а метод подкрутки ручек меняет их так, чтобы оценка улучшалась.

Пробные варианты и настоящий экзамен

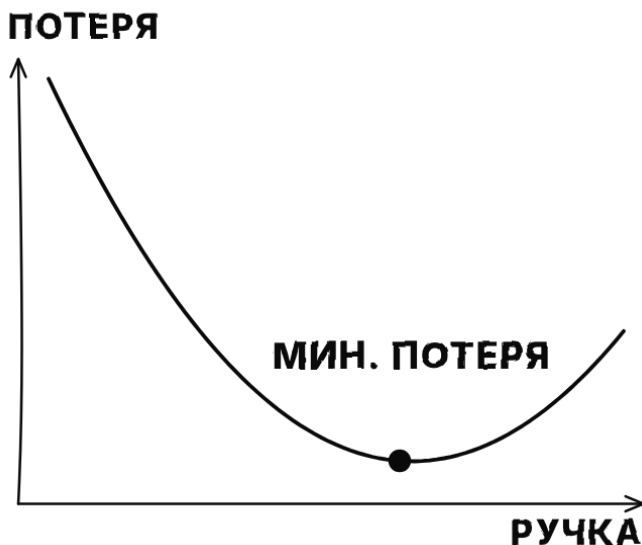
Студентка готовится по прошлогодним пробным вариантам, пока не начинает правильно отвечать на каждый вопрос. Потом приходит настоящий экзамен с вопросами, которых она никогда не видела, и ей приходится трудно. Она зазубрила те варианты, вместо того чтобы выучить предмет. Обучающиеся машины могут попасть точно в такую же ловушку, и эта часть — о том, как это заметить.



Глубокий — значит много шагов

Начнём с модели. Она принимает данные одного рода и выдаёт предсказание, часто другого рода: на вход фото, на выход — насколько человек улыбается; или на вход набор показаний датчиков, на выход — насколько необычными они выглядят. Простые модели годятся для простых задач, но задачам этой книги нужно что-то посильнее. Глубокое обучение отличают мощные модели, построенные из множества шагов, каждый из которых немного преобразует данные и передаёт их следующему, — шагов, сложенных один на другой. Из этой высокой стопки шагов и взялось слово «глубокий».

Дальше — как понять, что модель стала лучше? Люди могут спорить о том, улучшение ли какое-то изменение, поэтому обучающимся машинам нужна твёрдая оценка. Эту оценку называют целевой функцией, и по традиции её устраивают так, что меньше — лучше; поэтому её также называют функцией потерь. При предсказании числа обычная функция потерь быстро растёт, чем дальше догадка от истины; при раскладывании вещей по видам в итоге важно, как часто модель ошибается. Эту вторую оценку неудобно улучшать напрямую, поэтому на практике во время тренировки часто используют более гладкую заменяющую оценку. Спам-фильтр в итоге оценивают по тому, сколько писем он рассортировал неверно, даже если по пути он учится на более гладкой оценке.

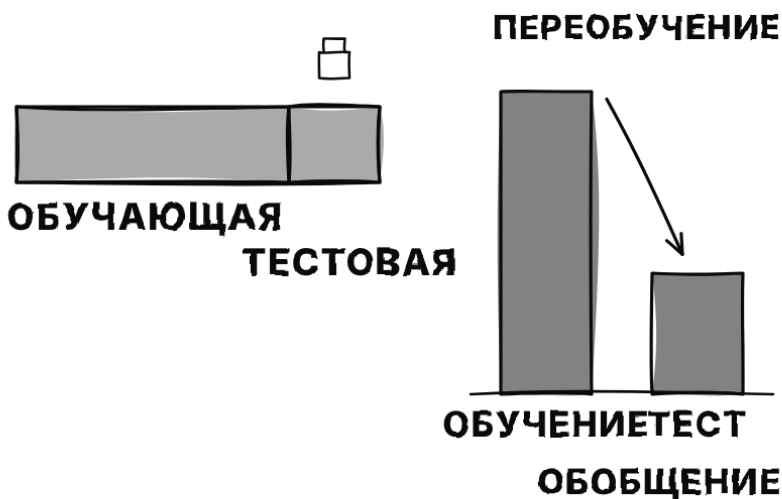


Спуск по склону потерь

Затем нужен способ крутить ручки так, чтобы потеря уменьшалась, и почти всё глубокое обучение использует один метод — градиентный спуск. На каждом шаге он проверяет для каждой ручки, увеличит или уменьшит потерю на тренировочных примерах её крошечный поворот. Потом он чуть поворачивает каждую ручку в ту сторону, где потеря уменьшается, и повторяет. Представьте, что спускаетесь по склону холма в густом тумане: долины не видно, но вы чувствуете, куда наклонена земля под ногами, делаете маленький шаг вниз и снова прощупываете.

Тут возвращается наша студентка. Ручки настраивают по одной стопке примеров, обучающей выборке, а проверяют на другой

стопке, отложенной в сторону и никогда не используемой для настройки, — тестовой выборке. Хорошая оценка на обучающей выборке — как хорошая оценка за пробные варианты: обнадеживает, но ничего не обещает. Умение хорошо справляться с новыми, невиданными примерами называется обобщением, а о модели, которая блистает на тренировочных примерах, но проваливается на новых, говорят, что она переобучена. Лента рекомендаций, которая идеально совпадает с тем, что вы кликали в прошлом месяце, но не предлагает ничего, что вам нужно дальше, — это как раз тот провал, который должна поймать тестовая выборка.



Обучающая и тестовая выборки

Запомним

Модель превращает данные в предсказания, а модели глубокого обучения — высокие стопки шагов, поэтому их и называют глубокими. Функция потерь — твёрдая оценка того, насколько плохо работает модель, и меньше — лучше. Градиентный спуск уменьшает потерю, прощупывая для каждой ручки, где «вниз», и делая туда небольшой шаг, снова и снова. Успех проверяют на тестовой выборке, на которой модель никогда не тренировалась, потому что модель, которая лишь зазубрила тренировочные примеры, переобучена.

Вопросы

1. Откуда взялось слово «глубокое» в глубоком обучении?
 - a. Из огромного количества данных, которое ему нужно
 - b. Из того, что его модели — высокие стопки шагов, каждый из которых преобразует данные и передаёт их дальше
 - c. Из того, насколько трудно понять его идеи
 - d. Из того, насколько низко может опуститься его потеря
2. Что делает градиентный спуск на каждом шаге?
 - a. Даёт всем ручкам новые случайные значения
 - b. Проверяет тестовую выборку и останавливается, если результат хорош

- c. Нашупывает для каждой ручки, в какую сторону потеря уменьшается, и чуть поворачивает каждую ручку туда
 - d. Добавляет примеры в обучающую выборку
3. Почему модель проверяют на тестовой выборке, на которой она никогда не тренировалась?
- a. Потому что хорошая оценка на знакомых примерах не показывает, что модель хорошо справится с новыми
 - b. Потому что обучающая выборка всегда слишком мала
 - c. Потому что на тестовой выборке потеря ниже
 - d. Потому что градиентный спуск работает только на тестовой выборке

Ответы

1. **b)** Модели глубокого обучения построены из множества шагов, сложенных один на другой, и из этой высокой стопки и взялось слово.
2. **c)** Он проверяет для каждой ручки, увеличит или уменьшит потерю её крошечный поворот, а потом чуть поворачивает каждую в сторону уменьшения.
3. **a)** Хорошая оценка на обучающей выборке — как хорошая оценка за пробные варианты: обнадёживает, но ничего не обещает насчёт новых вопросов.

Задачи

1. Объясните, чем спуск по склону холма в тумане похож на градиентный спуск. Что соответствует высоте земли, а что — вашему положению?

Подсказка: Что градиентный спуск старается сделать маленьким?

Ответ: Высота земли соответствует потере, а ваше положение — положениям ручек. Дна вы не видите, но чувствуете, где «вниз» там, где стоите, и делаете туда маленький шаг — так же, как градиентный спуск проверяет, в какую сторону каждая ручка уменьшает потерю, и чуть поворачивает её, снова и снова.

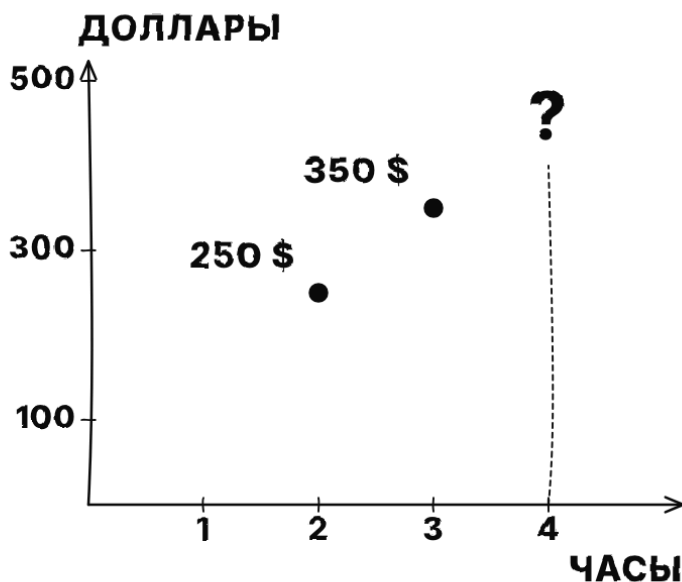
2. Модель права на 99 из 100 тренировочных примеров, но лишь на 60 из 100 тестовых. Что произошло и что об этом говорит история о студентке?

Подсказка: Сравните пробные варианты и настоящий экзамен.

Ответ: Модель переобучена: по сути она зазубрила тренировочные примеры и не обобщает на новые, как студентка, которая зазубрила пробные варианты, а потом растерялась перед новыми вопросами на настоящем экзамене.

Угадать, сколько

Сантехник три часа прочищает ваши засорившиеся трубы и выставляет счёт на триста пятьдесят долларов. Ваш друг вызывает того же сантехника на два часа и платит двести пятьдесят. Теперь сосед спрашивает, сколько будет стоить работа на четыре часа. Вы, наверное, ответите без всякого урока математики, и то, как вы это делаете, — суть этой части.



Два счёта сантехника

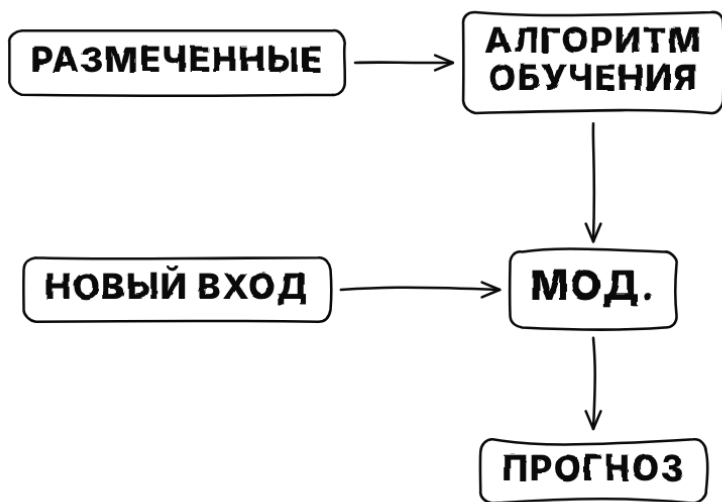
Сначала взглянем шире. Детектор слова пробуждения — лишь одна из многих задач, за которые может взяться машинное

обучение, и книга теперь проходит по основным видам, начиная с самого распространённого. В обучении с учителем нам даны примеры, в которых известно и то, что поступает на вход, — признаки, и правильный ответ — метка. Задача — построить модель, которая по признакам нового случая предсказывает его метку. Оно называется обучением с учителем, потому что мы выступаем учителями и даём модели размеченные примеры, по которым она находит положения своих ручек. Точнее говоря, модель оценивает, насколько вероятен каждый ответ при том, что она видит.

Большая часть машинного обучения, хорошо работающего в бизнесе, — именно такого рода, потому что очень многие задачи укладываются в схему: угадать неизвестное по известному. Есть на этом снимке рак или нет? Как по-французски это английское предложение? Сколько будут стоить акции этой компании в следующем месяце, судя по отчётности этого месяца? Ваш спам-фильтр укладывается в ту же форму: ему показывают письмо, и он угадывает «спам» или «не спам». Схема проста, но принимает много форм; тексты любой длины, например, требуют других моделей, чем входы, которые всегда одного размера.

Книга описывает процесс и изображает его картинкой со стрелками. Соберите большую грудку примеров, случайно выберите несколько и получите правильный ответ для каждого. Иногда ответы уже записаны, например пережил ли пациент следующий год; иногда приходится нанимать людей, чтобы размечать вручную, например раскладывать фотографии по категориям. Эти размеченные примеры и есть обучающая выборка. Они поступают в метод обучения, и выходит обученная

модель; потом новые, никогда не виданные случаи поступают в модель, и её выходы принимают за её предсказания.



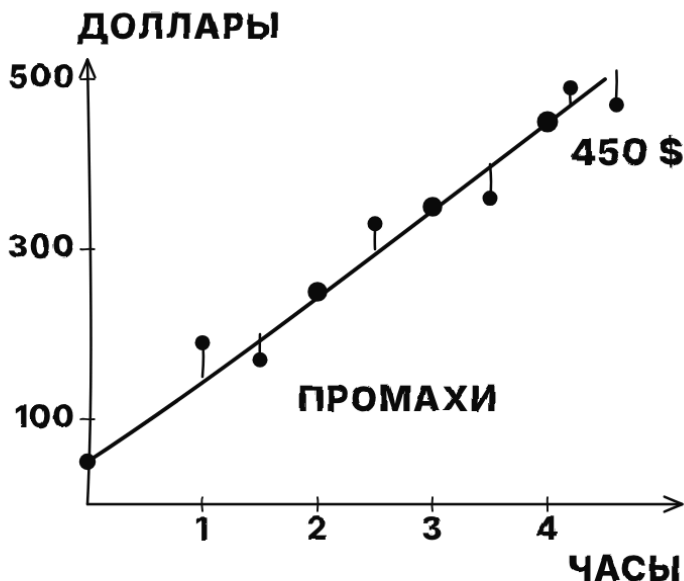
От размеченных примеров к предсказаниям

Самый простой для понимания вид обучения с учителем — регрессия. Представьте таблицу продаж домов: по строке на каждый дом и по столбцу на каждый факт о нём — площадь, спальни, ванные и сколько минут пешком до центра города. Каждая строка — один пример, фиксированный список фактов об одном доме. Маленькая квартира в Нью-Йорке может быть крошечной площадью, одной спальней, одной ванной и часом ходьбы до центра; семейный дом в Питтсбурге — гораздо большей площадью, четырьмя спальнями, тремя ванными и десятью минутами ходьбы. Большинству классических методов

обучения нужно, чтобы каждый пример приходил в виде такого списка одной и той же длины.

Регрессией задачу делает то, какой ответ нужен: число, например цена, за которую продан дом. Если вы покупаете жильё, модель, обученная на прошлых продажах, могла бы оценить справедливую цену по фактам о доме. Удобное правило: любой вопрос «сколько?» — скорее всего, регрессия. Сколько продлится эта операция, сколько дождя выпадет здесь за следующие шесть часов, сколько дней этот пациент пробудет в больнице? Когда стриминговый сервис угадывает, сколько звёзд вы поставили бы фильму, — это тоже регрессия, и в 2009 году команда, которая делала это достаточно хорошо, выиграла приз в миллион долларов от Netflix.

А теперь вернёмся к сантехнику. Вы, вероятно, предположили, что больше часов стоит дороже и что есть фиксированная плата за вызов плюс ставка за каждый час. При таком предположении два счёта сразу дают ответ: лишний час стоил сто долларов, значит, ставка — сто в час, а пятьдесят долларов — просто за то, что он пришёл. Если вы это проследили, вы уже понимаете идею линейной регрессии, самого простого её вида.



Плата за вызов плюс ставка

Здесь два счёта точно укладываются в правило. В реальных данных так бывает редко, потому что ответы сдвигает и то, что вы не измерили: засор потруднее, дорога подольше, неудачный день. Тогда модель не может совпасть с каждым примером и старается держать свои догадки в целом как можно ближе к настоящим ответам. Обычный способ измерять «близость» наказывает большие промахи гораздо сильнее маленьких, и книга в основном будет пользоваться им. Авторы добавляют, что такой выбор естествен, когда случайные отклонения ведут себя как обычный случайный шум.

Запомним

Обучение с учителем — это обучение предсказывать метку по признакам на примерах, где дан правильный ответ.

Большая часть успешного машинного обучения в реальном мире работает так, от медицинских снимков до перевода.

Регрессия — обучение с учителем, где ответ — число: любой вопрос «сколько?». Два счёта сантехника раскрывают плату за вызов плюс почасовую ставку — это и есть идея линейной регрессии. Когда реальные данные не укладываются точно, модель держит свои догадки как можно ближе, сильнее всего наказывая большие промахи.

Вопросы

1. Какой из этих вопросов — задача регрессии?
 - a. Это письмо — спам или нет?
 - b. Есть ли рак на этом снимке?
 - c. Сколько дней этот пациент пробудет в больнице?
 - d. Как по-французски это предложение?
2. Почему такое обучение называют обучением с учителем?
 - a. Потому что мы даём модели размеченные примеры, по которым она находит положения ручек
 - b. Потому что кто-то проверяет каждое предсказание вручную
 - c. Потому что оно применяется в основном в бизнесе

- d. Потому что модель может работать, только пока за ней следят
3. Почему модель, обученная на настоящих ценах домов, обычно не может точно совпасть с каждой продажей?
- a. Потому что у домов слишком мало признаков
 - b. Потому что цены сдвигает и то, чего нет в таблице, например спешка покупателя
 - c. Потому что регрессия работает только для сантехников
 - d. Потому что обучающую выборку никогда не размечают
4. Как обычная мера близости относится к большому промаху по сравнению с маленьким?
- a. Не учитывает большие промахи
 - b. Считает все промахи одинаково
 - c. Учитывает только самый большой промах
 - d. Наказывает большой промах гораздо сильнее

Ответы

1. c) Регрессия — обучение с учителем, где ответ — число; любой вопрос «сколько?» — скорее всего, регрессия.
2. a) Мы выступаем учителями и даём модели размеченные примеры, по которым она находит положения своих ручек.
3. b) В реальных данных ответы сдвигает и то, что вы не измерили, поэтому модель держит свои догадки в целом как можно ближе.
4. d) Обычный способ измерять близость наказывает большие промахи гораздо сильнее маленьких.

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 каждый час, каковы плата и ставка и сколько стоила бы работа на 4 часа?

Две работы различаются на один час и на 100 долларов, значит, каждый час стоит 100 долларов.

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

Проверим по другому счёту: 50 плюс 2 часа по 100 — это 250, как и было выставлено.

Работа на 4 часа стоила бы 50 плюс 4 часа по 100.

Ответ: 100 долларов в час плюс 50 долларов за приход; работа на 4 часа стоит 450 долларов

Задачи

1. Разделите на регрессию и не регрессию: сколько дождя выпадет за следующие шесть часов; кошка на этом фото или собака; сколько часов займёт эта операция; за какую цену продастся этот дом.

Подсказка: Ответ — число, ответ на вопрос «сколько?»?

Ответ: Дождь, длительность операции и цена дома требуют чисел, значит, это регрессия. Кошка или собака требует категории, значит, это не регрессия.

2. Второй сантехник взял 200 долларов за 1 час и 400 долларов за 3 часа. Если расценки устроены так же, каковы его фиксированная плата и почасовая ставка?

Подсказка: Два лишних часа добавили 200 долларов.

Ответ: Два часа добавили 200 долларов, значит, ставка — 100 долларов в час; если вычесть из первого счёта 100 за один час, остаётся 100 долларов, значит, фиксированная плата — 100 долларов.

Что это?

Представьте красивый бледный гриб, выросший у вас в саду. Вы фотографируете его приложением, и оно говорит, что на восемьдесят процентов уверено: гриб безвреден. Пожарили бы вы его на ужин? Запомните свой ответ; книга вернётся к этому грибу.

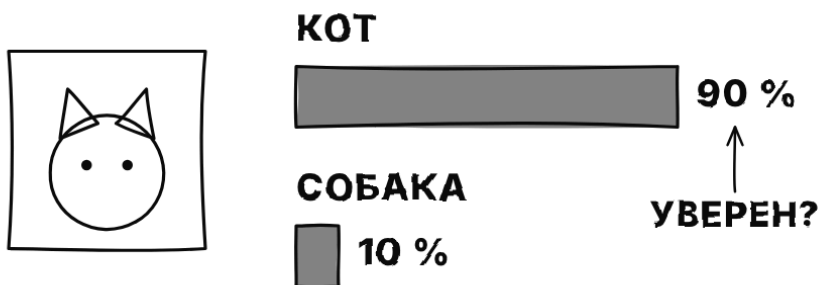


Какой это класс?

Регрессия отвечает на вопрос «сколько?», но многие вопросы звучат как «что из этого?». Возьмём банковское приложение, которое читает чек по фотографии. Когда картинка разрезана на

маленькие кусочки, по одному на каждый рукописный знак, остаётся решить, какой знак из известного набора изображён на каждом кусочке. Такие задачи называются классификацией, и для них нужны несколько иные инструменты, чем для регрессии, хотя многое переносится.

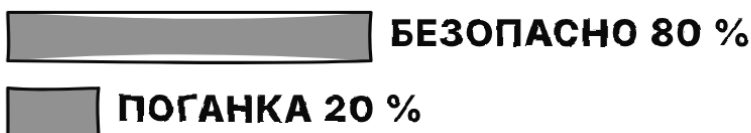
При классификации модель смотрит на признаки примера, например на пиксели изображения, и говорит, к какой категории, или классу, он относится, из фиксированного набора вариантов. Для рукописных цифр классов десять, от нуля до девяти. В самом простом случае классов только два, например кошка или собака, и это называется бинарной классификацией. Ваш спам-фильтр — бинарная классификация: каждое письмо либо спам, либо нет. Там, где регрессия выдаёт число, классификация выдаёт выбор класса. Когда классов больше двух, как у всех цифр и букв рукописного текста, это называется многоклассовой классификацией.



Вероятность для каждого класса

Модель, которая может сказать только категоричное «кошка» или «собака», оказывается трудно обучать, поэтому вместо этого она обычно выдаёт вероятность для каждого класса. Увидев картинку, она может сказать «девяносто процентов — кошка», и это можно понимать как «уверена на девяносто процентов». Величина этой вероятности даёт представление о том, насколько модель не уверена. Как регрессию обычно тренируют, держа квадраты промахов маленькими, так и у классификации есть своя обычная оценка того, насколько модель ошибается, со странным названием «перекрёстная энтропия»; книга объяснит её гораздо позже.

А теперь вернёмся к грибу. Допустим, приложение говорит, что с вероятностью двадцать процентов это бледная поганка, а значит, с вероятностью восемьдесят — нет. Самый вероятный ответ — «безопасно», но съест его только безумец. Удовольствие от одного хорошего ужина не стоит шанса один к пяти умереть. Чтобы решить, что делать, мало знать, какой ответ самый вероятный; нужно взвесить каждый выбор с учётом того, насколько вероятен каждый исход и насколько он хорош или ужасен. Съесть — значит рискнуть катастрофой; выбросить гриб — стоит самое большее одного ужина. И осторожность была верной: гриб на картинке в книге действительно бледная поганка.

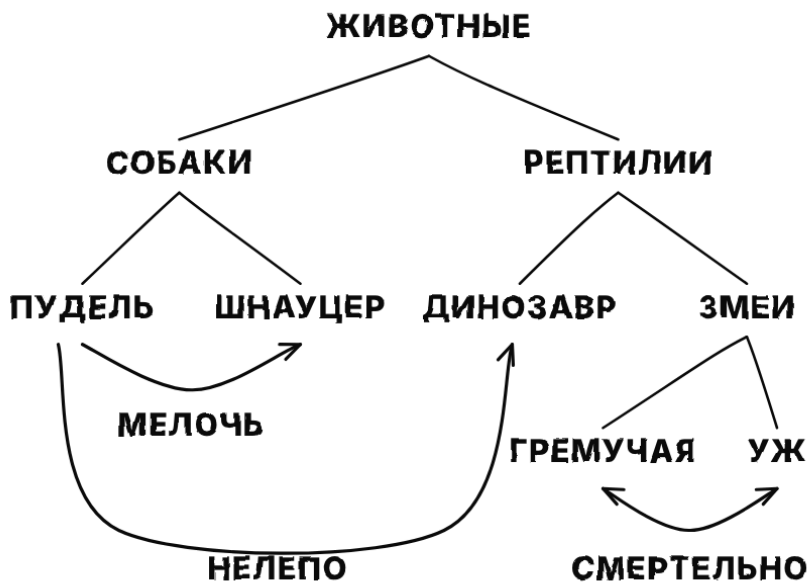


Вероятно — не значит безопасно

Опыт

В следующий раз, когда приложение или сайт покажет вам процент уверенности, спросите себя, чего будет стоить, если правдой окажется менее вероятный ответ. Вы обнаружите, что девяноста процентов «да» вполне хватает, чтобы выбрать песню, и совсем не хватает, чтобы перейти дорогу или принять лекарство.

Классификация может быть ещё тоньше, потому что не все ошибки равны. Иногда классы образуют родословное древо — так натуралист Линней раскладывал животных по вложенным группам, — и тогда спутать близких родственников — меньшая ошибка, чем спутать дальних. Принять пуделя за шнауцера простительно; принять пуделя за динозавра — нелепо. Это называется иерархической классификацией. Но какое древо важно, зависит от того, для чего модель. Гремучая змея и безобидный подвязочный уж — близкие родственники, но спутать их может быть смертельно.



Одни ошибки хуже других

Запомним

Классификация отвечает на вопрос «что из этого?»: она относит каждый пример к одному классу из фиксированного набора. Два класса — бинарная классификация, больше двух — многоклассовая. Модели обычно дают вероятность для каждого класса, и это заодно показывает, насколько они уверены. Действовать нужно не всегда по самому вероятному классу: двадцати процентов шанса, что это бледная поганка, достаточно, чтобы гриб не есть. Когда классы образуют родословное древо, одни ошибки хуже других, и какие важнее всего, зависит от применения, как с гремучими змеями и ужами.

Вопросы

1. Какая из этих задач — классификация, а не регрессия?
 - a. За сколько продается этот дом?
 - b. Какая цифра написана на этом кусочке чека?
 - c. Сколько часов займёт этот ремонт?
 - d. Сколько дождя выпадет завтра?
2. Приложение на восемьдесят процентов уверено, что гриб — не бледная поганка. Почему его всё равно не стоит есть?
 - a. Потому что восемьдесят процентов — наименее вероятный ответ

- b. Потому что приложение обучали на слишком малом числе грибов
 - c. Потому что двадцать процентов шанса смертельного исхода перевешивают один хороший ужин
 - d. Потому что грибы нельзя классифицировать по фото
3. Почему принять гремучую змею за подвязочного ужа хуже, чем подсказывает их близость в родословном древе?
- a. Потому что насколько плоха ошибка, решают её последствия, а не только близость классов
 - b. Потому что змей нельзя классифицировать иерархически
 - c. Потому что модель дала обеим одинаковую вероятность
 - d. Потому что гремучие змеи реже встречаются в данных
4. Почему классификаторы обычно дают вероятность для каждого класса, а не один твёрдый ответ?
- a. Потому что твёрдые ответы дольше выводятся на экран
 - b. Потому что вероятности всегда верны
 - c. Потому что пользователи предпочитают числа
 - d. Потому что модель, дающую только категоричный ответ, трудно обучать, а вероятности ещё и показывают, насколько она уверена

Ответы

1. **b)** Классификация отвечает на вопрос «что из этого?»: решить, какой знак из известного набора изображён на кусочке.
2. **c)** Чтобы решить, нужно взвесить каждый выбор с учётом того, насколько вероятен каждый исход и насколько он хорош или ужасен;

шанс один к пяти умереть не стоит ужина.

3. а) Какое дерево важно, зависит от того, для чего модель; две змеи — близкие родственники, но спутать их может быть смертельно.

4. d) Модель, которая может сказать только категоричное «кошка» или «собака», трудно обучать, поэтому она даёт вероятность для каждого класса, и это заодно показывает, насколько она не уверена.

Задачи

1. Разделите на бинарную классификацию, многоклассовую классификацию и регрессию: спам или не спам; какая из десяти цифр на кусочке чека; сколько пациент пробудет в больнице; какая буква или цифра — рукописный знак.

Подсказка: Ответ — число, один из двух классов или один из многих?

Ответ: Спам или не спам — бинарная классификация. Десять цифр, а также буквы вместе с цифрами — многоклассовая классификация. Срок пребывания в больнице — число, значит, это регрессия.

2. Погодное приложение говорит, что с вероятностью девяносто процентов дождя не будет. Объясните, почему вы спокойно оставите зонт дома, идя на прогулку, но не оставите ценную картину на ночь в саду при том же прогнозе.

Подсказка: Сравните в каждом случае, насколько вероятен плохой исход и насколько он плох.

Ответ: Шанс дождя тот же, один к десяти, но вред разный. Немного промокнуть на прогулке — небольшая цена, так что риск приемлем; испорченная картина — большая потеря, так что даже шанс один к десяти не стоит того. Как и с грибом, решение зависит и от вероятности, и от последствий.

Теги, рейтинги и ваша лента

Вспомните старую немецкую сказку о бременских музыкантах: осёл, на спине у него собака, на собаке кошка, а на самом верху петух. Покажите эту картинку программе, обученной отвечать «кошка или собака?», и она застрянет. Ей нужно выбрать одно, а там есть и то и другое, да ещё осёл и петух. Как бы хороша ни была программа, вопрос, на который её научили отвечать, — не тот вопрос для этой картинки.

МНОГО МЕТОК

ПЕТУХ

КОТ

ПЁС

ОСЁЛ

КОТ

ПЁС

ОСЁЛ

ПЕТУХ

Одна картинка, несколько меток

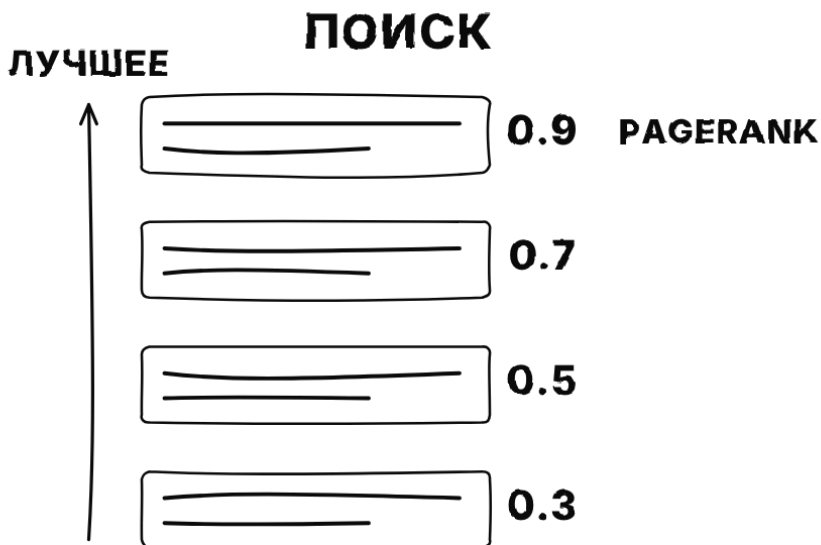
Когда пример может относиться сразу к нескольким классам, задача называется классификацией с несколькими метками. Вместо того чтобы выбрать один ответ, модель может сказать: кошка, и собака, и осёл, и петух. Так работает автоматическая расстановка тегов. Запись в технологическом блоге может нести от пяти до десяти тегов, например «машинное обучение», «языки программирования», «облачные вычисления». Теги обычно ходят компаниями: записи об облачных вычислениях часто упоминают облачный сервис Amazon, а записи о машинном обучении часто упоминают графические чипы, на которых его тренируют.

Иногда набор возможных тегов огромен. В Национальной медицинской библиотеке США обученные сотрудники помечают каждую статью, добавляемую в её большой медицинский указатель, терминами из списка примерно в двадцать восемь тысяч предметных рубрик. Теги позволяют исследователям найти всё, что написано по какому-то вопросу, но работа идёт медленно, и статья может ждать их около года. Машинное обучение может тем временем выдавать предварительные теги, пока их не проверит человек, и уже много лет проводятся соревнования о том, как делать это хорошо.

Поиск — родственная задача, но вопрос меняется. При поиске в интернете подходящих страниц обычно много; важно, какие из них показать наверху. Один способ — дать каждой странице оценку и показывать сначала самые высокие оценки.

Первоначальный секретный ингредиент Google, PageRank, был такой оценкой, и, как ни странно, она вовсе не зависела от того, что вы набрали. Простой фильтр отбирал страницы, совпадающие с вашими словами, а PageRank затем ставил вперёд

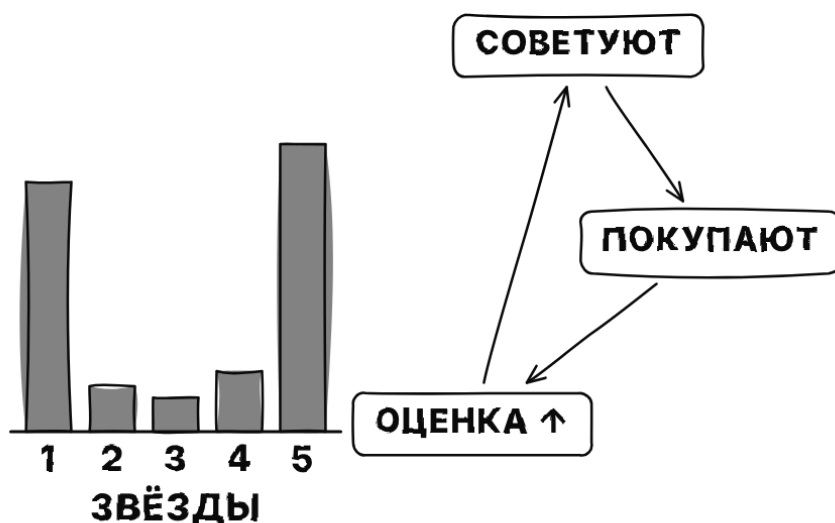
более авторитетные страницы. Сегодня поисковые системы используют машинное обучение и модели поведения людей, чтобы оценить каждую страницу для конкретного запроса.



Поиск ранжирует по оценке

Рекомендации — близкий родственник поиска, с одним большим отличием: они личные. Любитель научной фантастики и поклонник старых комедий должны видеть совсем разные советы о фильмах, и то же касается покупок, музыки и новостей. Иногда люди сами говорят системе, что думают, — звёздами и отзывами. Чаще они сообщают это только своими действиями, например пропуская песню, что может значить, что она им не нравится, а может — просто что она не подошла к моменту. Когда ваша лента рекомендаций перестраивается после того, как вы

пропустили несколько видео, она учится именно на таком тихом сигнале. В самом простом виде система выучивает оценку для каждого товара, например ожидаемое число звёзд или шанс, что вы его купите, и показывает вам товары с самыми высокими оценками. Книга даже показывает страницу книг о глубоком обучении, которые крупный интернет-магазин предложил одному из её авторов.



Перекошенные оценки, петли самоподпитки

Построенные наивно, такие системы имеют глубокие изъяны. Во-первых, обратная связь перекошена: люди в основном оценивают то, что им очень понравилось или очень не понравилось, поэтому оценки скапливаются на одной звезде и на пяти, а посередине их странно мало. 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: система продвигает товар, больше людей его покупает, покупки делают его на вид лучше, и система продвигает его ещё сильнее. Поэтому лента, похоже, показывает вам всё больше одного и того же, стоит вам несколько раз на это кликнуть. Как справляться с перекошенной обратной связью, со стимулами людей и с такими петлями — всё ещё открытые вопросы исследований.

Запомним

Классификация с несколькими метками позволяет примеру относиться сразу к нескольким классам, как четыре животных на одной картинке, и так работает автоматическая расстановка тегов. Наборы тегов могут быть огромными, как двадцать восемь тысяч медицинских рубрик, где машинное обучение предлагает предварительные теги. Поиск упорядочивает подходящие результаты по оценке и показывает лучшие первыми; PageRank был ранней оценкой, не учитывавшей, что вы набрали. Рекомендации — это личное ранжирование, выученное по оценкам и по тому, что люди делают. Перекошенные оценки и петли, в которых рекомендации подпитывают собственный успех, делают рекомендации труднее, чем кажется.

Вопросы

1. Почему программа «кошка или собака» проваливается на картинке с бременскими музыкантами?
 - a. Потому что картинка — рисунок, а не фото
 - b. Потому что она должна выбрать один класс, а на картинке сразу несколько животных
 - c. Потому что ослов распознать труднее, чем кошек
 - d. Потому что она была недостаточно точной
2. Чем PageRank, ранняя оценка Google, отличался от сегодняшних оценок поиска?
 - a. Он использовал звёздные оценки пользователей
 - b. Он оценивал каждую страницу для конкретных набранных слов
 - c. Он не зависел от того, что набрали; подходящие страницы отбирал отдельный фильтр
 - d. Он упорядочивал страницы случайно
3. Почему оценки часто скапливаются на одной звезде и на пяти?
 - a. Потому что люди в основном оценивают то, что им очень понравилось или очень не понравилось
 - b. Потому что рекомендательная система прячет оценки в три звезды
 - c. Потому что большинство товаров либо идеальны, либо сломаны
 - d. Потому что у звёздной шкалы только два значения

4. О какой петле обратной связи в рекомендациях предупреждает книга?
- a. Пользователи дважды оценивают один и тот же товар
 - b. Оценки теряются, когда песню пропускают
 - c. Система рекомендует только то, что никто не покупал
 - d. Система продвигает товар, больше людей его покупает, поэтому он выглядит лучше и его продвигают ещё сильнее

Ответы

1. **b)** Ей нужно выбрать одно, а там есть и кошка, и собака, и не только; вопрос, на который её научили отвечать, — не тот вопрос для этой картинки.
2. **c)** PageRank, как ни странно, не зависел от того, что вы набрали; простой фильтр отбирал подходящие страницы, а PageRank ставил вперёд более авторитетные.
3. **a)** Обратная связь перекошена: люди в основном оценивают то, что им очень понравилось или очень не понравилось, поэтому средних оценок странно мало.
4. **d)** Система продвигает товар, больше людей его покупает, покупки делают его на вид лучше, и система продвигает его ещё сильнее.

Задачи

1. Для каждого случая скажите, может ли быть верен один класс или сразу несколько: фото одного питомца, кошки или собаки; запись в блоге об облачных вычислениях и программировании; медицинская статья о диабете и сердце.

Подсказка: Может ли пример относиться сразу к нескольким классам?

Ответ: Один питомец — обычная классификация с одним классом. Запись в блоге и медицинская статья могут каждая нести сразу несколько тегов, так что это классификация с несколькими метками — так и работает автоматическая расстановка тегов.

2. Вы пропускаете в музыкальном приложении три песни подряд. Объясните, почему приложение не может быть уверено, что они вам не нравятся, и что это за вид обратной связи.

Подсказка: Что ещё может означать пропуск?

Ответ: Пропуск — обратная связь, которую вы даёте своими действиями, а не оценкой. Он может значить, что песни вам не нравятся, но может значить и просто то, что они не подошли к моменту, поэтому приложению нужно обращаться с этим сигналом осторожно.

Когда важен порядок

Поставьте фильм на паузу в случайный момент и посмотрите на один-единственный кадр. Рука мужчины поднята; он машет, бросает или собирается ударить? Прокрутите секунду до и секунду после, и ответ очевиден. Один кадр может ввести в заблуждение; историю рассказывают кадры вокруг него.

КАДРЫ ВОКРУГ



ЛЮБАЯ ДЛИНА



ИСТОРИЯ БОЛЕЗНИ

Один кадр обманывает, кадры рассказывают

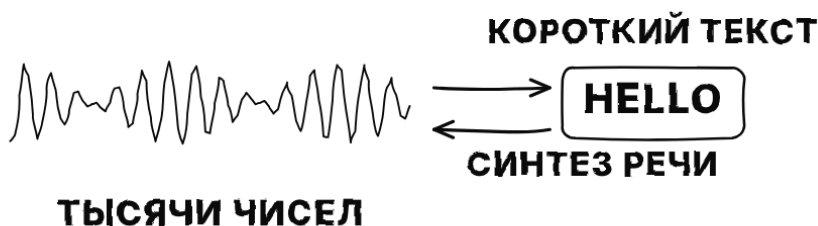
Всё до сих пор имело фиксированную форму. Дом приходил одним и тем же коротким списком фактов, фото — одной и той же сеткой пикселей, а ответом была одна цена или одна

вероятность на класс. И каждый пример забывался в тот же миг, как модель на него ответила, будто каждый случай никак не связан с предыдущим. Для домов это нормально, но многие задачи устроены иначе.

В видеоролике может быть сколько угодно кадров, и то, что происходит в одном кадре, гораздо понятнее, когда модель смотрит и на соседние. С языком то же самое: при машинном переводе на вход поступает целое предложение на одном языке, а на выходе получается целое предложение на другом. В отделении реанимации больницы модель могла бы наблюдать за пациентами и поднимать тревогу всякий раз, когда риск смерти в ближайшие сутки превышает некий порог. Было бы глупо каждый час забывать историю каждого пациента и судить о нём только по последним показаниям. Такие задачи называются обучением на последовательностях, потому что модель принимает последовательности, или выдаёт их, или и то и другое. Когда и вход, и выход — последовательности любой длины, это называется обучением «последовательность в последовательность».

TOM HAS DINNER IN
ENT — — —

WASHINGTON WITH SALLY
ENT — ENT



Метки по порядку; длинный звук, короткий текст

Книга выделяет несколько частных случаев. Первый — разметка слов в предложении, где каждое слово получает одну метку, в том же порядке. Можно помечать каждое слово как существительное, глагол и так далее или отмечать, какие слова называют людей, места или организации: в «Tom has dinner in Washington with Sally» помечаются имена Tom, Washington и Sally.

Второй — распознавание речи: на вход поступает запись, а на выходе получается письменная расшифровка. Трудность в том, что звук измеряется тысячи раз в секунду, так что одно произнесённое слово может растянуться на тысячи измерений, и аккуратного соответствия «один к одному» между звуком и буквами нет. Выход намного короче входа. Когда ваш голосовой

помощник выписывает ваш вопрос на экран, он только что решил эту задачу, сжав длинный поток звука в горстку слов. Люди хорошо распознают речь даже по хрипящей телефонной линии, а для компьютеров это грозная задача. Обратная задача, синтез речи, превращает короткий текст в длинную звукозапись.

Третий случай — машинный перевод, и здесь появляется новый поворот: части не обязаны идти в том же порядке. Немецкий, например, любит ставить глагол в конец. «Haben Sie sich schon dieses grossartige Lehrwerk angeschaut?» означает «Вы уже посмотрели этот замечательный учебник?» Если переводить по словам в немецком порядке, получится «Вы себя уже этот замечательный учебник посмотрели?» — так по-английски никто не скажет. Чат-бот, которому вы пишете, сталкивается со всем этим сразу, и не только: чтобы решить, что сказать дальше, он должен держать в уме весь разговор до сих пор и опираться на знания о мире. Определить, в каком порядке человек читает веб-страницу, или вести долгий разговор — это всё ещё активные области исследований.

НЕМЕЦКИЙ

АНГЛИЙСКИЙ

HABEN SIE

HAVE YOU

SCHON

ALREADY

DIESES LEHRWERK

LOOKED AT

ANGESCHAUT?

THIS TEXTBOOK?

ГЛАГОЛ В КОНЦЕ

Перевод меняет порядок

Запомним

Многие задачи не приходят в фиксированной форме: видео, предложения и истории болезни разворачиваются во времени, и каждая часть имеет смысл только вместе с другими. Обучение на последовательностях работает со входами или выходами-последовательностями, а обучение «последовательность в последовательность» — и с теми и с другими, любой длины. Разметка даёт каждому слову одну метку по порядку; распознавание речи сжимает длинный звук в короткий текст, а синтез речи делает обратное. Перевод ещё труднее, потому что соответствующие части могут идти в другом порядке, как немецкие глаголы в конце предложения.

Вопросы

1. Почему было бы глупо, если бы модель в реанимации судила о каждом пациенте только по последним показаниям?
 - a. Потому что последние показания всегда неверны
 - b. Потому что история пациента во времени помогает увидеть настоящий риск
 - c. Потому что модель может читать только одно число за раз
 - d. Потому что тревогу нужно поднимать каждый час

2. Что делает распознавание речи трудным по сравнению с разметкой слов в предложении?
- a. Одно слово может растянуться на тысячи звуковых измерений, так что звук и буквы не соответствуют друг другу один к одному
 - b. Речь всегда фиксированной длины
 - c. Каждому слову нужно дать метку части речи
 - d. Выход длиннее входа
3. Какую новую трудность добавляет машинный перевод?
- a. Во входе нет слов
 - b. Выход всегда короче входа
 - c. Соответствующие части могут идти в другом порядке, как немецкие глаголы в конце
 - d. Каждое слово получает ровно одну метку

Ответы

1. **b)** Было бы глупо каждый час забывать историю каждого пациента и судить о нём только по последним показаниям.
2. **a)** Звук измеряется тысячи раз в секунду, так что одно произнесённое слово растягивается на тысячи измерений, и аккуратного соответствия «один к одному» нет.
3. **c)** При переводе части не обязаны идти в том же порядке; немецкий, например, любит ставить глагол в конец.

Задачи

1. Для каждой задачи скажите, выход примерно такой же длины, как вход, намного короче или намного длиннее: разметка

каждого слова предложения; превращение записи в текст; чтение текста вслух.

Подсказка: Сравните, сколько частей поступает на вход и сколько выходит.

Ответ: Разметка даёт одну метку на слово, так что выход такой же длины, как вход. Распознавание речи превращает тысячи звуковых измерений в несколько слов, так что выход намного короче. Синтез речи превращает короткий текст в длинную запись, так что выход намного длиннее.

2. Объясните на примере поставленного на паузу фильма, почему модель, которая забывает каждый пример после ответа, плохо подходит для видео.

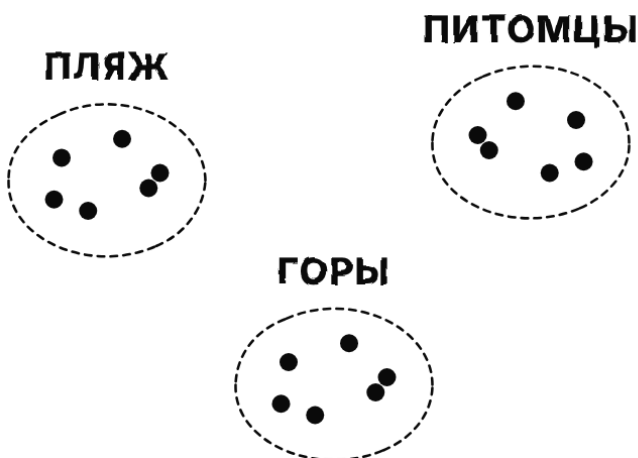
Подсказка: Что помогает отличить взмах рукой от броска?

Ответ: Один кадр сам по себе может быть двусмысленным, как поднятая рука, которая может и махать, и бросать. Кадры до и после всё проясняют, так что модель, которая судит о каждом кадре отдельно и забывает остальные, выбрасывает ровно тот контекст, который ей нужен.

Учиться без ответов

Пролистайте галерею на телефоне и найдите альбомы, которые она составила сама: пляжи, питомцы, горы, одни и те же несколько лиц. Никто не говорил ей, какое фото — пляж. Она нашла группы сама, не имея ничего, кроме картинок. Как машина может чему-то научиться, если никто не даёт ей ответов?

КЛАСТЕРИЗАЦИЯ



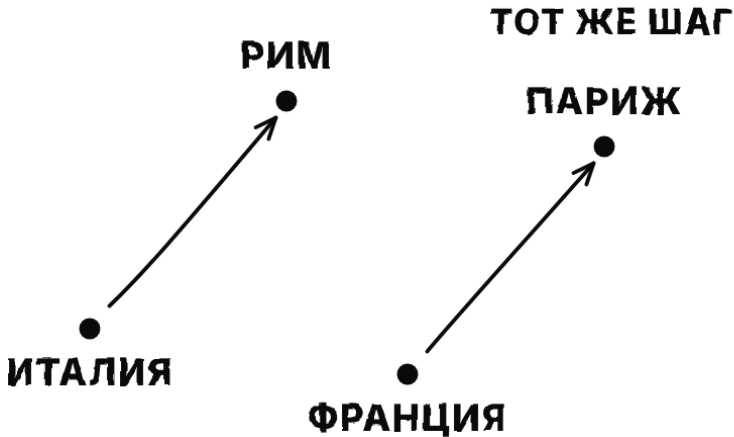
Группы, найденные без ответов

Обучение с учителем, говорит книга, похоже на работу у очень властного начальника, который стоит у вас за плечом и говорит, что именно делать в каждой ситуации. Скучно, наверное, но

угодить легко: заметить закономерность и повторять за начальником. Противоположная работа — начальник, который вручает вам гору данных и говорит: «Займитесь-ка с ними наукой о данных!» Звучит расплывчато, потому что так и есть. Обучение на данных без приложенных ответов называется обучением без учителя, и вопросы, которые можно задать, ограничены только вашим воображением.

Книга перечисляет несколько. Можно ли разложить данные на горстку естественных групп, например фото пейзажей, собак, младенцев, кошек и горных вершин, или покупателей, которые просматривают товары похожим образом? Это называется кластеризацией. Могут ли несколько чисел уловить главное? Путь брошенного мяча хорошо описывается его скоростью, размером и весом, а портные подгоняют одежду по небольшому набору мерок. Можно ли расположить слова как точки на карте, чтобы отношения между ними стали направлениями, так что путь от Италии к Риму — тот же шаг, что от Франции к Парижу? Можно ли найти скрытые причины того, что мы видим, например как цены на жильё, загрязнение, преступность, образование и зарплаты влияют друг на друга?

КАРТА СЛОВ

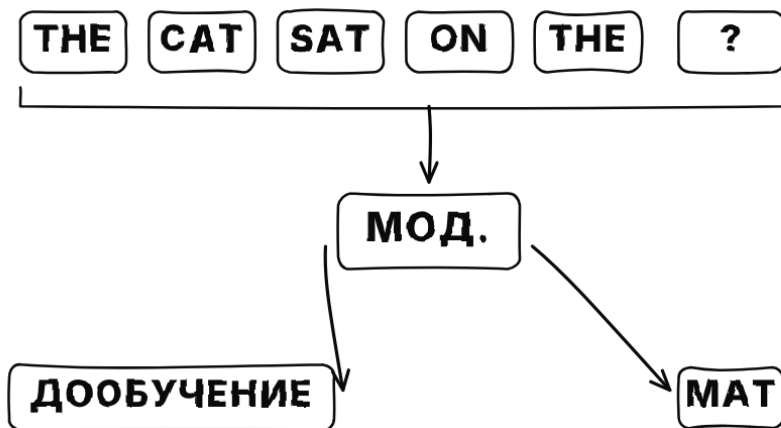


Слова как точки на карте

Одно из самых захватывающих недавних направлений — генеративные модели. Они изучают, как выглядят данные, так хорошо, что могут судить, насколько вероятен пример, или создавать совершенно новые собственные примеры. Генераторы изображений, которые рисуют картинку по вашему описанию, принадлежат к этому семейству; книга называет диффузионные модели среди новейших достижений.

Другое направление — самообучение, когда сами данные дают и вопросы, и ответы. Для текста: спрятать случайное слово и попросить модель угадать его по окружающим словам; огромные объёмы текста можно так использовать вовсе без разметки. Чат-бот, которому вы пишете, начинал с тренировки такого рода,

учась предсказывать пропущенные или следующие слова по огромному количеству текста. Для картинок модель можно попросить определить, где один вырезанный кусочек находился относительно другого, дорисовать закрытую часть или сказать, получены ли два изменённых изображения из одного оригинала. То, чему модель так научилась, потом можно использовать снова: её дообучают, немного дотренировав, для той конкретной задачи, которая вам на самом деле нужна.



Спрячь слово — угадай обратно

Запомним

У обучения с учителем — властный начальник со всеми ответами; обучение без учителя получает только данные и должно найти в них структуру. Оно может найти естественные группы, несколько чисел, улавливающих главное, отношения между словами и скрытые причины. Генеративные модели изучают данные так хорошо, что могут создавать собственные новые примеры. Самообучение составляет собственные вопросы из данных, например угадывание спрятанных слов, и то, чему оно научилось, потом дообучают для нужной задачи.

Вопросы

1. В чём главное отличие обучения с учителем от обучения без учителя?
 - a. Обучение без учителя использует меньше данных
 - b. При обучении без учителя данные приходят без ответов, и модель должна сама найти структуру
 - c. Обучению без учителя нужен наблюдающий человек
 - d. Обучение с учителем не справляется с фотографиями
2. Ваш телефон раскладывает фото на пляжи, питомцев и горы, хотя ему не говорили, что есть что. Как это называется?
 - a. Кластеризация

- b. Дообучение
 - c. Регрессия
 - d. Классификация с метками
3. Как самообучение получает свои вопросы и ответы, если никто не размечает данные?
- a. Копирует ответы модели, обученной с учителем
 - b. Просит пользователей оценивать примеры
 - c. Прячет часть данных, например слово, и учится угадывать её по остальному
 - d. Работает только на данных, размеченных давным-давно
4. Что происходит после того, как модель прошла самообучение?
- a. Её выбрасывают
 - b. Её используют только для угадывания спрятанных слов
 - c. Она становится моделью кластеризации
 - d. Её дообучают, немного дотренировав, для нужной нам задачи

Ответы

1. **b)** Обучение на данных без приложенных ответов называется обучением без учителя.
2. **a)** Раскладывание данных на горстку естественных групп, например пейзажи, собак и кошек, называется кластеризацией.
3. **c)** Для текста: спрятать случайное слово и попросить модель угадать его по окружающим словам, вовсе без разметки.

4. d) То, чему модель так научилась, можно использовать снова: её дообучают, немного дотренировав, для той конкретной задачи, которая вам нужна.

Задачи

1. Скажите, что из этого кластеризация, генеративная модель или самообучение: группировка покупателей по тому, как они просматривают товары; приложение, которое рисует картинку по вашему описанию; угадывание скрытого слова по окружающим словам.

Подсказка: Задача — найти группы, создать новые примеры или ответить на вопрос, составленный из самих данных?

Ответ: Группировка покупателей — кластеризация; рисование картинки по описанию — работа генеративной модели; угадывание скрытого слова — самообучение.

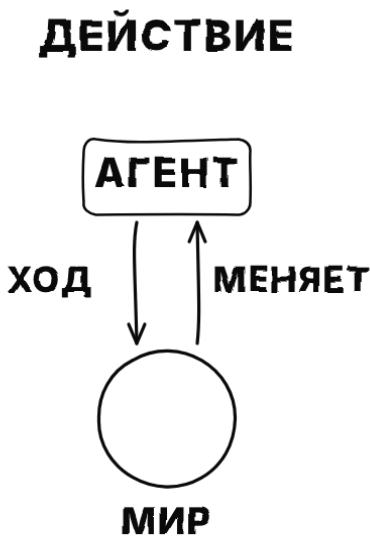
2. Объясните своими словами, почему мерки портного — хороший пример поиска «нескольких чисел, улавливающих главное».

Подсказка: Сколько чисел нужно, чтобы описать тело достаточно хорошо для подгонки одежды?

Ответ: Человеческое тело очень сложное, но небольшой набор мерок, например обхват груди, талии и длина, описывает его форму достаточно хорошо, чтобы подогнать одежду. Так же обучение без учителя может искать несколько чисел, которые улавливают важные свойства сложных данных.

Учиться на деле

Посмотрите, как маленький ребёнок учится строить башню из кубиков. Никто не даёт ему списка правильных ответов. Он ставит один кубик на другой, башня шатается, падает или стоит, и он пробует снова. Каждый ход меняет то, что он увидит дальше, и единственный приговор — устоит ли башня в конце.



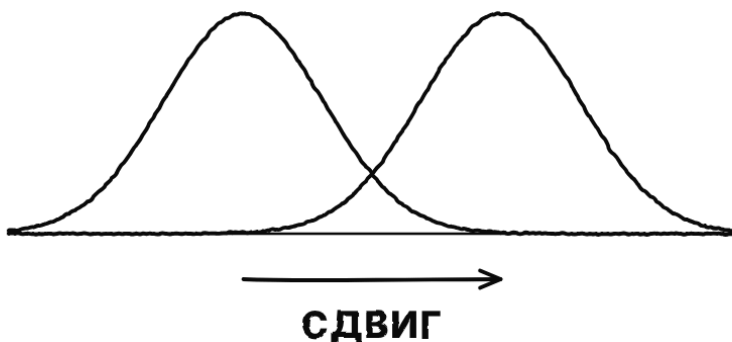
Учиться в отрыве или действовать

Всё до сих пор работало иначе. Мы заранее хватали большую грудку данных, отключали связь с миром и давали машине, ищущей закономерности, спокойно работать. Поскольку всё

обучение происходит после того, как машина отрезана от мира, это называется офлайн-обучением. В нём есть своя прелесть: можно думать о распознавании закономерностей, не беспокоясь о мире, который движется и отвечает. Но это ограничивает. Читатели рассказов Азимова о роботах представляют машины, которые не просто предсказывают, а действуют, а действия, в отличие от предсказаний, меняют мир, который машина увидит дальше. Чтобы тренировать такого агента, нужно учитывать, как его собственные действия формируют то, что будет дальше, а офлайн-обучение этого не умеет.

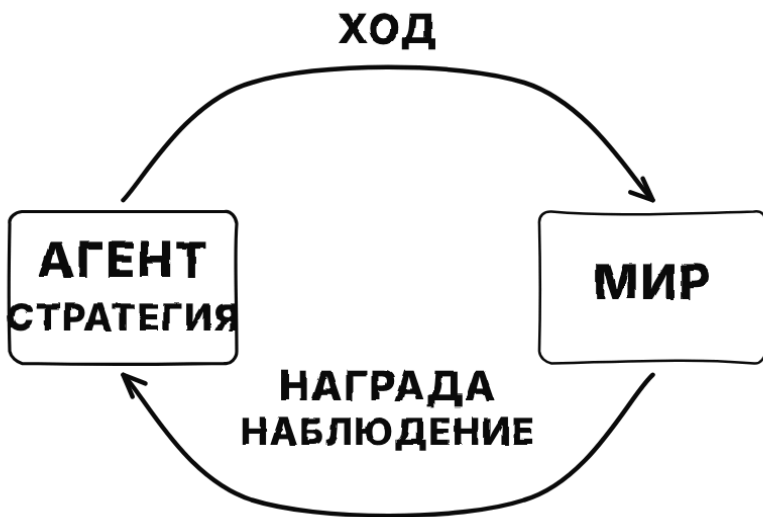
Как только мир начинает отвечать, появляются новые вопросы. Помнит ли он, что мы делали раньше? Хочет ли он нам помочь, как человек, который чётко говорит в распознаватель речи? Хочет ли он нас обыграть? Пример книги — спамеры: они всё время меняют свои письма, чтобы проскользнуть мимо вашего спам-фильтра. Меняется ли он со временем, сам или из-за наших инструментов? Всякий раз, когда данные при проверке отличаются от данных при тренировке, мы имеем дело с тем, что называется сдвигом распределения. Пример книги знаком со школы: делать домашние задания от ассистентов, а потом сдавать экзамен, составленный лектором.

ТРЕНИРОВОЧНЫЕ ТЕСТОВЫЕ



Тестовые данные могут сдвигаться

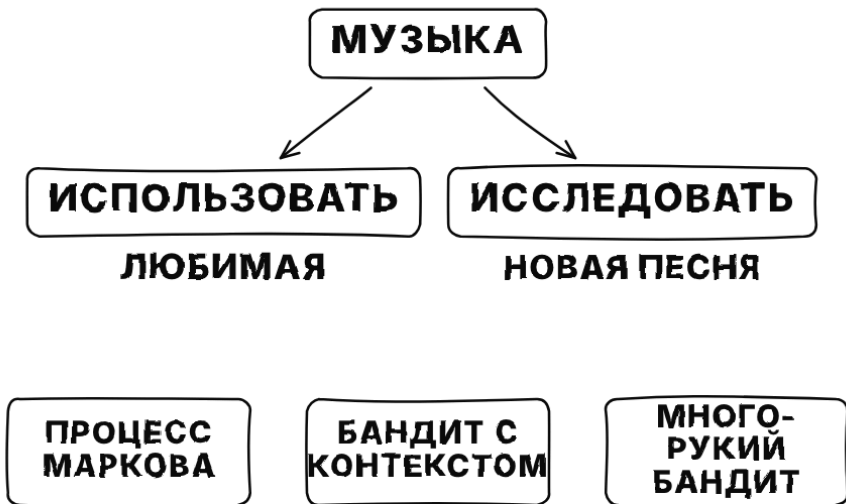
Главная схема для действующих машин — обучение с подкреплением. Его используют в роботах, в диалоговых системах и в компьютерных противниках в видеоиграх. В сочетании с глубоким обучением у него знаменитые успехи: программа, которая научилась обыгрывать людей в старые аркадные игры Atari, глядя только на экран, и AlphaGo, победившая чемпиона мира в настольной игре го. Картина — это цикл. На каждом шаге агент наблюдает мир, выбирает действие, и мир отвечает, иногда наградой; затем агент снова наблюдает, и так далее. Способ агента выбирать действия по тому, что он видит, называется его стратегией, и цель обучения с подкреплением — найти хорошую стратегию.



Цикл обучения с подкреплением

Схема чрезвычайно общая; даже обычную классификацию можно переодеть в обучение с подкреплением, с одним действием на каждый класс и наградой, зависящей от того, насколько верен выбор. Но оно может гораздо больше, чем обучение с учителем, потому что мир редко сообщает агенту лучшее действие. Он даёт только награду, а часто даже не говорит, какое действие её заслужило. В шахматах единственный настоящий приговор — в конце: победа или поражение. Какому из сорока ходов принадлежит заслуга или вина? Это задача распределения заслуг, и люди тоже с ней сталкиваются: тому, кого повысили на работе, приходится разбираться, какие из решений за прошедший год на самом деле к этому привели.

Есть и другие трудности. Агент может видеть лишь часть ситуации, как робот-уборщик, застрявший в одном из нескольких одинаковых чуланов, которому нужно помнить, как он туда попал, чтобы знать, где он. И он должен выбирать между лучшей известной ему стратегией и попыткой чего-то нового, что может оказаться лучше, жертвуя частью награды сейчас ради знания. Это компромисс между использованием и исследованием. Ваше музыкальное приложение сталкивается с ним каждый день: поставить песни, которые вам точно нравятся, или рискнуть новой, которую вы можете полюбить или пропустить.



Использовать или исследовать; более простые случаи

Браться за всё это сразу, возможно, слишком много, да и не в каждой реальной задаче всё это есть, поэтому исследователи изучают более простые частные случаи. Когда агент видит всю ситуацию целиком, задача называется марковским процессом принятия решений. Когда ситуация не зависит от того, что агент делал раньше, это задача контекстного бандита. А когда ситуации нет вовсе, а есть только ряд возможных действий с неизвестными наградами, как ряд игровых автоматов, это классическая задача многорукого бандита.

Запомним

Офлайн-обучение сначала собирает данные, а потом учится, будучи отрезанным от мира; действующий агент меняет мир, который увидит дальше. Когда данные при проверке отличаются от тренировочных, как когда экзамен составляет не тот, кто задавал домашние задания, это сдвиг распределения. Обучение с подкреплением — это цикл наблюдений, действий и получения наград, и его цель — хорошая стратегия выбора действий. Его самые трудные задачи — понять, какие действия заслужили позднюю награду, видеть лишь часть ситуации и балансировать между исследованием и использованием. Более простые частные случаи — марковский процесс принятия решений, контекстный бандит и многорукий бандит.

Вопросы

1. Почему офлайн-обучения недостаточно, чтобы тренировать действующего агента?
 - a. Потому что офлайн-обучение не может использовать большие наборы данных
 - b. Потому что действия агента меняют то, что он увидит дальше, а фиксированная груда данных этого не отражает
 - c. Потому что агенты никогда не получают наград
 - d. Потому что офлайн-обучение работает только с текстом
2. Домашние задания задают ассистенты, а экзамен пишет лектор. Какую задачу машинного обучения это иллюстрирует?
 - a. Распределение заслуг
 - b. Исследование
 - c. Сдвиг распределения: тестовые данные отличаются от тренировочных
 - d. Многорукого бандита
3. Шахматная программа выигрывает партию. Что трудно в обучении по этой одной награде?
 - a. Понять, какие из её многих ходов заслуживают заслуги
 - b. Узнать, выиграла ли она
 - c. Увидеть доску
 - d. Выбрать соперника
4. Ваше музыкальное приложение иногда ставит песню, которую вы никогда не слышали, вместо знакомой любимой. Зачем оно может так делать?

- a. Потому что оно забыло ваши любимые песни
- b. Потому что у него нет стратегии
- c. Потому что та песня популярнее
- d. Потому что оно жертвует частью верной награды сейчас ради знания, которое может найти что-то лучше

Ответы

1. **b)** Чтобы тренировать агента, нужно учитывать, как его собственные действия формируют то, что будет дальше, а офлайн-обучение этого не умеет.
2. **c)** Всякий раз, когда данные при проверке отличаются от данных при тренировке, мы имеем дело со сдвигом распределения; домашние задания и экзамен — пример из книги.
3. **a)** Единственный настоящий приговор приходит в конце, так что программе нужно понять, какие ходы заслуживают заслуги или вины: это задача распределения заслуг.
4. **d)** Оно должно выбирать между лучшей известной стратегией и попыткой чего-то нового: это компромисс между использованием и исследованием.

Задачи

1. Опишите цикл обучения с подкреплением для робота-пылесоса: каковы его наблюдения, действия и награды и какова его стратегия?

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

Ответ: Робот наблюдает окружение через свои датчики, выбирает действия, например ехать вперёд или поворачивать, и получает награды, например за собранную грязь, или штрафы за

столкновения. Его стратегия — правило выбора действия по тому, что он сейчас наблюдает, и обучение стремится сделать эту стратегию хорошей.

2. Робот-уборщик застрял в одном из нескольких одинаковых чуланов. Объясните, почему он не может понять, где находится, по тому, что видит прямо сейчас, и чем он мог бы воспользоваться вместо этого.

Подсказка: Как выглядит каждый чулан изнутри?

Ответ: Изнутри все чуланы выглядят одинаково, так что текущий вид не говорит ему, в каком он; он видит лишь часть ситуации. Ему нужно использовать то, что он видел раньше, до того как зашёл, чтобы определить своё местоположение.

Старые корни

Представьте воскресное утро пятьсот лет назад у немецкой церкви. Когда прихожане выходят, шестнадцать взрослых мужчин просят встать в ряд, пятка к носку. Общую длину их ступней измеряют и делят поровну на шестнадцать, и результат объявляют одним футом. Звучит причудливо, но это настоящий метод обучения на данных, и он описан в книге по геометрии шестнадцатого века Якоба Кёбеля.

ОБЫЧНЫЙ ФУТ

**САМАЯ
КОРОТКАЯ**

**САМАЯ
ДЛИННАЯ**



$$\text{total} \div 16$$

$$\text{rest} \div 14$$

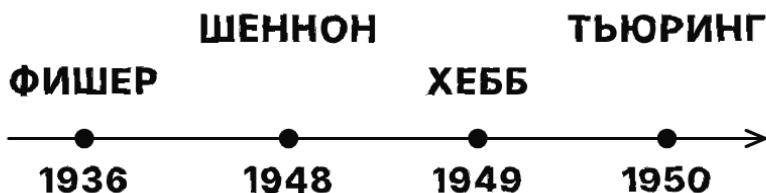
УСЕЧЁННОЕ СРЕДНЕЕ

Шестнадцать ступней, усечённое среднее

Глубокое обучение молодо, но желание учиться на данных и предсказывать, что будет дальше, очень старо, и оно лежит в основе значительной части науки и математики. Два великих математика дали свои имена закономерностям случая, которые используются повсюду до сих пор: Якоб Бернулли, около 1700 года, и Карл Фридрих Гаусс, столетием позже. Гаусс также придумал метод наименьших квадратов, который подгоняет прямую к измерениям, держа промахи маленькими, и который до сих пор применяют во всём, от страхования до медицинской диагностики. Такие инструменты отточили экспериментальную науку; закон Ома, связывающий ток через провод с напряжением на нём, — идеальное прямолинейное правило.

Измерители ступней даже улучшили свой метод. Чтобы защититься от необычных ступней, мужчин с самыми короткими и самыми длинными ступнями отправляли прочь и усредняли только остальных. Это один из самых ранних известных примеров усечённого среднего — среднего, которое не учитывает крайние значения. Некоторые системы оценок пользуются тем же приёмом, отбрасывая самые высокие и самые низкие баллы, чтобы несколько диких мнений не раскачали итог.

Статистика по-настоящему взлетела, когда данные стали собирать в больших масштабах. Один из её первопроходцев, Рональд Фишер, построил большую часть её теории и применил её к генетике, и несколько его методов до сих пор стандартные. Небольшой набор данных, опубликованный им в 1936 году, — измерения цветков ириса — до сих пор иногда используют, чтобы показать методы обучения. Фишер был также сторонником евгеники — напоминание о том, что у злоупотребления наукой о данных такая же долгая история, как и у её добрых применений.

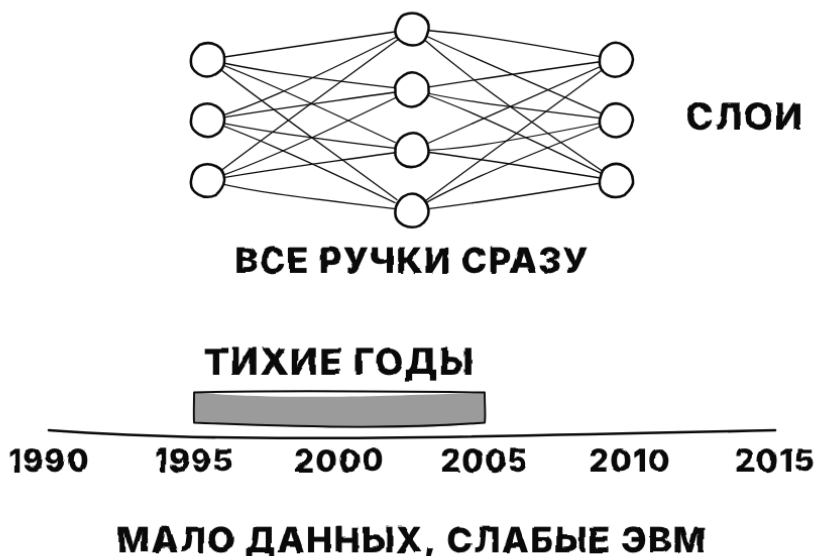


Старые корни обучения на данных

Другие корни пришли из иных мест. Клод Шеннон основал теорию информации, а Алан Тьюринг — теорию вычислений. Тьюринг спросил: «Могут ли машины мыслить?» — и предложил то, что теперь называют тестом Тьюринга: машина считается разумной, если человек, обменивающийся с ней письменными сообщениями, не может легко отличить её от человека. Сегодняшние чат-боты сделали этот старый вопрос очень ощутимым.

Изучение мозга добавило ещё больше. Психолог Дональд Хебб предположил, что связи между нервными клетками крепнут, когда ими успешно пользуются, и это правило теперь носит его имя. Эта идея привела к более поздним методам обучения и к

основному рецепту сегодняшнего глубокого обучения: усиливать то, что работает, ослаблять то, что нет. Поэтому же эти системы называют нейронными сетями, по сетям нервных клеток, хотя сходство с биологией со временем стало слабее. В их основе две идеи: слои простой обработки, сложенные один за другим, и способ подстраивать сразу все ручки всей сети, называемый обратным распространением ошибки.



Слои и тихие годы

После всплеска раннего прогресса исследования нейронных сетей застопорились примерно с 1995 по 2005 год. Их тренировка требовала огромной вычислительной мощности, которой не хватало, а наборы данных были маленькими: собрание из шестидесяти тысяч рукописных цифр считалось огромным, и

цветки Фишера 1936 года всё ещё были в ходу. При таком малом количестве данных и вычислений другие методы, с названиями вроде ядерных методов и деревьев решений, на практике работали лучше. Они к тому же тренировались часами, а не неделями, и шли с надёжными математическими гарантиями.

Запомним

Обучению на данных века, от усреднения ступней шестнадцати мужчин до наименьших квадратов Гаусса. Отбрасывание крайних значений перед усреднением, усечённое среднее, — одно из его самых ранних усовершенствований. Статистика, теория информации, теория вычислений и изучение мозга — всё это питало машинное обучение, и злоупотребление им так же старо, как и его добрые применения. Нейронные сети опираются на сложенные слои и на подстройку всех их ручек сразу, но они застопорились на десятилетие, потому что вычислений не хватало, а наборы данных были малы.

Вопросы

1. Почему измерители ступней отправляли прочь мужчин с самыми короткими и самыми длинными ступнями?
 - а. Чтобы ряд был короче
 - б. Чтобы необычные ступни не искажали среднее
 - с. Потому что нужно было только четырнадцать мужчин

- d. Чтобы фут получился длиннее
2. О чём напоминает история Рональда Фишера?
- a. О том, что статистика появилась только в двадцатом веке
 - b. О том, что маленькие наборы данных бесполезны
 - c. О том, что наукой о данных можно злоупотреблять, и это делается столько же, сколько её используют во благо
 - d. О том, что цветы — лучшие данные
3. Почему исследования нейронных сетей застопорились примерно с 1995 по 2005 год?
- a. Вычислительной мощности не хватало, а наборы данных были маленькими
 - b. Было доказано, что нейронные сети не работают
 - c. Обратное распространение ошибки ещё не изобрели
 - d. Исследователи потеряли интерес к мозгу

Ответы

1. **b)** Чтобы защититься от необычных ступней, крайних отправляли прочь и усредняли только остальных: это усечённое среднее.
2. **c)** Фишер поддерживал евгенику — напоминание о том, что у злоупотребления наукой о данных такая же долгая история, как и у её добрых применений.
3. **a)** Тренировка требовала огромной вычислительной мощности, которой не хватало, а наборы данных были маленькими.

Задачи

1. Пять судей оценивают прыжок в воду: 6, 7, 7, 8 и 1. Объясните, почему усечённое среднее, без самой высокой и самой низкой оценки, даёт более справедливый результат, чем обычное среднее, и скажите, какие оценки оно оставляет.

Подсказка: Какая оценка выбивается из остальных?

Ответ: Оценка 1 сильно выбивается и потянула бы обычное среднее вниз. Если отбросить самую высокую и самую низкую, остаются 7, 7 и 7, так что усечённое среднее — 7, и оно отражает мнение большинства судей.

2. Объясните своими словами, что проверяет тест Тьюринга.

Подсказка: Представьте разговор, который ведётся только письменно.

Ответ: Человек обменивается письменными сообщениями с кем-то, кто может быть машиной или человеком. Если человек не может легко отличить ответы машины от человеческих, Тьюринг предлагал считать машину разумной.

Великое ускорение

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 сюрприз: чипы, созданные, чтобы игры хорошо выглядели, оказались именно тем, чего ждали нейронные сети. Это одна из нитей в истории того, как глубокое обучение из застоя оказалось повсюду.

РОСТ С 1970 ГОДА

ДАННЫЕ



ВЫЧИСЛЕНИЯ



ПАМЯТЬ



Что выросло и насколько

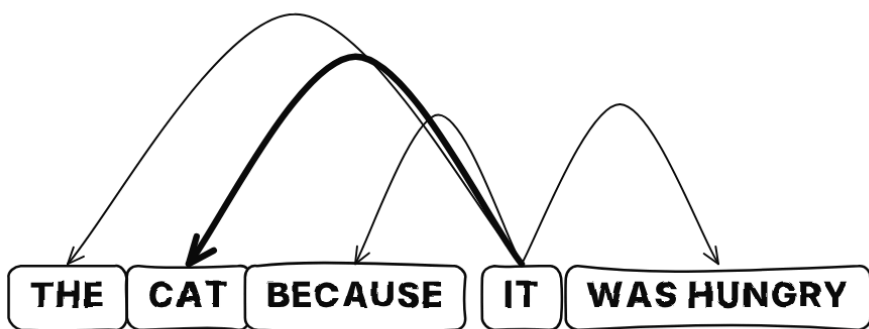
Перемены начались с данных. Интернет, компании, обслуживающие онлайн сотни миллионов людей, дешёвые и хорошие датчики во всём и всё дешевеющие хранилища вместе дали данные в количествах, каких никто не видел. Одновременно вычисления продолжали дешеветь, а графические чипы, созданные для компьютерных игр, — графические процессоры, GPU, — оказались необычайно быстрыми в арифметике нейронных сетей. Внезапно методы, казавшиеся безнадёжно медленными, стали доступны.

Книга показывает это таблицей, десятилетие за десятилетием. В 1970 году типичным набором данных была сотня с небольшим цветков ириса, а у компьютера была крошечная память, и он делал около ста тысяч вычислений в секунду. К 2020 году набором данных могла быть целая социальная сеть, триллионы объектов, а одна машина, полная графических чипов, могла делать примерно миллион миллиардов вычислений в секунду. Память тоже выросла, но гораздо меньше, чем данные, а вычислительная мощность выросла даже быстрее данных. Поэтому побеждают методы, которые бережливы с памятью, но охотно тратят много вычислений на настройку своих ручек, и это дало глубоким нейронным сетям преимущество над старыми методами. Поэтому же несколько старых идей, от простых многослойных сетей до сетей, запоминающих последовательности, были заново открыты после лет на полке.

Но прогресс не сводился к старым рецептам на более быстрых машинах; книга сравнивает последнее десятилетие с кембрийским взрывом — всплеском, когда животная жизнь вдруг стала разнообразной. Новые идеи посыпались одна за другой. Одна, называемая дропаутом, борется с переобучением,

намеренно добавляя случайный шум внутрь сети, пока она учится. Другая, внимание, позволяет модели оглядываться ровно на ту часть входа, которая ей нужна, вместо того чтобы втискивать целое предложение в память перед переводом, и длинные предложения стали переводиться гораздо точнее.

ВНИМАНИЕ



ТРАНСФОРМЕР: ТОЛЬКО ВНИМАНИЕ

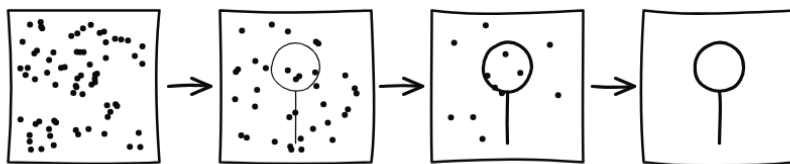
Внимание оглядывается назад

Построенная целиком на внимании, архитектура «Трансформер» становится всё лучше по мере роста данных, модели и вычислений и теперь работает для языка, картинок, речи и многого другого. Один и тот же Трансформер даже обучили играть в игры Atari, описывать изображения, разговаривать и управлять роботом. Языковые модели, которые предсказывают текст по другому тексту, обретали всё новые способности по мере

увеличения. Чат-бот, которому вы пишете, — одна из них: книга называет ChatGPT, языковую модель, настроенную следовать тому, чего хотят люди, которая может помочь исправить программный код или написать с вами рассказ.

Генеративные модели тоже совершили скачки. Генеративно-состязательные сети сталкивают две сети друг с другом: одна делает поддельные примеры, другая пытается отличить подделку от настоящего, и создатель совершенствуется, пока проверяющий не будет обманут, — так получаются убедительные лица людей, которых никогда не было. Позже диффузионные модели научились обрабатывать постепенное добавление шума, выстраивая картинку шаг за шагом из помех. Генераторы изображений, рисующие картинку по вашему описанию, например названные в книге DALL-E 2 и Imagen, работают так.

ДИФФУЗИЯ



УБРАТЬ ШУМ

Картинка из шума

Инженерия значила не меньше идей. Когда одного графического чипа мало, тренировку можно разделить между сотнями или тысячами чипов, работающих вместе. Ту когда-то стандартную задачу распознавания изображений, тренировка для которой занимала дни, так сократили до менее чем семи минут. Запуск множества копий симулированного мира параллельно тоже позволил обучению с подкреплением обыграть лучших людей в го, в игры Atari и в стратегию StarCraft, потому что такие ученики лучше всего совершенствуются, когда могут пробовать миллионы раз.

Наконец, программное обеспечение сделало всё это доступным каждому. Бесплатные общие инструментарии для построения

нейронных сетей быстро распространяли идеи; у первого поколения были названия вроде Caffe, Torch и Theano, за ними последовали TensorFlow и другие. Новейший вид, например PyTorch и JAX, позволяет писать модели в том же естественном стиле, что и обычные научные программы на Python, и это сильно облегчило эксперименты.

Запомним

Глубокое обучение возродилось, когда огромные наборы данных встретились с дешёвыми хранилищами и быстрыми вычислениями, прежде всего с графическими чипами, созданными для игр. Вычисления росли быстрее данных, а данные — быстрее памяти, и это дало преимущество глубоким сетям и вернуло старые идеи. Новые идеи — дропаут, внимание, Трансформер, языковые модели, состязательные сети и диффузионные модели — вызвали всплеск прогресса. Тренировку распределили по множеству чипов и множеству симуляций, а бесплатные инструментариумы вроде PyTorch сделали методы доступными каждому.

Вопросы

1. Почему графические чипы, созданные для игр, так много значили для глубокого обучения?

- a. Они необычайно быстро выполняют арифметику нейронных сетей
 - b. Они хранят больше данных, чем обычная память
 - c. Они были дешевле любого другого компьютера
 - d. К ним прилагались бесплатные программы
2. Вычислительная мощность росла быстрее данных, а данные — быстрее памяти. Каким методам это дало преимущество?
- a. Методам, которые хранят каждый пример в памяти
 - b. Методам, бережливым с памятью, но охотно тратящим много вычислений на настройку ручек, например глубоким сетям
 - c. Методам, которым вообще не нужна тренировка
 - d. Методам, которые используют только маленькие наборы данных
3. Какую проблему решает внимание при переводе длинного предложения?
- a. Делает модель меньше
 - b. Избавляет от необходимости в тренировочных данных
 - c. Позволяет модели оглядываться на нужную часть вместо того, чтобы сначала втискивать всё предложение в память
 - d. Добавляет шум, чтобы не было переобучения
4. Как работают вместе две сети в генеративно-сопоставительной сети?
- a. Обе делают картинки, и оставляют лучшую

- b. Одна переводит, другая проверяет грамматику
- c. Они делят тренировочные данные между собой
- d. Одна делает подделки, другая пытается их распознать, и создатель совершенствуется, пока проверяющий не будет обманут

Ответы

1. **a)** Графические чипы, созданные для игр, оказались необычайно быстрыми в арифметике нейронных сетей.
2. **b)** Побеждают методы, бережливые с памятью, но охотно тратящие много вычислений на настройку своих ручек, и это дало преимущество глубоким нейронным сетям.
3. **c)** Внимание позволяет модели оглядываться ровно на ту часть входа, которая ей нужна, вместо того чтобы втискивать целое предложение в память перед переводом.
4. **d)** Одна сеть делает поддельные примеры, другая пытается отличить подделку от настоящего, и создатель совершенствуется, пока проверяющий не будет обманут.

Задачи

1. Объясните своими словами, почему книга сравнивает последнее десятилетие глубокого обучения с кембрийским взрывом, и назовите две новые идеи, стоящие за ним.

Подсказка: Сводился ли прогресс только к более быстрым машинам со старыми методами?

Ответ: Кембрийский взрыв — внезапный всплеск новых форм жизни; в последнее десятилетие случился внезапный всплеск новых методов, а не только старые рецепты на более быстрых машинах. Примеры — дропаут, который борется с переобучением с помощью

шума, а также внимание и построенный на нём Трансформер, позволяющие моделям работать с длинными входами и улучшаться по мере роста.

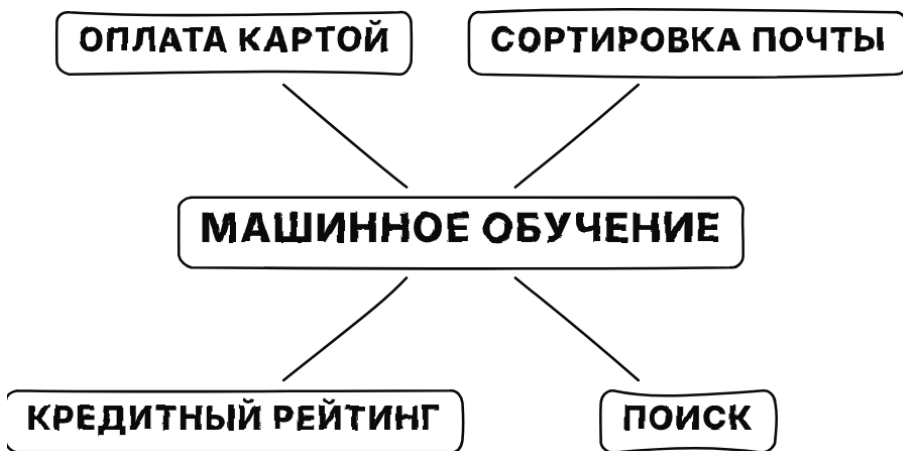
2. Почему запуск множества симулированных миров параллельно помог обучению с подкреплением обыграть людей в го и игры Atari?

Подсказка: Чего обучающемуся с подкреплением нужно очень много?

Ответ: Ученики с подкреплением совершенствуются, пробуя действия и видя награды, и им нужно огромное число попыток. Параллельно работающие симуляции дают миллионы таких попыток гораздо быстрее, чем мог бы реальный мир.

Повсюду, но почти незаметно

Заплатите картой за что-нибудь в интернете, и за мгновение до того, как платёж пройдёт, программа уже решит, не похоже ли это на мошенничество. Отправьте письмо, и машина, скорее всего, прочтает рукописный адрес. Ни то ни другое не ощущается как искусственный интеллект, но и то и другое — машинное обучение в работе, и уже много лет. Большая его часть, говорит книга, просто скрыта от глаз.



Машинное обучение вне поля зрения

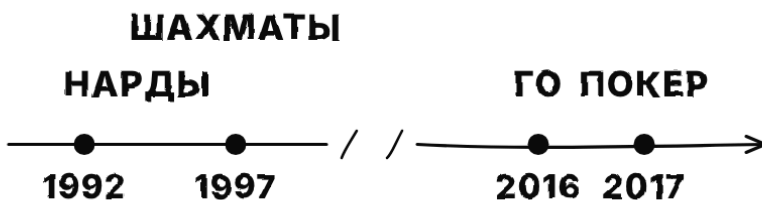
Отчасти оно распространилось так быстро потому, что инструменты стали простыми. Одни люди делают лучшие программы, другие — лучшие модели, и каждая сторона выигрывает от работы другой. В 2014 году тренировка одного довольно простого классификатора была домашним заданием, которое стоило давать новым аспирантам по машинному обучению в ведущем университете. Сегодня та же работа занимает меньше десяти строк кода и вполне по силам любому программисту.

Некоторые успехи давние. Машины сортируют почту, читая рукописные адреса, с 1990-х годов, и как раз оттуда взялось знаменитое собрание рукописных цифр. Они читают чеки, оценивают кредитоспособность людей и проверяют платежи на мошенничество за кулисами большинства крупных платёжных сервисов. Шахматные программы сильны уже десятилетиями, а машинное обучение незаметно управляет поиском, рекомендациями и личным порядком того, что вы видите в интернете.

В центр внимания искусственный интеллект недавно вывело то, что он начал решать задачи, которые люди считали невозможными, — задачи, с которыми обычные потребители сталкиваются напрямую, и за многими из них стоит глубокое обучение. Голосовые помощники вроде Siri, Alexa и помощника Google вполне сносно отвечают на устные просьбы, от включения света до записи на стрижку, и это, пожалуй, самый заметный признак искусственного интеллекта в повседневной жизни. Их ключевой ингредиент, распознавание речи, для некоторых применений стал таким же точным, как люди. Распознавание объектов на фотографиях резко улучшилось: на стандартной

проверке доля фото, для которых правильного ответа не было даже среди пяти лучших догадок модели, упала с более чем четверти в 2010 году до чуть больше двух из сотни к 2017-му. Похожие скачки произошли в определении птиц по их пению и в диагностике рака кожи.

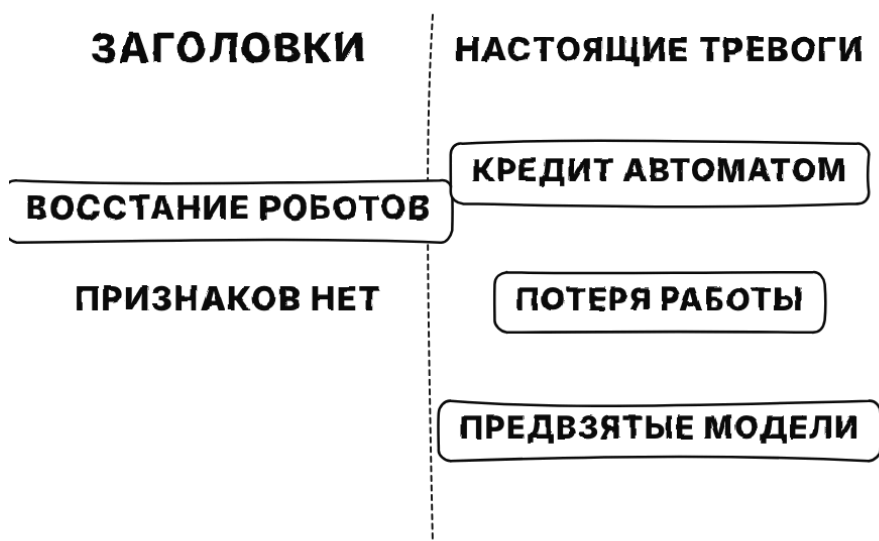
ИГРЫ, ВЫИГРАННЫЕ МАШИНАМИ



Игры, в которых машины научились побеждать

Игры раньше были мерилom человеческих способностей, и машины поднимались по этой шкале ступень за ступенью. Программа научилась нардам с помощью обучения с подкреплением; Деер Блэу обыграл чемпиона мира по шахматам Гарри Каспарова благодаря специальному оборудованию и быстрому перебору возможных ходов. Го, с его неизмеримо большим числом позиций, пала перед AlphaGo, которая

соединила глубокое обучение с хитрым способом выборки возможных вариантов будущего. Покер был трудным, потому что не видно карт других игроков, и всё же программа под названием Libratus обыграла профессионалов. Беспилотные автомобили тоже сильно продвинулись: некоторые машины уже часть времени ведут себя сами, хотя полная автономность пока недостижима: вождение требует видеть, рассуждать и следовать правилам, а глубокое обучение в основном отвечает за зрение. Робототехника, логистика, биология, физика частиц и астрономия тоже обязаны машинному обучению недавними успехами.



Страхи из заголовков и настоящие тревоги

Популярные статьи часто тревожатся о грядущем восстании роботов — о машинах, которые обретут сознание и решат пойти против нас. Авторы на этот счёт спокойны. Современные системы строят и тренируют под конкретные цели, а за любой видимостью общего интеллекта стоит смесь правил, эмпирических приёмов и статистических моделей. Инструментов для общего интеллекта, который мог бы понять сам себя и перестроить себя для решения любой задачи, говорят они, просто ещё нет.

Настоящие тревоги ближе. Искусственный интеллект уже влияет на жизнь людей: кредитоспособность оценивают автоматически, автопилоты выполняют большую часть работы по управлению транспортом, а решения об освобождении под залог опираются на статистические данные. Многие рутинные работы, которые делают люди, можно автоматизировать и автоматизируют; сельскохозяйственные роботы могут удешевить органические продукты, но и заменить сборщиков урожая, а поскольку простым ручным трудом во многих странах занято огромное число людей, последствия для общества могут быть глубокими. А модели, применяемые небрежно, могут быть предвзятыми по признаку расы, пола или возраста, что особенно тревожно, когда они помогают принимать важные решения. Как модель найма из одной из прошлых частей, система, решающая, кому дать кредит, может незаметно скопировать несправедливость прошлого. Авторы считают эти опасения гораздо более насущными, чем злой сверхинтеллект, и призывают пользоваться такими системами осторожно.

Запомним

Машинное обучение десятилетиями тихо работает, сортируя почту, ловя мошенников и упорядочивая то, что мы видим в интернете. Простые программные инструменты теперь позволяют любому программисту сделать то, что когда-то было упражнением для аспирантов. Глубокое обучение принесло заметные скачки: голосовые помощники, распознавание речи и изображений, диагностика рака кожи и победы в го и покере. Авторы не видят признаков восстания сознательных машин; настоящие тревоги — потерянные рабочие места и предвзятые решения, которые требуют осторожного применения.

Вопросы

1. Почему книга говорит, что машинное обучение часто скрыто от глаз?
 - a. Потому что компании по закону держат его в секрете
 - b. Потому что оно тихо работает в повседневных системах вроде проверки на мошенничество и сортировки почты, которые не ощущаются как ИИ
 - c. Потому что оно работает только по ночам
 - d. Потому что у него пока нет практических применений
2. Чем в одном отношении покер был для машин труднее шахмат?

- a. В покере больше правил
 - b. Партии в покер длятся дольше
 - c. Игроки не видят карт друг друга, так что ситуация видна лишь отчасти
 - d. Для покера нужно специальное оборудование
3. Какое опасение авторы считают самым насущным?
- a. Потерянные рабочие места и предвзятые автоматические решения
 - b. Сознательную машину, которая решит пойти против человечества
 - c. Нехватку памяти у компьютеров
 - d. То, что игры станут слишком лёгкими

Ответы

1. **b)** Проверки на мошенничество и чтение адресов не ощущаются как искусственный интеллект, но и то и другое — машинное обучение в работе: большая его часть скрыта от глаз.
2. **c)** Покер был трудным, потому что не видно карт других игроков.
3. **a)** Авторы считают автоматизацию работы и предвзятость гораздо более насущными, чем злой сверхинтеллект.

Задачи

1. Назовите три способа, которыми машинное обучение незаметно влияет на ваш день, используя примеры из лекции.

Подсказка: Подумайте о платежах, почте и том, что вы видите в интернете.

Ответ: Платёж картой автоматически проверяется на мошенничество; рукописный адрес на письме может прочитать машина; а результаты поиска, рекомендации и порядок того, что вы видите в интернете, задаёт машинное обучение.

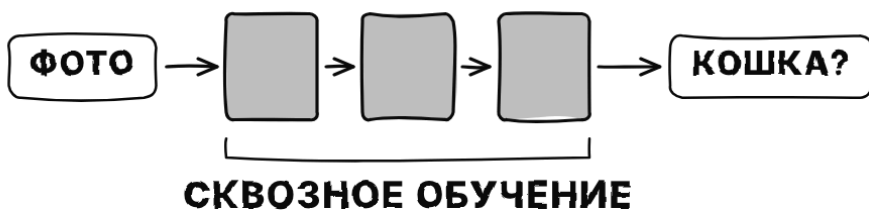
2. Объясните своими словами, почему авторы не боятся восстания сознательного ИИ, но тревожатся из-за предвзятости.

Подсказка: Из чего на самом деле построены сегодняшние системы и что происходит, когда ими пользуются небрежно?

Ответ: Сегодняшние системы строят и тренируют под конкретные цели из правил, эмпирических приёмов и статистических моделей, и инструментов для машины, способной понять и перестроить себя, нет. Но небрежно применяемые модели могут быть предвзятыми по признаку расы, пола или возраста, и когда они помогают решать вопросы о кредитах, залоге или работе, эта несправедливость вредит реальным людям уже сейчас.

Что делает его глубоким

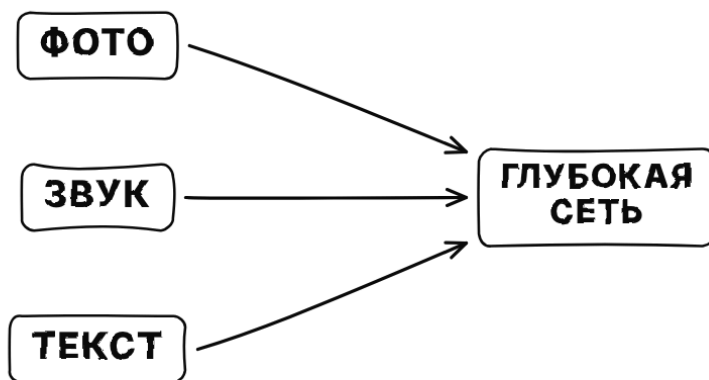
Представьте, что учите робота узнавать кошек по-старому. Сначала вы, умный человек, пишете правила, чтобы находить на фото края, углы и участки шерсти. Потом передаёте найденное простой обучающейся машине, которая решает: «кошка» или «не кошка». Если ваш искатель краёв упустит что-то важное, никакое обучение в конце этого не восполнит. Глубокое обучение отбрасывает это разделение, и в этом суть его отличия.



Ручная обработка или сквозное обучение

Глубокое обучение — это часть машинного обучения, построенная на нейронных сетях со многими слоями. Оно глубокое именно в этом смысле: его модели выучивают много слоёв шагов, каждый из которых немного преобразует данные и передаёт их дальше. В каком-то смысле у любого машинного обучения несколько стадий, и первая обычно — какая-то подготовка признаков. Глубокое обучение выделяется тем, что каждый из его многих слоёв выучивается, все вместе, из данных.

Поэтому оно блистает именно там, где спотыкались старые методы: сырой звук, сырые пиксели фотографий и предложения любой длины, которые нужно перевести на другой язык. Книга выделяет одну общую черту большинства методов глубокого обучения — так называемую сквозную тренировку. Вместо того чтобы собирать систему из частей, каждую из которых настраивают отдельно, вы строите всё целиком и настраиваете всё вместе. В компьютерном зрении больше десяти лет лучшие результаты давали спроектированные вручную искатели признаков — знаменитые детекторы краёв и тому подобное, — которые превращали изображения в списки чисел для простой модели. Решающим умением было придумывать хитрые способы готовить данные вручную. Но человеческая изобретательность мало что может против метода, который автоматически перебирает миллионы вариантов, и когда глубокое обучение взяло верх, ручные искатели признаков сменились фильтрами, которые настраивает сама машина, — с лучшей точностью.



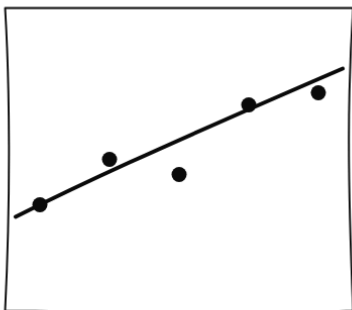
Один вид сети для всех областей

Так что глубокое обучение заменяет не только простую модель в конце, но и долгую ручную подготовку перед ней. А поскольку так много в этой подготовке было свойственно одной области, стены между зрением, речью, языком, медицинскими данными и другими областями в основном рухнули; одни и те же инструменты теперь служат им всем. Поэтому один и тот же вид модели может стоять и за голосовым помощником, который вас слышит, и за чат-ботом, который вам отвечает.

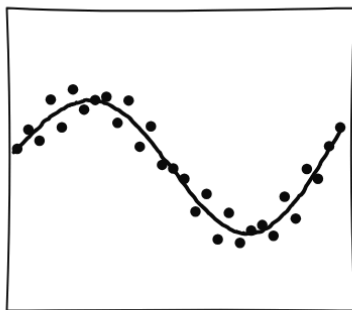
Есть и более глубокие сдвиги. Когда данных мало, моделям приходится опираться на упрощающие допущения о мире; когда данных много, они могут свободнее следовать данным. Книга сравнивает это с физикой прошлого века: когда появились

компьютеры, физики смогли напрямую моделировать поведение электронов, вместо того чтобы выводить упрощённые приближения вручную, — выиграв в точности, но часто потеряв часть понимания того, почему. Глубокое обучение также принимает решения хорошие, а не доказуемо лучшие, и готово пробовать идеи до того, как они доказаны, и этот практический дух вместе с наплывом новых талантов обеспечил быстрый прогресс.

МАЛО: ДОПУСКАТЬ



МНОГО: СЛЕДОВАТЬ



Меньше допущений при обилии данных

Наконец, сообщество гордится тем, что делится: инструменты, модели и обученные сети свободно выпускают университеты и компании. В этом же духе сама книга бесплатна для чтения, распространения и использования, и авторы надеются, что она

снижает порог для каждого, кто хочет учиться. На этом наша экскурсия по введению завершается: от правил, которые никогда не учатся, через многие виды задач обучения — к данным, чипам и общим инструментам, которые сделали глубокое обучение возможным.

Запомним

Машинное обучение — о системах, которые с опытом, обычно на данных, становятся лучше в какой-то задаче. Глубокое обучение использует нейронные сети со многими слоями, которые все выучиваются вместе из данных, сквозным образом. Оно заменило и простые модели, и ручную подготовку данных и объединило зрение, речь и язык одним набором инструментов. Его прогресс обеспечили обилие данных, быстрые графические чипы и бесплатные эффективные фреймворки, вместе с практической готовностью пробовать раньше, чем доказывать.

Вопросы

1. Чем глубокое обучение отличается от другого машинного обучения, у которого тоже несколько стадий?
 - a. Оно использует больше данных, чем любой другой метод
 - b. Каждый из его многих слоёв выучивается, все вместе, из данных

- c. Ему не нужны компьютерные чипы
 - d. Оно работает только с изображениями
2. Почему в компьютерном зрении фильтры, настроенные машиной, обыграли спроектированные вручную искатели признаков?
- a. Потому что человеческая изобретательность не может сравниться с методом, который автоматически перебирает миллионы вариантов
 - b. Потому что ручные искатели были запрещены
 - c. Потому что фильтрам не нужны были данные
 - d. Потому что края на фото не важны
3. Почему стены между зрением, речью и языком в основном рухнули?
- a. Потому что все три используют одни и те же данные
 - b. Потому что компьютеры стали быстрее
 - c. Потому что глубокое обучение заменило ручную подготовку данных, свойственную каждой области, и теперь им всем служат одни и те же инструменты
 - d. Потому что исследователи объединили свои университеты

Ответы

1. b) У любого машинного обучения есть стадии, но в глубоком обучении каждый из его многих слоёв выучивается вместе из данных.
2. a) Человеческая изобретательность мало что может против метода, который автоматически перебирает миллионы вариантов.

3. с) Поскольку так много в подготовке было свойственно одной области, её замена позволила одним и тем же инструментам служить зрению, речи, языку и не только.

Задачи

1. Объясните своими словами разницу между старым двухшаговым способом распознавать кошек и сквозной тренировкой.

Подсказка: Кто проектирует первый шаг в каждом случае?

Ответ: По-старому человек вручную придумывает правила для поиска краёв и углов, а простая обучающаяся машина выучивает только последний шаг; если ручной шаг что-то упустит, обучение этого не восполнит. При сквозной тренировке вся система, включая первые шаги, строится и настраивается вместе по данным, так что машина сама находит полезные признаки.

2. Назовите три вещи, которым курс приписывает недавний прогресс глубокого обучения, и одной фразой скажите, что дала каждая.

Подсказка: Одна — о примерах, другая — о железе, третья — о программах.

Ответ: Обилие данных из интернета и от дешёвых датчиков дало моделям достаточно примеров для обучения; быстрые графические чипы дали вычислительную мощность для тренировки больших сетей; а бесплатные эффективные фреймворки сделали построение и тренировку целых систем простыми для многих людей.