НЕЙРОСЕТИ

РОАДМАП

МАКСИМ БАЙТОВИЧ

Максим Байтович

Нейросети. Роадмап

https://litres.ru/74357313
SelfPub; 2026

Аннотация

Эта книга — ваш персональный навигатор в мире искусственного интеллекта, который меняет реальность быстрее, чем мы успеваем это осознать. Забудьте о сухих учебниках и сложных формулах: перед вами практическое руководство, написанное живым языком для тех, кто хочет оставаться востребованным завтра.

Автор проводит читателя по самому короткому и эффективному маршруту: от понимания того, как машина видит текст и «думает», до профессионального владения инструментами генерации контента и создания собственных ИИ-агентов. Вы узнаете, как превратить ChatGPT и Midjourney из игрушек в мощнейших сотрудников, как заставить нейросети работать с вашими документами и как автоматизировать рутину, освободив время для настоящего творчества.

Внутри — разбор реальных кейсов, техники профессионального промпт-инжиниринга и история фрилансера, который прошел путь от новичка до эксперта за один год.

Прочитав эту книгу, вы перестанете бояться ИИ и начнете использовать его как главное конкурентное преимущество.

Содержание

Максим Байтович
Нейросети. Роадмап

Введение. Почему именно сейчас?

В 2023 году слово «нейросеть» перестало быть термином из научной фантастики и превратилось в такое же обыденное явление, как «интернет» или «смартфон». Ещё вчера мы спорили, сможет ли машина написать связный текст, а сегодня она пишет дипломы, код, музыку и даже назначает свидания от нашего имени.

Мир изменился не за десять лет, а буквально за одну ночь — в тот момент, когда широкая публика впервые открыла ChatGPT.

И вот вы держите в руках эту книгу.

Возможно, вы:

Новичок, который слышит про ИИ отовсюду, но до сих пор не понимает, с какой стороны подойти;

Профессионал (маркетолог, юрист, врач, дизайнер), который чувствует, что нейросети вот-вот заберут его работу — или уже забирают;

Программист, который хочет не просто пользоваться чужими моделями, а создавать свои;

Или просто любопытный человек, который не хочет остаться за бортом новой технологической волны.

Эта книга — не энциклопедия и не сухой учебник по математике. Это дорожная карта (роадмап) — путеводитель по территории, на которую человечество только-только высадилось.

Почему эта книга называется «Роадмап»?

Потому что учить нейросети — это как учить иностранный язык. Можно годами зубрить грамматику (математику) и так и не заговорить. А можно выучить базовые фразы, поехать в страну (начать практиковаться) и выучить всё остальное по ходу дела.

В этой книге мы пойдём вторым путём.

Мы построим маршрут от понимания (как это работает на пальцах) через использование (как применять готовые ин-

струменты) к созданию (как строить свои модели). Мы будем двигаться от простого к сложному, но всегда — с пониманием, зачем мы делаем тот или иной шаг.

Как устроена эта книга

Часть I заложит фундамент. Мы разберёмся, что такое нейросеть на самом деле, почему она «думает» и почему иногда так глупо ошибается. Обещаю: математики будет ровно столько, сколько нужно, чтобы не бояться её всю оставшуюся жизнь.

Часть II — это практика выживания в новом мире. Вы научитесь профессионально пользоваться ChatGPT, Midjourney и другими инструментами. Вы узнаете, как заставить нейросеть работать на вас, а не против вас.

Часть III — для тех, кто хочет заглянуть под капот. Мы разберём, как обучаются большие языковые модели, что такое трансформеры и почему для обучения нужны такие дорогие видеокарты.

Часть IV — про интеграцию. Здесь мы соберём всё вместе: научимся подключать нейросети к своим проектам, делать ботов с памятью и создавать собственные приложения.

Часть V — философия. Поговорим о том, кому принад-

лежит творчество машины, заменит ли ИИ врачей и учителей, и что нам делать с тем, что профессии будущего ещё не придуманы.

Почему я пишу эту книгу

Я — не просто энтузиаст, я — практик. Я проходил этот путь сам: от первого удивления «Ого, оно разговаривает!» до построения сложных ML-систем. Я помню, как тонул в море терминов, как не мог понять, с чего начать, и как злился на то, что все объяснения либо слишком детские, либо слишком научные.

Эта книга — то, что я хотел бы прочитать сам два года назад.

Читайте как удобно

Вы можете читать книгу последовательно, как роман. А можете использовать как справочник — открывать нужную главу и брать из неё только то, что нужно прямо сейчас.

Если вы новичок — начните с Главы 1 и не торопитесь.

Если вы профессионал — сразу прыгайте в Часть II, а в Часть I возвращайтесь, чтобы закрыть пробелы.

Если вы инженер — Части III и IV станут вашим полем боя.

Маленькое обещание

К концу этой книги вы не станете кандидатом наук по искусственному интеллекту. Но вы перестанете бояться нейросетей. Вы поймёте их логику. Вы сможете использовать их в работе и в жизни. И, возможно, именно вы придумаете то, о чём остальные будут писать книги через десять лет.

Поехали.

Глава 1. Что такое нейросеть и почему это не магия?

Первое знакомство: Черный ящик или очень терпеливый ученик?

Когда человек впервые открывает ChatGPT и получает осмысленный ответ на сложный вопрос, первая реакция — благоговейный ужас. Кажется, что внутри сервера сидит разумное существо или, как минимум, джинн из бутылки.

На самом деле внутри — математика. Очень, очень много математики. Но математика не скучная, а удивительная.

Представьте себе очень старательного, но невероятно глупого ученика. Этот ученик способен только на одно действие: он смотрит на примеры и пытается найти закономерности. Если показать ему миллион фотографий кошек и миллион фотографий собак, он начнет замечать: у кошек уши острее, морда короче, усы заметнее. Он не понимает, что такое кошка. Но он выучивает признаки кошки. И когда вы показываете ему новую фотографию, он с высокой вероятностью говорит: «Это похоже на то, что я уже видел. Это кошка».

Вот это и есть нейросеть. Только работает она не с интуицией, а с числами.

Нейрон: Кирпичик, из которого всё построено

Чтобы понять нейросеть, не нужно заканчивать мехмат. Достаточно понять, как работает одна маленькая деталь — искусственный нейрон.

Биология на минималках

В нашем мозгу есть клетки — нейроны. Нейрон получает сигналы от других нейронов через отростки (дендриты). Если суммарный сигнал становится достаточно сильным, нейрон «возбуждается» и передает сигнал дальше по своему аксону.

Математика на минималках

Искусственный нейрон работает так же, но с числами.

Вход: Нейрон получает несколько чисел (например, яркость пикселей на картинке). Обозначим их x1,x2,x3...x1,x2,x3....

Веса: У каждого входа есть «важность» — вес (w1,w2,w3...w1,w2,w3...). Вес может быть отрицательным (если признак, наоборот, мешает).

Сумматор: Нейрон складывает всё вместе: Сумма=x1w1+x2w2+x3w3...Сумма=x1w1+x2w2+x3w3...

Активация: Если сумма больше порога, нейрон говорит «Да!» (выдает 1). Если меньше — «Нет» (выдает 0).

Всё.

Один нейрон — это просто линейное уравнение. Это умел делать ещё калькулятор вашего дедушки.

В чём же тогда магия? Перцептрон Розенблатта

В 1958 году психолог Фрэнк Розенблатт показал, что если взять кучу таких «нейронов» и соединить их в сеть, произойдет чудо. Машина начнет обучаться.

Возьмем простую задачу: отличить красное яблоко от зеленого.

Мы даем сети два числа: цвет (x1x1) и размер (x2x2).

Изначально веса (ww) стоят случайно. Сеть ошибается в 50% случаев.

И тут вступает в игру обучение.

Мы показываем сети яблоко. Допустим, красное.

Сеть говорит: «Зеленое!».

Мы говорим сети: «Неправильно».

Алгоритм (не сама сеть, а надзиратель) смотрит, какие веса привели к ошибке, и чуть-чуть меняет их в нужную сторону.

Повторяем 10 000 раз с разными яблоками.

Со временем веса настраиваются так, что ошибок становится всё меньше. Сеть «понимает», что если число, отвечающее за красный цвет, большое — пора говорить «Красное».

Главный секрет: Нейросеть никто не программирует. Ей никто не объясняет правила «Красное — это когда число больше 100». Она сама подбирает эти правила, перебирая примеры. В этом и заключается суть машинного обучения: не «как решить задачу», а «как научиться решать задачу самому».

Почему «глубокое» обучение такое мощное?

Перцептрон мог отличать яблоко от апельсина. Но он был бесполезен, если нужно было отличить фотографию вашей бабушки от фотографии вашей собаки. Почему?

Потому что «Бабушка» и «Собака» — это не числа вроде цвета и размера. Это сложные концепции.

Решение нашли не в новых идеях (идея нейросетей стара), а в железе и данных. Появились мощные видеокарты, и мы смогли строить глубокие сети (Deep Learning).

Как это работает?
Представьте конвейер на заводе.

Первый слой (низкий уровень): Смотрит на пиксели и учится видеть линии и границы. Горизонтальная линия, вертикальная, угол в 45 градусов.

Второй слой: Собирает линии в фигуры. «О, круг!», «О, треугольник ушей!».

Третий слой: Собирает фигуры в части объектов. «О, это похоже на глаз!», «А это — на нос».

Четвертый слой: Собирает части в объекты. «Глаза + уши + морда = Кошка».

Чем глубже сеть, тем более абстрактные вещи она видит. Нейросеть буквально строит иерархию понимания.

Итак, почему это не магия?

Магия — это когда вы не знаете правил.

Нейросеть — это когда правила выучены алгоритмом, но мы можем залезть внутрь и посмотреть, какие веса стали большими, а какие маленькими.

В следующих главах мы перестанем бояться и начнем пользоваться этим инструментом. Но запомните главное из этой главы:

Нейросеть — это статистическая машина, которая учится на примерах, подбирая миллионы чисел (весов), чтобы превращать входные данные (текст, картинку) в нужный результат (ответ, подпись).

Мини-итог главы 1:

Искусственный нейрон — это очень простое устройство: умножил, сложил, сравнил с порогом.

Сеть нейронов — это универсальный «подгоняльщик» формул. Она может выучить любую сложную зависимость.

«Обучение» — это процесс автоматической подстройки весов, чтобы уменьшить количество ошибок.

Глубина (много слоев) позволяет сети видеть в данных нечто большее, чем видит человек-аналитик.

Глава 2. Математический минимум (Без паники!)

или «Почему ваш калькулятор вдруг стал гением»
Страшное слово, которое все боятся

Признаюсь честно: когда я впервые открыл учебник по машинному обучению, на второй странице меня встретила формула, похожая на заклинание из «Гарри Поттера». Там были греческие буквы, двойные суммы и значки, которые я не видел со школы. Я закрыл вкладку и пошел смотреть сериал.

Прошло несколько лет, и я понял удивительную вещь: нейросети строят не математики. Ну, то есть математики их придумали, но пользуются ими — дизайнеры, маркетологи, врачи, журналисты и даже дети. И все они прекрасно живут без высшей математики.

Но есть одно «но».

Если вы хотите не просто пользоваться нейросетями, а понимать, почему они иногда тупят, и как их приручить — вам

нужно освоить три простые идеи. Всего три. Я обещаю, что объясню их так, как объяснил бы другу за кружкой пива.

Вот они:

Градиентный спуск — как нейросеть учится на ошибках.

Матрицы — как компьютер хранит информацию.

Вероятность — почему нейросеть никогда не уверена на 100%.

Поехали.

Идея первая: Градиентный спуск, или Как слепой человек находит дно долины

Представьте, что вы стоите на склоне горы. Ночь. Туман. Вы ничего не видите. Но вам нужно спуститься вниз, в долину, где находится решение вашей задачи (например, идеальный рецепт борща).

Что вы сделаете? Вы потрогаете землю ногой. Если под ногой есть наклон влево — вы шагнете влево. Если впереди круто вниз — шагнете вперед. Вы будете делать маленькие шаги в сторону наибольшего наклона.

Рано или поздно вы окажетесь на дне.

Именно это делает нейросеть.

Только вместо горы — ландшафт ошибки. Чем выше вы находитесь — тем больше нейросеть ошибается. Чем ниже — тем она умнее. А вместо ноги — математическая операция, которая называется «производная». Она показывает, в какую сторону нужно изменить веса (те самые числа внутри нейронов), чтобы ошибка стала чуть меньше.

Важные детали, которые объясняют поведение нейросетей:

Шаг обучения (Learning Rate). Если вы будете шагать слишком широко, вы перепрыгнете долину и окажетесь на другом склоне. Ошибка не уменьшится, а увеличится. Если шагать слишком маленькими шажками — вы будете спускаться вечность, и обучение займет годы. Инженеры тратят кучу времени, чтобы подобрать правильный размер шага.

Локальный минимум (Ловушка). Иногда вы спускаетесь и попадаете в маленькую ямку на склоне горы. Вам кажется, что вы на дне. Но на самом деле настоящая долина — еще ниже, за перевалом. Нейросети иногда «застревают» в таких ямках и думают, что уже всему научились, хотя на самом де-

ле могли бы быть умнее. Поэтому иногда мы запускаем обучение заново с другой точки горы.

Итерации (Эпохи). Нейросеть не спускается за один раз. Она делает шаг, смотрит на результат, делает еще шаг. Один проход по всем данным называется «эпохой». Иногда нужно 10 эпох, иногда — 1000. Поэтому обучение нейросети занимает часы, дни, а иногда и недели.

Запомните эту картинку: Нейросеть — это слепой путник, который спускается с горы ошибок, делая маленькие шаги в сторону наибольшего наклона. Никакой магии. Просто терпение и миллионы шагов.

Идея вторая: Матрицы, или Почему фотография — это просто таблица

Когда вы смотрите на фотографию кота, вы видите кота. Когда компьютер смотрит на ту же фотографию, он видит таблицу чисел.

Серьезно.

Любая картинка — это прямоугольник, разбитый на пиксели. Каждый пиксель — это три числа: сколько в нем красного, зеленого и синего (RGB). Если картинка размером 1000×1000 пикселей, то это миллион пикселей и три милли-

она чисел.

Матрица — это просто таблица чисел. Ничего страшного. В Excel вы каждый день работаете с матрицами, даже не замечая этого.

А теперь представьте, что текст — это тоже матрица. Каждое слово — это вектор (набор чисел), который отражает его смысл. Например, слово «король» может быть закодировано так, что оно будет находиться в математическом пространстве рядом со словом «мужчина», но далеко от слова «огурец».

Компьютеры любят матрицы. Видеокарты (те самые GPU, которые нужны для нейросетей) — это машины по перемножению гигантских таблиц чисел с невероятной скоростью. Когда вы слышите, что «обучение идет на мощном железе», знайте: там просто очень быстро перемножают таблицы.

Зачем вам это знать? Чтобы не бояться слова «тензор». Тензор — это злая версия матрицы. Матрица — это прямоугольная таблица (2D). А тензор — это таблица в 3D, 4D или вообще 100D. Но суть та же: просто числа, разложенные по полочкам.

Идея третья: Вероятность, или Почему нейросеть никогда не говорит «Я не знаю»

Спросите у ChatGPT: «Какова столица Франции?»
Он ответит: «Париж» — и будет уверен.

Но на самом деле внутри него нет железобетонного факта «Париж — столица». Внутри — распределение вероятностей.

Нейросеть смотрит на ваши слова и вычисляет: с какой вероятностью следующее слово в ответе должно быть именно «Париж», а не «Лион», «Бордо» или «Крокодил».

Для слова «Париж» вероятность 99.9%.
Для слова «Лион» — 0.05%.
Для слова «Крокодил» — одна миллиардная процента.

Но она выбирает из этого списка. И иногда выбирает неправильно! Именно поэтому нейросети иногда «галлюцинируют» — выдают красивую и уверенную чушь. Они не врут специально. Просто в их вероятностной таблице слово «чушь» на долю секунды показалось им самым подходящим, и они его выдали с апломбом профессора.

Отсюда два практических вывода:

Никогда не доверяйте фактам от нейросети на 100%. Она

— генератор вероятных текстов, а не база данных. Проверяйте.

Параметр «Температура» (о котором мы поговорим позже) — это ручка настройки азарта. Если понизить температуру — сеть будет выбирать самые вероятные слова (будет скучной, но точной). Если повысить — начнет импровизировать и халлюцинировать (будет креативной, но неадекватной).

Маленькое резюме

Я прошу вас запомнить из этой главы всего три вещи:

Обучение — это слепой спуск с горы ошибок. Нейросеть делает шажок, смотрит, ошиблась ли она, и корректирует направление.

Данные — это таблицы чисел. Все ваши тексты, картинки и видео для компьютера — скучная числовая матрица.

Результат — это всегда вероятность. Нейросеть не «знает», она «предполагает». Иногда — очень удачно.

Всё, математика на сегодня закончилась. Выдыхайте.

В следующей главе мы перейдем к самому интересному — к анатомии. Мы посмотрим, как устроены разные типы нейросетей. Почему одни сети умеют видеть, другие — писать, а третьи — рисовать. Вы узнаете, чем сверточные сети отличаются от трансформеров, и почему «Внимание» — это не просто слово из заголовков статей, а механизм, который перевернул мир.

Обещаю: будет интересно. И ни одной формулы... почти.

Глава 3. Архитектуры: Анатомия мозга машины

или «Почему одни сети рисуют, а другие считают налоги»
Не все нейросети одинаково полезны

Когда в разговоре кто-то говорит: «Нейросеть сгенерировала картинку», — на самом деле за этим стоит конкретная архитектура. Это как сказать «автомобиль перевёз груз», не уточняя, был это Ferrari или КамАЗ. И то, и другое — транспорт, но задачи у них разные.

Нейросети тоже бывают разные. Одни лучше работают с таблицами, другие — с фотографиями, третьи — с текстом. И если вы хотите понимать, что происходит внутри ChatGPT или Midjourney, вам нужно познакомиться с тремя-четырьмя главными «породами».

Мы пройдемся по эволюции: от простых трудяг до гениальных болтунов.
1. Полносвязные сети: Старые добрые трудяги

Самый простой и самый старый тип. Мы с вами уже встре-

чались с ними в первой главе.

Представьте себе колонну солдат, стоящих в несколько шеренг.

Первая шеренга получает данные (например, размер дома и район).

Вторая шеренга получает сигналы от первой.

Третья — от второй.

Последняя шеренга выдает ответ (например, цену дома).

Главное правило: каждый нейрон в одной шеренге связан с каждым нейроном в следующей. Поэтому такие сети называются полносвязными.

Где они хороши?

Они прекрасно решают задачи, где данные уже разложены по полочкам: таблицы Excel, списки параметров, финансовые показатели. Если у вас есть таблица с данными о клиентах и вы хотите предсказать, кто из них уйдет к конкурентам, — полносвязная сеть справится отлично.

Где они бесполезны?

Покажите полносвязной сети фотографию кота. Она посмотрит на каждый пиксель отдельно. Пиксель 1 — серый. Пиксель 2 — серый. Пиксель 3 — белый. Но она не поймет, что эти пиксели складываются в ухо. Для неё это просто мешанина из миллиона независимых чисел.

И это главная трагедия полносвязных сетей: они не видят структуру. Не понимают, что пиксели, стоящие рядом, образуют контур. Поэтому для картинок пришлось изобрести кое-что получше.

2. Сверточные сети (CNN): Глаза машины

В 1980-х годах один умный француз по имени Ян Лекун посмотрел на полносвязные сети и сказал: «Ребята, вы неправильно смотрите на картинки». И он придумал свертку.

Как видит сверточная сеть?

Представьте, что вы смотрите на картину через маленькое окошко размером 3×3 пикселя. Вы смотрите на уголок картины, находите там диагональную линию. Затем передвигаете окошко на один пиксель вправо — опять линия. Ещё вправо — опять линия. Вы сканируете всю картину этим окошком и составляете карту линий.

Потом берете другое окошко и ищете на картине круги. Третье окошко ищет углы. Четвертое — пятна.

Потом вы берете результаты первого сканирования и сканируете их новыми окошками. Теперь вы ищете не просто линии, а комбинации линий: например, «две диагональные линии, сходящиеся под углом» — это крыша дома.

Зачем это нужно?

Потому что кошачье ухо — это не случайный набор пикселей. Это структура: изогнутая линия, которая находится в определенном месте относительно глаза. Сверточная сеть учится выхватывать эти структуры автоматически.

Где они живут сейчас?

Распознавание лиц в вашем телефоне.

Автопилот в машине (видит разметку, знаки, пешеходов).

Медицина: поиск опухолей на снимках МРТ (да, сети находят рак лучше некоторых врачей).

Но у сверточных сетей есть слабость: они видят пространство, но не видят время. Они не понимают последовательности. Поэтому для текста и речи понадобился другой зверь.

3. Рекуррентные сети (RNN): Память золотой рыбки

Когда вы читаете предложение «Я люблю свою собаку, потому что она...», — вы понимаете, что «она» — это собака. Вы помните начало предложения.

А полносвязная сеть не помнит. Она читает каждое слово как новое, с чистого листа. Ей скажешь «она» — и она спросит: «Кто — она?»

Рекуррентная сеть решает эту проблему элегантным образом:

У неё есть петля. Нейрон передает сигнал не только следующему слою, но и самому себе. Это как короткая память. Сеть получает слово «собака», кладет его себе в карман, а когда приходит слово «она» — залезает в карман и говорит: «Ага, раньше была собака, значит она — это про неё».

Проблемы с памятью

Одна беда: карман у такой сети маленький. Если вы спросите её: «В начале этого абзаца я упомянул серую кошку. О ком я говорю в последнем предложении?» — она уже забудет. Пять-десять шагов назад она ещё помнит, а вот сто — уже нет. Это называется проблемой долгой краткосрочной памяти.

Инженеры придумали улучшение — LSTM (Long Short-Term Memory). Это как рекуррентная сеть, но с отдельным механизмом «решать, что запомнить, а что выбросить». Как умный блокнот: что-то записал в черновик, что-то вычеркнул.

Рекуррентные сети долго были королями перевода и распознавания речи. Но потом пришли они.

4. Трансформеры: Революция внимания

В 2017 году группа инженеров из Google опубликовала статью с дерзким названием «Attention is All You Need» — «Внимание — это всё, что вам нужно».

Они предложили архитектуру, которая отказалась от петель и памяти. Вместо этого — механизм внимания.

Что такое внимание?

Когда вы читаете фразу: «The animal didn't cross the street because it was too tired», — как вы понимаете, к чему относится «it»? К животному или к улице?

Вы это делаете не просто запоминая предыдущие слова. Вы взвешиваете все слова одновременно. Ваш мозг смотрит

на всё предложение целиком и подсвечивает: «animal» — важно, «street» — менее важно, «tired» — важно. Вы расставляете акценты.

Трансформер делает то же самое.

Он берет предложение и для каждого слова вычисляет: «Насколько слово А связано со словом Б?»

Он не читает текст слева направо, как рекуррентная сеть. Он смотрит на весь текст одновременно и строит карту связей.

Почему это гениально?

Во-первых, скорость. Рекуррентная сеть читает по одному слову, как человек, который водит пальцем по строчке. Трансформер видит всю страницу разом. Это можно распараллелить на видеокартах, что делает обучение в сотни раз быстрее.

Во-вторых, глубина понимания. Трансформер может уловить связь между словами, которые находятся друг от друга на расстоянии в тысячу слов. Потому что он «внимателен» ко всему тексту сразу.

Кто живет внутри трансформера?

Все современные звезды — это трансформеры:

GPT (Generative Pre-trained Transformer) — та самая модель, которая пишет тексты.

BERT — модель, которая отлично понимает запросы в поисковиках.

CLIP — модель, которая связывает текст и картинки (поэтому мы можем написать «кот в космосе» и получить картинку).

Когда вы пользуетесь ChatGPT, вы общаетесь с гигантским трансформером, который прошёл через механизм внимания миллиарды раз.

5. Диффузия: Художники, которые рисуют из шума

Пару слов о том, как сети рисуют картинки. Если вы видели работы Midjourney или Stable Diffusion, вы наверняка задавались вопросом: «Откуда это берется?»

Ответ — диффузионные модели.

Идея, которая кажется безумной, но работает:

Берем чистый шум. Просто случайные пиксели. Как

«снег» на старом телевизоре.

Говорим сети: «Представь, что в этом шуме спрятан кот в космосе. Убери лишний шум и оставь кота».

Сеть начинает постепенно убирать шум. Шаг за шагом. Сначала появляется силуэт, потом контуры, потом детали. Через 50 шагов из хаоса рождается картинка.

Как сеть этому учится?

Ей показывают миллионы пар: «шумная картинка» «чистая картинка». Она учится предсказывать, что нужно убрать, чтобы остался осмысленный образ.

Так работают все современные генераторы изображений.

Итог главы: Шпаргалка

Архитектура Сильная сторона Пример использования

Полносвязные (Dense) Таблицы, числа Прогноз цен, скоринг

Сверточные (CNN) Пространство, картинки Распознавание лиц, медицина

Рекуррентные (RNN/LSTM) Последовательности Распознавание речи (раньше)

Трансформеры Текст, связи, всё сразу ChatGPT, переводчики

Диффузия Генерация из шума Midjourney, Stable

Diffusion

Теперь, когда вы слышите «нейросеть», вы можете с умным видом спросить: «А какая архитектура?» Это, кстати, хороший способ отличить дилетанта от профессионала.

Глава 4. Как нейросеть видит мир?

или «Почему для компьютера "король" — это просто стрелочка в космосе»

Странная проблема: Компьютер не понимает слов

Давайте начнём с эксперимента. Откройте любой переводчик и попросите его перевести фразу «Я хочу есть». Вы получите «I want to eat». Отлично.

А теперь спросите у того же переводчика: «Что общего между "король" и "мужчина"?»

Он зависнет. Не потому что переводчик глупый, а потому что компьютер в принципе не понимает смысла слов. Для него слово «король» — это просто набор из шести букв, кодировка символов. Ноль или единица. Он не знает, что король — это человек, что у него есть власть, что он мужчина.

А теперь представьте, что нам нужно научить нейросеть отличать хорошие отзывы от плохих. «Мне всё понравилось» — хорошо. «Это ужас» — плохо. Пока слова совпадают — всё просто. Но что если клиент напишет: «Ну вы даёте, ребята. Сервис — огонь!» Формально слово «огонь» может

быть и плохим (пожар?), а на деле — это высшая похвала.

Как объяснить компьютеру, что «огонь» — это «хорошо»?

Ответ гениален и прост: мы перестали объяснять словами. Мы начали показывать числами.

Идея, которая изменила всё: Смысл как координата

В 2013 году команда исследователей из Google предложила алгоритм Word2Vec. Идея была настолько элегантной, что сейчас её называют «революцией эмбеддингов».

Суть такая: любое слово можно представить в виде точки в многомерном пространстве.

Звучит жутко? На самом деле это как карта сокровищ.

Представьте обычную карту города. У каждой точки на карте есть две координаты: X (долгота) и Y (широта). Магазин, парк, дом — всё имеет свой адрес в этом плоском мире.

Теперь представьте, что у нас не две координаты, а три. Как в игре или в фильме. Добавляется высота. Точки больше не на плоскости, а в объёме. Можно описать самолёт, который летит над городом.

А теперь добавьте не три координаты, а триста. Или тысячу. Мы не можем это нарисовать или представить, но математически это работает. Каждая координата описывает какое-то скрытое свойство слова.

Например, у нас есть вектор (набор чисел) для слова «король». И есть вектор для слова «королева». Где-то в недрах этих чисел есть измерение 17, которое отвечает за «пол». У короля там, условно, стоит 0.9, а у королевы — -0.9.

Измерение 42 отвечает за «власть». И там у обоих будет высокое значение.

А измерение 88 — за «еду». Там у обоих почти ноль.

Как нейросеть строит эту карту?

Она ничего не знает о королях и власти. Она просто читает миллиарды предложений.

Игра «Найди соседа»:

Нейросеть видит фразу: «Король приказал...»

И фразу: «Королева приказала...»

Она замечает: слова «король» и «королева» часто встречаются в одинаковых контекстах (приказал, трон, дворец, власть). Значит, они похожи. Она ставит их рядом.

Слова «огурец» и «король» встречаются в разных кон-

текстах. Значит, между ними нужно поставить стену. Развести подальше.

Прочитав все книги мира, нейросеть строит идеальную карту языка, где:

«Король» и «королева» стоят рядом.

«Огурец» и «помидор» стоят рядом.

«Радость» и «счастье» почти в одной точке.

А «радость» и «ипотека» — на разных полюсах.

Знаменитая формула: Король минус мужчина плюс женщина равно королева

Это не шутка. Это реальное свойство таких векторных пространств.

Если вы возьмёте вектор слова «король», вычтете из него вектор слова «мужчина» и прибавите вектор слова «женщина», вы получите точку в пространстве. Если вы посмотрите, какое слово находится ближе всего к этой точке, — это будет «королева».

Компьютер не понимал логики. Он просто выучил геометрию. Он увидел, что разница между «король» и «королева» в языке точно такая же, как между «мужчина» и «женщина». Это называется лингвистическая регулярность.

Другие примеры:

«Париж» — «Франция» + «Россия» = «Москва».

«Ходить» — «Ходил» + «Говорить» = «Говорил».

Машина неосознанно выучила грамматику и семантику через географию точек. Это как если бы вы выучили иностранный язык, просто глядя на карту звёздного неба, где созвездия — это смыслы.

Но слова — это только начало. А картинки?

Через несколько лет после Word2Vec инженеры задались вопросом: если можно сделать вектор для слова, можно ли сделать вектор для фотографии кота?

Оказалось, да. И это была ещё одна революция.

CLIP (Contrastive Language–Image Pre-training) — модель, которая научилась помещать картинки и текст в одно и то же пространство.

Как это работает?

Модели показывают миллионы пар: фотография собаки + подпись «фотография собаки». Фотография заката + подпись «красивый закат».

Модель учится ставить точку картинки рядом с точкой текста.

Результат? Мы можем написать «кот в скафандре на луне», и модель нарисует точку в этом многомерном пространстве. А потом найдёт картинку, которая находится ближе всего к этому описанию.

Конец ознакомительного фрагмента.