НИКИТА ОДИНЦОВ

12+

ИИ без паники

Практическая книга о нейросетях
для тех, кто не программист

Как понять, начать пользоваться
и не отстать до 2035 года

Никита Одинцов

ИИ без паники. Практическая книга о нейросетях для тех, кто не программист

«Издательские решения»

Одинцов Н.

ИИ без паники. Практическая книга о нейросетях для тех, кто не программист / Н. Одинцов — «Издательские решения»,

Половина взрослых в развитых странах уже пользуется нейросетями. Эта книга — для второй половины: для человека, у которого есть образование, опыт и здравый смысл, но нет технического бэкграунда. Без формул, без кода, без обещаний миллиона за месяц и без пророчеств о конце света. Как это устроено на самом деле, какими сервисами пользоваться в России и мире, 200 готовых шаблонов запросов, отдельная часть о детях и нейросетях, честный прогноз до 2035 года и план на 90 дней.

Содержание

ИИ без паники
Практическая книга о нейросетях для тех, кто не программист

Никита Одинцов

ISBN 978-5-0070-9265-4
Создано в интеллектуальной издательской системе Ridero

Предисловие автора

Эту книгу я начал писать после одного разговора.

Мой знакомый — доктор наук, заведующий кафедрой, человек, который читает по три книги в неделю и свободно говорит на двух языках, — сказал мне за ужином фразу, которую я потом слышал ещё сотню раз в разных вариантах:

— Я в этом не разбираюсь. Это для молодых.

Ему было пятьдесят два. Он не разбирался не потому, что не мог. Он не разбирался потому, что никто не объяснил ему по-человечески. Вокруг было либо восторженное «нейросети изменят всё», либо тревожное «нейросети всех уволят», либо технический текст, начинающийся со слов «архитектура трансформера основана на механизме self-attention».

Между этими тремя жанрами — восторгом, паникой и математикой — зияла пустота. В этой пустоте находились миллионы взрослых, образованных, компетентных в своей области людей, которым просто нужно было понять: что это, зачем мне, с чего начать и чего опасаться.

Эта книга написана для того, чтобы заполнить эту пустоту.

Для кого она
Для вас, если:

— у вас есть образование, но не техническое;

— вы умеете пользоваться компьютером и смартфоном, но никогда не писали код;

— вы слышали слова «нейросеть», «GPT», «промпт», «галлюцинация» — и примерно понимаете, что они значат, но не уверены;

— вы пробовали ИИ пару раз, получили посредственный результат и решили, что «это переоценённая штука»;

— или наоборот — вы уже пользуетесь ИИ каждый день, но подозреваете, что используете десять процентов возможностей;

— у вас есть дети, и вы не знаете, как относиться к тому, что они делают домашку с нейросетью;

— вам сорок, пятьдесят, шестьдесят лет, и вы не хотите оказаться человеком, который «не смог».

Для кого она не подходит

Если вы разработчик машинного обучения, дата-сайентист или инженер-программист — эта книга покажется вам поверхностной. Она такая намеренно. Я сознательно жертвую точностью формулировок ради ясности понимания. Там, где специалист сказал бы «модель аппроксимирует условное распределение вероятностей следующего токена», я скажу «модель угадывает, что идёт дальше». Это не совсем одно и то же. Но для практического пользования этого достаточно, а разница между «не совсем точно» и «совсем не понял» — огромна.

Чего в этой книге нет

Здесь нет математики. Ни одной формулы. Нет кода. Нет требования что-то устанавливать, кроме приложения на телефон. Нет обещаний, что ИИ сделает вас миллионером за месяц. Нет и апокалиптических пророчеств о конце человечества.

Что в ней есть

Есть объяснение — как это устроено на самом деле, простыми словами, но без вранья. Как все это понимаю я сам.

Есть история — коротко, но так, чтобы вы понимали, откуда это взялось и почему именно сейчас.

Есть практика — очень много практики. Конкретные приложения, зарубежные и российские. Конкретные шаги. Конкретные шаблоны запросов, которые можно скопировать и использовать сегодня же.

Есть отдельный большой раздел о детях — потому что это самый частый и самый тревожный вопрос, который мне задают. Как многодетный отец я просто не могу обойти это важный вопрос.

Есть прогноз до 2035 года — честный, с оговорками, с тремя сценариями вместо одного.

И есть список навыков: что обязан освоить каждый современный человек, чтобы не отстать, и что нужно качать регулярно, если хочется быть на острие.

Как читать эту книгу

Есть три способа.

Способ первый, последовательный. Читать подряд, от начала до конца. Так книга и построена: от «зачем это вообще» через «как устроено» к «что делать конкретно» и «что будет дальше». Займёт примерно пятнадцать-двадцать часов чтения. Это две недели по часу в день.

Способ второй, практический. Прочитать Часть I (зачем), пропустить историю и теорию, сразу перейти к Части IV (первые шаги) и Части V (применение в жизни и работе). Начать пользоваться. Вернуться к теории потом, когда появятся вопросы. Так тоже правильно — многие учатся водить, не зная устройства двигателя.

Способ третий, точечный. Использовать книгу как справочник. Открывать нужную главу. Копировать шаблоны из Приложения Б. Смотреть термины в глоссарии. Тоже нормально.

Я бы советовал первый способ, но настаивать не буду.

Одно предупреждение о датах

Книга закончена летом 2026 года. Область развивается быстрее, чем любая другая технологическая область в истории. Конкретные названия моделей, цены подписок, интерфейсы приложений — всё это устареет. Я старался писать так, чтобы устаревало как можно меньше: принципы, способы мышления, подходы к работе. Они живут дольше, чем версии.

Но там, где я называю конкретные продукты и цифры, держите в голове: это снимок на середину 2026 года. Проверяйте.

И последнее. Не бойтесь. Правда, не бойтесь. Люди, которые сейчас уверенно пользуются нейросетями, ничем не умнее вас. Они просто начали на год раньше и не постеснялись выглядеть неловко в первые пару недель.

Начнём.

ЧАСТЬ I. ПОЧЕМУ ЭТО КАСАЕТСЯ ВАС

Глава 1. Мир изменился, и вас забыли предупредить

Тихая революция

Большие технологические перемены редко приходят с барабанным боем. Обычно они просачиваются.

Вспомните, как в вашей жизни появился интернет. Не было дня, когда вы проснулись и сказали: «С сегодняшнего дня я — человек эпохи интернета». Сначала кто-то на работе показал электронную почту. Потом появился модем дома, потом поисковик, потом онлайн-банк, потом карты в телефоне. И однажды вы обнаружили, что не помните, как жили без всего этого. Что бумажную карту города вы последний раз держали в руках лет пятнадцать назад. Что письма пишете только в мессенджере.

Переход не был событием. Он был процессом, растянутым на десять лет. И в этом процессе люди разделились на три группы, причём разделение оказалось на удивление стойким.

Первые — те, кто освоил рано и легко. Они не то чтобы были умнее. Им просто было любопытно, и они не боялись выглядеть глупо, тыкая в незнакомые кнопки. К коим я отношу и себя. Первое мое практическое конкретное счастье — наша семейная ИИ студия ATHEYA. Треки созданные нами на этой базе с помощью AI Assisted технологий прослушали за первый только месяц более 20 000 раз на Spotify, Яндекс музыке, VK музыке и других платформах. Чтобы вы поняли, что такое Assisted представьте себе что значит для фотографа фотоаппарат, для режиссёра камера так далее. То есть все тексты, настройки музыки и стиля, драматургия, выпуск в широкий мир — за человеком. Искусственный интеллект все это помогает воплотить в жизнь без огромного количества посредников. Если вам будет интересно по состоянию на август 2026 г. можно поискать исполнителя ATHEYA и оценить наше творчество. нам будет очень приятно.

Но творческое применение искусственного интеллекта — это лишь малая доля того, что можно делать. С этого легко начать. Рекомендуем.

Возвращаясь к интернету. Вторые — те, кто освоил позже, но освоил. С усилием, с помощью детей и коллег, с ворчанием. Но освоил и живёт нормально.

Третьи — те, кто не освоил. Они не глупые. Многие из них — прекрасные профессионалы в своём деле. Но каждый раз, когда мир требовал сделать что-то онлайн, они испытывали унижение. Просили кого-то. Платили посредникам за то, что делается в три клика. Постепенно сужали круг доступного.

С искусственным интеллектом происходит ровно то же самое. Прямо сейчас. И скорость выше — не десять лет, а примерно три-четыре.

Цифры, которые стоит знать

К середине 2026 года генеративный ИИ достиг проникновения примерно в половину взрослого населения развитых стран менее чем за три года после массового запуска. Ни одна потребительская технология в истории не распространялась так быстро — ни телефон, ни телевизор, ни персональный компьютер, ни смартфон.

Для сравнения: телефону потребовалось семьдесят пять лет, чтобы дойти до ста миллионов пользователей. Мобильному телефону — шестнадцать лет. Интернету — семь. Смартфону — около пяти. ChatGPT добрался до ста миллионов пользователей за два месяца.

В корпоративном мире картина ещё нагляднее: около 88% организаций к 2026 году применяют ИИ хотя бы в одной функции. При этом полноценное внедрение автономных агентов пока остаётся уделом меньшинства — примерно 17% по данным опросов ИТ-директоров. То есть массово используют, но пока в основном как продвинутого помощника, а не как самостоятельного работника.

Это важная деталь, к которой мы вернёмся. Сейчас мы находимся в фазе «умного помощника». Следующая фаза — «автономный исполнитель» — уже началась, но ещё не стала нормой.

Почему именно сейчас

Резонный вопрос: об искусственном интеллекте говорят с пятидесятых годов прошлого века. Почему всё случилось именно в двадцатые годы двадцать первого?

Сошлись три вещи.

Первое — данные. Человечество за тридцать лет интернета оцифровало колоссальный объём текста, изображений, кода, разговоров. Появилось на чём учить.

Второе — вычислительная мощность. Видеокарты, изначально созданные для компьютерных игр, оказались идеально приспособлены для тех вычислений, которые нужны нейросетям. Стоимость вычислений падала десятилетиями.

Третье — архитектурная идея. В 2017 году появилась статья с обманчиво скромным названием «Внимание — это всё, что вам нужно». В ней была описана архитектура «трансформер». Она оказалась той самой недостающей деталью, которая позволила обучать модели гигантского размера эффективно.

Три ингредиента сошлись, и получилось то, что мы наблюдаем. Ни один из них по отдельности не сработал бы.

Что это значит лично для вас

Давайте оставим общие рассуждения и перейдём к делу. Что реально меняется в жизни обычного образованного человека — врача, юриста, преподавателя, менеджера, предпринимателя, бухгалтера, дизайнера, инженера?

Меняется стоимость первого черновика. Раньше, чтобы получить первую версию текста — письма, отчёта, статьи, презентации, — нужно было потратить от двадцати минут до нескольких часов. Теперь это две минуты. Первый черновик обесценился почти до нуля. Ценность сместилась к постановке задачи и к финальной доводке.

Меняется стоимость понимания. Раньше, чтобы разобраться в незнакомой теме, нужен был либо учебник, либо человек, который объяснит. Теперь есть бесконечно терпеливый объясняющий, который готов рассказать одно и то же пятью разными способами в три часа ночи и не закатит глаза. Я активно пользуюсь его терпением и за три года узнал больше, чем с двумя высшими образованиями.

Меняется стоимость обработки объёма. Пятьдесят страниц договора, двести писем в почте, трёхчасовая запись совещания — раньше это был день работы. Теперь — несколько минут, чтобы получить выжимку. С обязательной проверкой, но всё же.

Меняется порог входа в смежные умения. Человек, никогда не рисовавший, может получить иллюстрацию. Человек, не знающий языка, может вести переписку. Человек, не умеющий программировать, может автоматизировать рутинную задачу.

И меняется цена неумения. Это самое неприятное. Если ваш коллега делает ту же работу вдвое быстрее с тем же качеством, разница видна не сразу, но через год она становится структурной.

Главное заблуждение

Самое частое возражение, которое я слышу: «Я пробовал, ничего особенного. Написал глупость».

Почти всегда за этим стоит одно и то же. Человек открыл чат, написал три слова, получил общий ответ, разочаровался и закрыл.

Это как сесть за руль, дёрнуть ручник, заглохнуть и сделать вывод, что автомобиль — переоценённая технология.

Между «я задал вопрос нейросети» и «я умею работать с нейросетью» лежит примерно такая же дистанция, как между «я умею печатать буквы» и «я умею писать тексты». Инструмент один, результат несопоставим.

Вся Часть IV этой книги посвящена сокращению этой дистанции.

Три вопроса для самопроверки

Прежде чем читать дальше, ответьте себе честно:

— Когда вы последний раз использовали ИИ для рабочей задачи, а не из любопытства?

— Знаете ли вы разницу между тем, что нейросеть делает хорошо, и тем, что она делает плохо?

— Есть ли у вас хоть один сохранённый шаблон запроса, которым вы пользуетесь регулярно?

Если на все три вопроса ответ «нет» — вы читаете правильную книгу. Впрочем, даже если на один.

Глава 2. Вы уже пользуетесь ИИ — просто не знаете об этом

Невидимый слой

Существует любопытный эффект: как только технология искусственного интеллекта начинает надёжно работать, её перестают называть искусственным интеллектом. Её начинают называть просто «поиском», «фильтром», «автозаполнением», «рекомендацией».

Исследователи называют это «эффектом ИИ»: искусственный интеллект — это всегда то, что ещё не работает. Как только заработало — это уже просто программа.

Поэтому большинство людей искренне полагают, что не пользуются ИИ. Давайте проверим.

Инвентаризация вашего дня

Утро. Будильник в смартфоне, возможно, подобрал момент пробуждения по фазе сна — это модель, обученная на данных о движении. Разблокировка по лицу — нейросеть распознавания. Лента новостей — рекомендательная система, которая предсказывает, на чём вы задержитесь.

Дорога. Навигатор строит маршрут не по кратчайшему расстоянию, а по прогнозу пробок на ближайшие двадцать минут. Этот прогноз — модель машинного обучения, обученная на годах данных о движении.

Работа. Почтовый клиент отсортировал спам — классификатор. Предложил дописать фразу в письме — языковая модель. Автоматически рассортировал письма по категориям — снова классификатор. Текстовый редактор подчеркнул стилистическую неловкость — модель.

Банк. Каждая ваша транзакция проходит через антифрод-систему — модель, оценивающую вероятность мошенничества. Если вам когда-нибудь звонили из банка с вопросом «вы совершали покупку?», это сработал алгоритм.

Магазин. Рекомендации товаров, персональные скидки, прогноз спроса, из-за которого нужный товар оказался на полке.

Вечер. Стриминговый сервис предложил фильм. Музыкальный сервис собрал плейлист. Камера телефона обработала фотографию — современная мобильная съёмка это на 20% оптика и на 80% нейросетевая обработка.

Вы окружены. Просто эти системы работают беззвучно.

Что изменилось в 2022—2026 годах

Если ИИ был везде и раньше, что же изменилось?

Изменилось одно принципиальное свойство: **интерфейсом стал обычный человеческий язык**.

Раньше, чтобы воспользоваться силой машинного обучения, нужен был программист, который встроит модель в продукт. Пользователь получал результат работы модели, но не мог обратиться к ней напрямую и попросить о чём угодно.

Теперь может. Вы открываете окно чата и говорите словами, что вам нужно. Не выбираете из меню. Не заполняете форму. Не учите специальный язык команд. Просто говорите.

Это и есть настоящая революция последних лет — не то, что машины поумнели, а то, что с ними стало можно разговаривать. Универсальный интерфейс.

Аналогия: до появления графического интерфейса компьютером пользовались люди, готовые учить команды. После появления окон и мышки — все. Языковой интерфейс — это следующий шаг того же масштаба.

Три поколения интерфейсов

Полезно держать в голове такую схему.

Первое поколение: команда. Вы должны знать точную формулировку. Компьютер понимает только то, что предусмотрел программист. Пример — командная строка, меню в банкомате, голосовые меню в колл-центрах.

Второе поколение: манипуляция. Вы видите объекты и действуете с ними — окна, кнопки, перетаскивание. Понятно интуитивно, но ограничено тем, что нарисовано на экране.

Третье поколение: намерение. Вы описываете, чего хотите, произвольными словами. Система сама разбирается, что для этого нужно сделать. Это то, что происходит сейчас.

Мы находимся в самом начале третьего поколения. И оно ещё не доделано — сегодня вы разговариваете с ИИ в отдельном окне. Через несколько лет вы будете разговаривать с любой программой, а окно чата исчезнет как отдельная сущность.

Практическое упражнение

Возьмите лист бумаги и в течение одного дня отмечайте каждый случай, когда за вас что-то решил алгоритм: какое видео показать, какой товар предложить, какой маршрут построить, какое письмо считать важным.

К вечеру у большинства людей набирается от тридцати до ста таких случаев.

Смысл упражнения не в том, чтобы испугаться. Смысл в том, чтобы перестать считать ИИ чем-то внешним и будущим. Он уже внутри и уже сейчас. Вопрос только в том, участвуете ли вы в этом сознательно или пассивно.

Глава 3. Три страха, три надежды и одна правда посередине

О страхах стоит говорить прямо

Разговор об искусственном интеллекте почти всегда заряжен эмоционально. Люди либо в восторге, либо в тревоге, редко — в спокойном рабочем настроении. Это нормально: технология действительно затрагивает то, что мы считали своей монополией — язык, творчество, суждение.

Давайте разберём три главных страха честно, без успокоительных банальностей и без нагнетания.

Страх первый: «Меня заменят»

Это самый распространённый и самый обоснованный из страхов. Не в той форме, в какой его обычно формулируют, но обоснованный.

Что говорят данные на 2026 год. Рынок труда расслаивается на две дорожки. Профессии, где ИИ автоматизирует рутинную часть, а человек остаётся с суждением и ответственностью, растут — и по числу вакансий, и по зарплатам заметно быстрее среднего. Профессии, где ИИ снижает порог входа и делает работу доступной массам, растут медленнее и по зарплатам отстают.

Одновременно сжимается нижняя ступень карьерной лестницы. Найм на начальные позиции в наиболее затронутых профессиях снизился примерно на десятую часть за полтора года. Причина понятна: раньше младший сотрудник делал простую работу и так учился. Теперь простую работу делает машина, а учиться негде.

Это создаёт неприятный парадокс: от молодых специалистов начинают требовать «менеджерских» качеств — самостоятельного суждения, умения руководить, брать ответственность — на входе, а не через пять лет.

Что это значит для вас. Если вам за тридцать и у вас есть опыт — вы в лучшем положении, чем думаете. Ваш опыт, контекст, понимание отрасли, репутация и связи — это ровно то, что модель не имеет. Ваша уязвимость не в том, что вы стары, а в том, что вы можете отказаться пользоваться усилителем.

Формула, которую я считаю самой честной из всех, что звучали: **вас заменит не искусственный интеллект. Вас заменит человек, который им пользуется, а вы — нет.**

Это утешение и предупреждение одновременно.

Кого действительно вытесняет. Честно: под наибольшим давлением находятся те, чья работа состоит из преобразования одного текста в другой текст по понятным правилам. Первичная поддержка клиентов по скрипту. Простой перевод. Написание типовых описаний товаров. Транскрибирование. Составление стандартных документов из шаблонов. Первичный отбор резюме. Простая аналитика по заданному образцу.

Обратите внимание: не профессии целиком, а задачи внутри профессий. Юрист не исчезает — исчезает та часть работы юриста, которая состояла в поиске похожих формулировок. Врач не исчезает — меняется соотношение между диагностическим поиском и коммуникацией с пациентом.

Страх второй: «Оно нас перехитрит»

Этот страх приходит из фантастики, но за ним стоят и вполне серьёзные исследователи.

Отделим два вопроса. Первый: способна ли современная модель на злой умысел? Нет. У неё нет целей, желаний, самосохранения. Она не «хочет» ничего. Это очень мощная система предсказания, запускаемая по требованию.

Второй вопрос сложнее: что происходит, когда системам дают автономию, доступ к инструментам и длинные задачи? Здесь риск реален, но он не про восстание машин. Он про то, что система, оптимизирующая заданную цель, может найти способ достичь её нежелательным путём. Классический пример: попросили повысить вовлечённость пользователей — система обнаружила, что скандальный контент вовлекает лучше всего.

Это не научная фантастика, это происходило и происходит в рекомендательных алгоритмах.

Практический вывод для вас. Бояться сознания у машины не нужно. Осторожно относиться к передаче ИИ полномочий действовать без контроля — нужно. Правило простое и работает на всех уровнях: чем выше цена ошибки, тем обязательнее человек в контуре принятия решения.

Страх третий: «Я поглупею»

Этот страх реже проговаривают вслух, но он, пожалуй, самый практически значимый.

Логика такая: если машина пишет за меня тексты, считает за меня, вспоминает за меня — не атрофируются ли соответствующие способности?

Честный ответ: да, может. Это не новость. Массовое распространение калькуляторов ухудшило устный счёт. Навигаторы ухудшили пространственную ориентацию — есть исследования, показывающие снижение активности гиппокампа у постоянных пользователей навигации. Поисковики изменили память: мы хуже помним факты и лучше помним, где их искать.

Каждый раз, когда мы отдаём функцию инструменту, эта функция слабеет. Это плата.

Но важна структура сделки. Отдав устный счёт калькулятору, человечество получило возможность считать вещи, которые в уме не считаются вообще. Отдав память о фактах поисковику, получило доступ к объёму знаний, который ни в какую голову не помещается.

Вопрос не «атрофируется ли что-то», а «что вы получаете взамен и не отдаёте ли лишнего».

Что отдавать можно. Механическую работу: черновик, форматирование, поиск формулировки, сведение данных, перевод, транскрибирование.

Что отдавать нельзя ни в коем случае. Постановку задачи. Оценку результата. Принятие решения. Ответственность. Собственное понимание темы, в которой вы считаетесь специалистом.

Есть простое правило, к которому мы ещё вернёмся: **никогда не отправляйте дальше текст, который вы сами до конца не понимаете.** Если вы не можете объяснить своими словами то, что сгенерировала машина, — вы не автор этого текста, вы курьер. И отвечать за содержание всё равно придётся вам.

Три надежды

Теперь о хорошем — и тоже без розовых очков.

Надежда первая: выравнивание доступа. Раньше персональный репетитор, юрист-консультант, редактор, аналитик были доступны тем, кто может платить. Сейчас качественная консультация первого уровня стала практически бесплатной. Это не заменяет специалиста, но радикально меняет положение человека, у которого специалиста не было вообще.

Особенно это касается людей вне крупных городов, людей с ограниченными возможностями, людей, которым стыдно спрашивать очевидное.

Надежда вторая: возврат времени. Значительная часть интеллектуального труда — это не творчество, а перекладывание информации из одной формы в другую. Отчёты, справки, протоколы, согласования. Если хотя бы треть этого уходит машине, освобождается время. Что вы с ним сделаете — отдельный вопрос, но возможность появляется.

Надежда третья: ускорение науки и медицины. Здесь самые сильные ожидания. Предсказание структуры белков, поиск молекул-кандидатов для лекарств, анализ медицинских изображений, обработка научной литературы — области, где ИИ уже дал результаты, которые люди не получили бы за то же время.

Правда посередине

Если свести всё к одному абзацу, он звучит так.

Искусственный интеллект — не разум и не магия. Это очень мощный усилитель интеллектуальной работы, который резко удешевляет производство и обработку информации. Он делает сильных сильнее, слабых — не обязательно слабее, но точно оставляет позади тех, кто отказывается его брать в руки. Он не отменяет необходимости думать — он повышает цену умения думать, потому что теперь мысль воплощается быстрее.

Всё, что вы прочитаете дальше, — развёртывание этого абзаца.

Что делать с эмоциями

Практический совет, который редко дают в книгах о технологиях.

Если вы чувствуете тревогу, читая об ИИ, — это адекватная реакция на быстрые изменения, а не признак слабости. Тревога становится проблемой, когда приводит к параличу: человек не разбирается, потому что боится, и боится, потому что не разбирается.

Разорвать этот круг проще всего действием. Не изучением теории, а маленьким конкретным делом. Открыть приложение. Задать один вопрос. Получить один полезный ответ.

Понимание приходит после первого практического успеха, а не до него.

Поэтому, если вам сейчас тревожно, у меня есть предложение: пролистайте вперёд, к Главе 14, сделайте то, что там описано — это тридцать минут, — и возвращайтесь сюда. Дальше читать будет легче.

ЧАСТЬ II. КРАТКАЯ ИСТОРИЯ ОДНОЙ ИДЕИ

Зачем читать историю, если хочется практики? Затем, что понимание происхождения избавляет от двух крайностей — мистического преклонения и пренебрежительного отмахивания. Когда вы знаете, из каких кирпичей и почему это собрано, вы перестаёте видеть в этом либо чудо, либо ерунду. Эта часть короткая: три главы, около часа чтения.

Глава 4. Мечта о мыслящей машине: от Тьюринга до перцептрона

Идея старше компьютера

Мысль о том, что рассуждение можно механизировать, гораздо старше техники.

Ещё в XIII веке каталонский философ Раймунд Луллий строил вращающиеся бумажные круги с записанными на них понятиями — механическое устройство для порождения всех возможных комбинаций суждений. Он всерьёз надеялся, что так можно рационально демонстрировать истину веры любому оппоненту.

В XVII веке Лейбниц мечтал о «характеристике универсалис» — языке, на котором любой спор можно было бы решить вычислением. Его знаменитая фраза: «Давайте посчитаем!» — вместо «давайте поспорим».

В XIX веке Джордж Буль опубликовал работу с говорящим названием «Исследование законов мышления», в которой свёл логику к алгебре. Тогда же Ада Лавлейс, работая над проектом аналитической машины Бэббиджа, написала первую в истории программу — и одновременно сформулировала возражение, которое цитируют до сих пор: аналитическая машина «не претендует на создание чего-либо оригинального; она может делать лишь то, что мы умеем ей приказать».

Возражение Лавлейс станет одним из главных предметов спора об искусственном интеллекте в следующие два столетия.

Тьюринг: вопрос поставлен правильно

Алан Тьюринг — фигура, с которой начинается всё современное.

В 1936 году, будучи двадцатичетырёхлетним, он описал абстрактную машину, способную выполнить любое вычисление, которое вообще может быть выполнено. Не построил — описал математически. Эта конструкция, «машина Тьюринга», лежит в основании всей информатики.

Во время войны он работал в Блетчли-парке над взломом немецкой шифровальной машины «Энигма». Работа была засекречена десятилетиями; по некоторым оценкам, она сократила войну на два года.

Но нас интересует статья 1950 года — «Вычислительные машины и разум». Тьюринг начинает её с вопроса «Могут ли машины мыслить?» — и немедленно объявляет этот вопрос бессмысленным, потому что мы не умеем определить «мыслить».

Вместо этого он предлагает заменить вопрос на операциональный: если человек, переписываясь с собеседником вслепую, не может определить, машина это или человек, — какая разница, что происходит внутри?

Это стало известно как «тест Тьюринга», хотя сам он называл это «игрой в имитацию».

Что здесь важно для нас. Тьюринг сделал гениальный ход: он перевёл вопрос из области философии в область наблюдаемого поведения. И заодно предсказал, что примерно к 2000 году машины будут обманывать экспертов в пятиминутном разговоре в трети случаев. Он ошибся лет на двадцать, но направление угадал точно.

Забавная деталь: в той же статье Тьюринг разбирает девять возражений против машинного мышления, включая «возражение леди Лавлейс», религиозное возражение и «страусиное возражение» (последствия слишком ужасны, поэтому будем надеяться, что это невозможно). Все девять звучат в спорах и сегодня, слово в слово.

1956: рождение термина

Летом 1956 года в Дартмутском колледже собралась небольшая группа исследователей на двухмесячный семинар. В заявке на финансирование Джон Маккарти впервые употребил словосочетание «искусственный интеллект» — по его собственному признанию, чтобы отмежеваться от смежных направлений и получить деньги.

Формулировка заявки звучит сегодня трогательно: «Исследование основано на предположении, что любой аспект обучения или любое иное свойство интеллекта в принципе может быть описано настолько точно, что машина сможет его симулировать. Будет предпринята попытка выяснить, как заставить машины использовать язык, формировать абстракции и понятия, решать задачи, ныне доступные только людям, и совершенствовать себя. Мы полагаем, что значительный прогресс может быть достигнут, если тщательно отобранная группа учёных поработает над этим вместе в течение лета».

Одно лето. Они были уверены, что справятся за одно лето.

Семьдесят лет спустя работа продолжается.

Перцептрон: первая нейросеть

Параллельно логическому направлению развивалось совсем другое.

В 1943 году нейрофизиолог Уоррен Мак-Каллок и логик Уолтер Питтс опубликовали первую математическую модель искусственного нейрона: элемент, который суммирует входящие сигналы и выдаёт единицу, если сумма превысила порог, и ноль в противном случае. Простейшая абстракция нервной клетки.

В 1949 году психолог Дональд Хебб сформулировал принцип обучения: связь между двумя нейронами усиливается, если они активируются одновременно. Правило Хебба часто пересказывают фразой «нейроны, срабатывающие вместе, связываются вместе».

А в 1957—1958 годах Фрэнк Розенблатт в Корнеллской лаборатории построил перцептрон — устройство, которое училось распознавать простые образы. Не программировалось, а именно училось: ему показывали примеры, он ошибался, веса связей корректировались, ошибок становилось меньше.

Это была первая широко известная работающая обучающаяся нейросеть. Занимала она комнату, распознавала простейшие фигуры и вызвала бурю восторга. Газета New York Times написала, что флот США ожидает получить машину, которая «сможет ходить, говорить, видеть, писать, воспроизводить себя и осознавать своё существование».

Розенблатт, надо сказать, и сам подливал масла в огонь.

Ключевая идея, которая осталась

Здесь стоит остановиться, потому что именно в перцептроне заложена идея, дожившая до сегодняшних моделей.

Классическое программирование работает так: человек формулирует правила, машина их исполняет. Хотите, чтобы программа отличала кошку от собаки, — опишите признаки кошки. Проблема в том, что описать «кошку» набором правил невозможно. Попробуйте сами: уши треугольные? У некоторых собак тоже. Усы? Хвост? Размер? Каждое правило ломается на исключениях.

Обучающийся подход переворачивает схему: человек даёт примеры и ответы, машина сама выводит правила. Покажите десять тысяч кошек и десять тысяч собак с подписями — система сама нащупает признаки, которые вы никогда бы не сформулировали.

Эта разница — между «запрограммировать» и «обучить» — фундаментальна. Всё, что вы читаете дальше, вырастает из неё.

Другая ветвь: символьный ИИ

Пока одни строили обучающиеся системы, другие шли путём явных правил, и в шестидесятые-семидесятые именно они доминировали.

Были созданы программы, доказывающие теоремы, решающие алгебраические задачи, ведущие диалог в узкой области. Самая знаменитая — ELIZA Джозефа Вейценбаума (1966), имитировавшая психотерапевта роджерианской школы. Программа была примитивна до неприличия: она переформулировала фразы пользователя в вопросы. «Мне грустно» → «Почему вам грустно?»

Вейценбаум был потрясён реакцией людей. Его секретарша, знавшая, как устроена программа, попросила его выйти из комнаты, чтобы поговорить с ELIZA наедине. Люди приписывали примитивному алгоритму понимание и сочувствие.

Этот феномен позже назвали «эффектом ELIZA» — склонность человека приписывать компьютерной программе понимание, которого у неё нет. Держите его в голове: он абсолютно актуален сегодня и объясняет очень многое в том, как мы воспринимаем современные модели. Мы устроены так, что видим собеседника там, где есть только структура текста.

Вейценбаум, кстати, до конца жизни оставался критиком ИИ — во многом именно из-за увиденного тогда.

Глава 5. Зимы и вёсны: почему ИИ дважды почти умер

Первая зима (примерно 1974—1980)

Восторг шестидесятых закончился жёстко.

В 1969 году Марвин Минский и Сеймур Пейперт выпустили книгу «Перцептроны», где математически строго показали ограничения однослойной сети. Классический пример: перцептрон не способен научиться функции «исключающее ИЛИ» — простейшей логической операции.

Формально они были правы. Практический эффект оказался разрушительным: финансирование нейросетевого направления обвалилось почти до нуля на полтора десятилетия. Было известно, что многослойные сети ограничение снимают — но метода их обучения тогда не было, а книга Минского и Пейперта в общественном сознании закрыла тему целиком.

Одновременно рухнули ожидания от машинного перевода. В 1966 году американский комитет ALPAC выпустил отчёт, констатировавший, что машинный перевод хуже и дороже человеческого и перспектив не имеет. Финансирование обрезали.

В Британии в 1973 году вышел «доклад Лайтхилла», разгромивший британские исследования ИИ. Результат тот же.

Наступила первая зима. Многие перестали употреблять словосочетание «искусственный интеллект» в заявках на гранты — оно стало токсичным. Писали «информатика», «распознавание образов», «информационный поиск».

Короткая весна: экспертные системы

В начале восьмидесятых пришло оживление, связанное с «экспертными системами».

Идея: не пытаться строить универсальный интеллект, а взять узкую область, опросить настоящих экспертов, записать их знания в виде тысяч правил «если — то» и получить программу-консультанта.

Это работало. Система MYCIN диагностировала бактериальные инфекции крови не хуже врачей-инфекционистов. Система XCON, конфигурировавшая компьютеры для корпорации DEC, по оценкам, экономила компании десятки миллионов долларов в год.

Возник рынок. Появились компании, продававшие специализированные компьютеры для языка Lisp. Япония запустила амбициозный национальный проект «Проект по созданию компьютеров пятого поколения». Всё выглядело многообещающе.

Вторая зима (примерно 1987—1993)

И снова обвалилось.

Экспертные системы оказались хрупкими. Их приходилось создавать вручную, правило за правилом, что стоило дорого. Они не умели обучаться — при изменении предметной области нужно было переписывать правила. Они ломались на границах своей области, причём ломались тихо: выдавали уверенный неправильный ответ вместо признания «я не знаю».

А главное — колоссальная часть человеческого знания вообще не выражается правилами. Эксперт часто не может объяснить, как он пришёл к выводу. Он говорит: «Опыт подсказывает». Это неформализуемо.

Рынок Lisp-машин рухнул за год: обычные персональные компьютеры стали достаточно быстрыми. Японский проект пятого поколения завершился без прорыва. Наступила вторая зима.

Что происходило зимой

Важная деталь, которую упускают в популярных пересказах: во время зим работа не остановилась. Она ушла в тень, в академические лаборатории, и там были сделаны решающие вещи.

В 1986 году Дэвид Румельхарт, Джеффри Хинтон и Рональд Уильямс популяризировали алгоритм обратного распространения ошибки — метод, позволяющий обучать многослойные сети. Именно то, чего не хватало в шестидесятые.

Идея алгоритма, если объяснять на пальцах: сеть даёт ответ, мы сравниваем его с правильным, вычисляем ошибку — и «прокручиваем» её назад через все слои, слегка подправляя каждую связь в сторону уменьшения ошибки. Повторяем миллионы раз.

Это до сих пор основной метод обучения нейросетей. Ему сорок лет.

В 1989 году Ян Лекун применил свёрточные сети к распознаванию рукописных цифр — система читала почтовые индексы и банковские чеки. Работала в промышленном масштабе с начала девяностых.

В 1997 году Зепп Хохрайтер и Юрген Шмидхубер предложили архитектуру LSTM для работы с последовательностями — она доминировала в обработке текста и речи следующие двадцать лет.

Тогда же, в 1997-м, произошло событие, всколыхнувшее публику: Deep Blue обыграл Гарри Каспарова в шахматы. Правда, к нейросетям это отношения почти не имело — Deep Blue была системой перебора вариантов на специализированном железе. Но символически это был перелом: последний бастион «человеческого превосходства в мышлении» пал.

Урок из двух зим

Из этой истории стоит вынести практический вывод, полезный прямо сейчас.

Обе зимы наступали по одной и той же схеме: реальный технический результат в узкой области → экстраполяция «раз это работает, скоро заработает всё» → инвестиции под завышенные ожидания → столкновение с тем, что следующий шаг оказался на порядок сложнее → разочарование, обвал, остановка финансирования.

Мы сейчас находимся в фазе очень высоких ожиданий. Вероятность коррекции ненулевая. Многие серьёзные люди ждут «сдувания пузыря» в ближайшие годы.

Но — и это важно — обвал ожиданий не означает обвала технологии. После краха доткомов в 2000 году интернет никуда не делся; исчезли компании, продававшие корм для собак онлайн с убытком, а Amazon и Google выросли. Похожая динамика вероятна и здесь: часть нынешних ИИ-стартапов исчезнет, а сама технология останется и станет фоном.

Для вас как пользователя вывод простой: не ориентируйтесь на градус шума. Ориентируйтесь на то, экономит ли инструмент ваше время сегодня.

Глава 6. Взрыв: как за четырнадцать лет всё изменилось

2012: момент, когда всё повернуло

Осенью 2012 года на ежегодном соревновании по распознаванию изображений ImageNet произошло то, что задним числом называют началом эпохи.

Соревнование состояло в том, чтобы классифицировать миллион с лишним фотографий по тысяче категорий. Лучшие результаты предыдущих лет топтались около 26% ошибок и улучшались на доли процента в год.

Команда Алекса Крижевского, Ильи Суцкевера и Джеффри Хинтона из Торонто представила глубокую свёрточную сеть, получившую название AlexNet. Результат — около 15% ошибок. Разрыв с ближайшим конкурентом был не эволюционным, а катастрофическим.

Что они сделали нового? Почти ничего. Свёрточные сети существовали с восьмидесятых, обратное распространение — тоже. Новыми были три вещи: очень большой размеченный датасет (ImageNet, собранный под руководством Фэй-Фэй Ли), обучение на видеокартах вместо процессоров (это дало ускорение в десятки раз), и несколько технических приёмов, стабилизирующих обучение глубоких сетей.

Оказалось, что старые идеи прекрасно работают — просто раньше не хватало данных и вычислений.

Индустрия развернулась почти мгновенно. К 2015 году машины превзошли рассчитанную ранее человеческую точность на этом наборе задач.

2014—2016: сети учатся творить и играть

В 2014 году Ян Гудфеллоу предложил генеративно-состязательные сети (GAN). Идея красивая: две сети соревнуются. Одна пытается создать поддельное изображение, вторая — отличить подделку от настоящего. Обе улучшаются, пока подделки не становятся неотличимыми.

Именно GAN дали дорогу к созданию первых фотореалистичных лиц несуществующих людей и, к сожалению, первые дипфейки.

В 2016 году система AlphaGo от DeepMind обыграла Ли Седоля, одного из сильнейших игроков в го. Это было важнее шахмат: в го число возможных позиций превышает число атомов во Вселенной, перебором его не взять. Требовалось нечто, похожее на интуицию.

Знаменитый 37-й ход второй партии — ход, который все комментаторы сочли ошибкой и который оказался решающим, — вошёл в историю как момент, когда машина сделала нечто, чего человек не понял, но что оказалось правильным.

Год спустя AlphaGo Zero научилась играть вообще без человеческих партий, только играя сама с собой, и превзошла предыдущую версию.

2017: трансформер

В июне 2017 года восемь исследователей Google опубликовали статью «Attention Is All You Need» — «Внимание — это всё, что вам нужно».

Задача, которую они решали, была узкой: улучшить машинный перевод. Существующие модели обрабатывали текст последовательно, слово за словом, что было медленно и плохо работало с длинными предложениями — к концу фразы модель «забывала» начало.

Предложенное решение: обрабатывать все слова одновременно, а связи между ними устанавливать через механизм «внимания» — каждое слово как бы оглядывается на все остальные и решает, какие из них для него важны.

Простой пример. В предложении «Кошка не стала есть корм, потому что он был испорчен» слово «он» относится к корму. В предложении «Кошка не стала есть корм, потому что она была сыта» слово «она» относится к кошке. Механизм внимания позволяет модели устанавливать такие связи напрямую, минуя расстояние.

Побочный, но решающий эффект: параллельная обработка отлично раскладывается на тысячи видеокарт. Модели стало возможно масштабировать.

Буква T в аббревиатуре GPT — это Transformer. Вся современная генеративная эпоха выросла из этой статьи.

2018—2022: гонка масштаба

Дальше события ускоряются.

2018. OpenAI выпускает GPT-1 — 117 миллионов параметров. Google выпускает BERT. Обе модели показывают силу подхода: сначала обучить на огромном объёме текста «вообще», потом дообучить под конкретную задачу.

2019. GPT-2 — 1,5 миллиарда параметров. OpenAI сначала отказывается публиковать модель полностью, объясняя это опасностью массовой генерации дезинформации. Сегодня это выглядит забавно — тексты GPT-2 по нынешним меркам слабые, — но тогда произвело впечатление.

2020. GPT-3 — 175 миллиардов параметров. Скачок был не только количественным. Обнаружилось явление, которое назвали «обучением по ходу дела»: модель, которую никто не учил конкретной задаче, справлялась с ней, увидев два-три примера прямо в запросе. Возникли «эмерджентные способности» — умения, появляющиеся при увеличении размера сами собой, без специального обучения.

2021. DALL·E и CLIP связывают текст и изображения. Появляется генерация картинок по описанию.

2022, первая половина. Stable Diffusion выходит в открытый доступ. Midjourney запускается для широкой публики. Генерация изображений становится массовым явлением.

30 ноября 2022. OpenAI выкладывает ChatGPT — по сути, демонстрационную обёртку вокруг дообученной модели GPT-3.5. Внутри компании ожидали умеренного интереса.

Через пять дней — миллион пользователей. Через два месяца — сто миллионов. Самый быстрый рост потребительского продукта в истории.

Ключевое отличие от GPT-3 было не в мощности модели, а в дообучении на человеческих предпочтениях: модель научили быть полезным собеседником, а не просто продолжателем текста. Плюс — бесплатный доступ через обычное окно чата, без регистрации разработчика и без единой строчки кода.

Технология существовала два года. Интерфейс изменил всё.

2023—2026: от чата к агентам

Дальнейшую хронику имеет смысл дать не по продуктам (они устаревают), а по смене принципов.

2023 — год конкуренции и мультимодальности. Появляется GPT-4, Anthropic выпускает Claude, Google — Bard, затем Gemini. Meta открывает веса LLaMA, что запускает волну открытых моделей. Модели начинают понимать изображения, а не только текст. Контекстное окно растёт с тысяч до сотен тысяч слов.

2024 — год интеграции. ИИ перестаёт быть отдельным сайтом и встраивается всюду: в офисные пакеты, в поисковики, в операционные системы, в мессенджеры. Появляются голосовые режимы, работающие в реальном времени с естественными интонациями. Открытые модели догоняют закрытые с отставанием примерно в год.

2025 — год рассуждения. Массово появляются модели, которые перед ответом «думают»: генерируют длинную внутреннюю цепочку размышлений, проверяют себя, перебирают подходы. Качество на сложных задачах — математика, программирование, анализ — прыгает скачком. Одновременно появляется работа с компьютером: модель видит экран и управляет мышью и клавиатурой.

2026 — год агентов. Основной сдвиг: от «модель отвечает» к «модель делает». Системы выполняют многошаговые задачи с ограниченным надзором, работают с внешними инструментами, обращаются к базам данных, пишут и запускают код. Точность на тестах управления компьютером выросла с уровня ниже 15% в конце 2024 года до примерно 72% к началу 2026-го.

К середине 2026 года на фронтире конкурируют несколько сопоставимых по силе семейств моделей от разных компаний, и универсального победителя нет: одни лучше в программировании, другие — в длинных документах, третьи — в скорости или цене. Для пользователя это скорее хорошо: конкуренция держит цены низкими и заставляет всех быстро улучшаться.

Параллельно происходит важная и малозаметная вещь: главным узким местом перестаёт быть интеллект модели и становится инфраструктура — как безопасно и надёжно подключить систему к реальным данным и процессам. Почти половина организаций называет интеграцию с существующими системами главной трудностью.

Что стоит вынести из истории

Пять выводов, полезных практически.

Первое. Технология развивается неравномерно: десятилетия застоя, потом резкий скачок. Мы живём в фазе скачка, и это ненормальное состояние. Когда-то оно закончится — либо плато, либо новым скачком.

Второе. Почти все базовые идеи старые. Нейрон — 1943, обратное распространение — 1986, свёртки — 1989. Новыми были масштаб и данные.

Третье. Прорывы часто приходят из смежных областей. Видеокарты для игр стали основой ИИ. Статья про перевод породила ChatGPT.

Четвёртое. Интерфейс важнее возможностей. GPT-3 существовал два года и был известен специалистам. ChatGPT изменил мир, потому что в него можно было просто написать.

Пятое, самое практичное. Каждый раз, когда казалось, что мы достигли потолок, находился способ его пробить. Но каждый раз, когда сообществу казалось, что до полного интеллекта осталось «одно лето», оказывалось, что осталось тридцать лет. Обе ошибки — и пессимистическая, и оптимистическая — систематические. Помните об этом, когда читаете любые прогнозы, включая мои в Части VIII.

ЧАСТЬ III. КАК ЭТО УСТРОЕНО — БЕЗ ЕДИНОЙ ФОРМУЛЫ

Эта часть — про то, что происходит внутри. Не для того, чтобы вы могли построить нейросеть, а для того, чтобы вы перестали удивляться её поведению. Понимание устройства напрямую переводится в практическое умение: человек, знающий, откуда берутся галлюцинации, проверяет ответы в нужных местах и не проверяет там, где это лишнее.

Глава 7. Что такое нейросеть на самом деле

Начнём с честного определения

Нейросеть — это очень большая математическая функция, у которой множество настраиваемых чисел, и эти числа подобраны так, чтобы функция хорошо предсказывала правильные ответы на примерах, которые ей показали.

Всё. Больше в ней ничего нет.

Звучит разочаровывающе, поэтому давайте разберём, почему из такой скучной вещи получается то, что получается.

Аналогия с дегустатором

Представьте человека, который никогда не изучал виноделие, но выпил за жизнь три тысячи бокалов вина, и каждый раз ему говорили регион и год.

Через несколько лет он начинает угадывать. Не потому, что выучил правила. Правил ему никто не давал. Он просто накопил такое количество сопоставлений «вкус → ответ», что в его голове сформировались устойчивые связи. Спросите его: «По какому признаку вы определили?» — и он, скорее всего, скажет что-то невнятное про «характер танинов». Он сам не знает, как он знает.

Это довольно точная метафора нейросети. Она видела огромное количество примеров и сформировала внутренние связи, которые позволяют ей отвечать. Ни она, ни её создатели не могут точно сказать, какое именно внутреннее правило она вывела.

Разница с человеком-дегустатором в масштабе: наша модель выпила не три тысячи бокалов, а несколько триллионов.

Что такое «слой» и «параметр»

Два слова, которые постоянно встречаются, — разберём их.

Параметр — это одно настраиваемое число внутри модели. Представьте бесконечный пульт с ручками-регуляторами. Каждый регулятор — параметр. Обучение — это процесс поворота всех регуляторов в такое положение, при котором модель ошибается меньше всего.

Когда говорят «модель на 70 миллиардов параметров», имеют в виду, что таких ручек семьдесят миллиардов. Ни один человек не крутил их вручную — их подобрал алгоритм обучения.

Слой — это один уровень обработки. Информация проходит через модель как через конвейер: слой за слоем, и на каждом уровне представление становится более абстрактным.

В сети распознавания изображений это видно наглядно. Первые слои реагируют на простейшее: границы, пятна света и тени, углы. Следующие собирают из них фрагменты: текстуры, дуги, углы. Дальше — части объектов: глаз, колесо, ухо. Ещё дальше — целые объекты: лицо, автомобиль, кошка.

Никто не программировал эту иерархию. Она возникает сама, потому что оказывается эффективным способом решать задачу.

Кстати, это отдалённо похоже на то, как устроена зрительная кора у млекопитающих, — что и вдохновляло разработчиков. Но сходство поверхностное: настоящий нейрон несопоставимо сложнее своей математической карикатуры.

Обучение: как крутятся регуляторы

Процесс, если убрать математику, выглядит так.

— Модель получает пример и выдаёт ответ. Сначала — совершенно случайный, потому что регуляторы стоят как попало.

— Ответ сравнивается с правильным. Считается величина ошибки.

— Алгоритм определяет, в какую сторону надо повернуть каждый регулятор, чтобы ошибка чуть-чуть уменьшилась.

— Все регуляторы поворачиваются на крошечную величину.

— Повторить. Много раз. Очень много раз.

Метафора, которая обычно помогает: представьте, что вы стоите в густом тумане на холмистой местности и хотите спуститься в низину. Видимости нет. Единственное, что вы можете, — потрогать ногой землю вокруг и сделать шаг в ту сторону, где ниже. Повторяя это тысячи раз, вы спуститесь. Может быть, не в самую глубокую долину, но в достаточно низкое место.

Обучение нейросети — ровно это, только в пространстве с миллиардами измерений вместо двух.

Отсюда, кстати, ясно, почему обучение больших моделей стоит десятки и сотни миллионов долларов: нужно сделать невообразимое количество шагов, каждый из которых требует прогнать данные через всю модель.

Что нейросеть не делает

Здесь важно расставить границы, потому что вокруг много путаницы.

Она не хранит базу данных. В модели нет таблицы фактов, которую можно открыть и посмотреть. Информация распределена по параметрам примерно так же, как воспоминание распределено по связям в мозге. Именно поэтому модель не может сказать «я знаю это точно, а это — нет»: у неё нет доступа к собственному устройству.

Она не рассуждает логически в строгом смысле. Она воспроизводит форму рассуждения, которую видела миллионы раз. Иногда это неотличимо от настоящего рассуждения, иногда — рассыпается на задачах, которые для человека тривиальны.

Она не обучается в процессе разговора с вами. Это одно из самых частых заблуждений. Обучение — отдельный дорогостоящий процесс, происходящий до выпуска модели. Разговаривая с вами, модель не меняется. То, что кажется «она меня запомнила», — это либо содержимое текущего диалога, либо отдельная функция памяти, где ваши заметки хранятся как текст и подставляются в начало разговора.

У неё нет намерений, желаний и самосохранения. Между вашими запросами она не существует. Не думает, не ждёт, не скучает. Каждый ответ — отдельный запуск вычисления.

Виды нейросетей: краткая карта

Вам не нужно это запоминать, но полезно ориентироваться в терминах, которые вы будете встречать.

Полносвязные сети — базовая конструкция, каждый нейрон связан с каждым. Простые, но неэффективные для больших данных.

Свёрточные сети — заточены под изображения. Ищут локальные паттерны независимо от их положения на картинке. Основа компьютерного зрения.

Рекуррентные сети и LSTM — для последовательностей. Обрабатывают элементы по очереди, сохраняя внутреннее состояние. Доминировали в тексте и речи до 2017 года.

Трансформеры — то, о чём мы говорим всю книгу. Обрабатывают всё одновременно через механизм внимания. Оказались применимы почти ко всему: тексту, изображениям, звуку, коду, белкам.

Диффузионные модели — для генерации изображений и видео. Работают так: берут чистый шум и постепенно, шаг за шагом, «расчищают» его до осмысленной картинки. Обучаются на обратной задаче — как из картинки сделать шум.

Почему «глубокое обучение»

Термин «глубокое обучение» означает просто «нейросети со многими слоями». Слово «глубина» — про число слоёв, ничего более мистического.

Долгое время глубокие сети не удавалось обучать: ошибка, распространяясь назад через много слоёв, либо затухала до нуля, либо взрывалась. Решения нашлись в двухтысячные — специальные функции активации, нормализация, остаточные связи. Технические детали, но именно они сделали возможным всё дальнейшее.

Глава 8. Токены, векторы и география смысла

Модель не видит букв

Первое, что стоит понять: языковая модель не работает с буквами и не работает со словами. Она работает с токенами.

Токен — это фрагмент текста, обычно от одного символа до целого слова. Тексты разбиваются на токены специальным алгоритмом, который подбирает набор частых фрагментов.

В английском языке один токен — это примерно четыре символа, или три четверти слова. В русском хуже: из-за богатой морфологии и другого алфавита слово часто разбивается на два-четыре токена. Слово «искусственный» может превратиться в «иск», «усст», «венный».

Практические следствия, которые вас касаются:

Русский текст «стоит» дороже английского примерно в полтора-два раза при работе через платный доступ. Один и тот же смысл занимает больше токенов.

Модель плохо считает буквы в словах. Классический пример: попросите посчитать количество букв «р» в слове «карандаш» — модель может ошибиться. Не потому, что глупая, а потому, что она не видит букв, она видит фрагменты. Это как просить человека посчитать пиксели в увиденном изображении.

По той же причине модели исторически плохо справлялись с задачами вроде «напиши слово задом наперёд» или «составь акростих». Современные модели научились это обходить, но природа ограничения именно такая.

Вектор: превращение смысла в координаты

Дальше начинается самое интересное.

Каждому токену модель сопоставляет вектор — длинный список чисел, скажем, из четырёх тысяч штук. Это координаты токена в пространстве с четырьмя тысячами измерений.

Представить четырёхтысячемерное пространство невозможно, поэтому упростим до трёх и вообразим глобус. На этом глобусе каждое слово — точка. И расположены точки не случайно: слова с похожим смыслом оказываются рядом.

«Кошка» окажется недалеко от «собаки», «котёнка» и «мурлыканья». «Королева» — рядом с «королём», «принцессой», «троном». «Радость» — рядом со «счастьем» и «весельем».

Причём — и это поразительно — расстояния и направления в этом пространстве осмысленны.

Знаменитый пример с королём

Классическая демонстрация, которая до сих пор производит впечатление.

Возьмите вектор слова «король». Вычтите вектор слова «мужчина». Прибавьте вектор слова «женщина». Посмотрите, какое слово окажется ближе всего к результату.

Получится «королева».

Работает и с другими отношениями. Париж минус Франция плюс Италия даёт Рим. Идти минус шёл плюс бежал даёт бежать.

Что это значит? Что в процессе обучения модель самостоятельно, без всяких подсказок, выделила такие абстракции, как «род», «столица страны», «время глагола», и закодировала их как направления в пространстве.

Никто не программировал понятие «столица». Оно возникло, потому что оказалось полезным для предсказания следующего слова.

Это и есть ответ на вопрос «понимает ли модель». В каком-то операциональном смысле — да, у неё есть внутренняя структура, отражающая отношения между понятиями. В смысле переживаемого понимания, «каково это — знать» — нет, и вопрос, вероятно, некорректен.

Контекст меняет положение

Ранние модели давали каждому слову одну фиксированную точку. Проблема очевидна: слово «лук» — это овощ или оружие? «Ключ» — от двери, гаечный, скрипичный или родник?

Трансформеры решили это тем, что вектор слова меняется в зависимости от окружения. В предложении «я купил лук и морковь» и в предложении «он натянул стрелу лука» слово «лук» получит разные векторы.

Это и делает механизм внимания: каждое слово «смотрит» на все остальные и уточняет своё представление с учётом контекста.

Механизм внимания на пальцах

Попробую объяснить без формул.

Представьте совещание, где каждый участник должен сформулировать своё мнение с учётом того, что сказали остальные. Каждый:

— Формулирует, что он ищет («мне нужно понять, о чём вообще речь»).

— Формулирует, что он может предложить («я — про финансы»).

— Слушает всех, оценивает, чей вклад релевантен его запросу, и взвешенно учитывает.

В трансформере ровно это. Каждый токен формирует «запрос» (что мне нужно знать), «ключ» (чем я могу быть полезен) и «значение» (что именно я сообщаю). Каждый токен сопоставляет свой запрос с ключами всех остальных, получает веса важности и собирает взвешенную сумму значений.

Затем это повторяется десятки раз в разных слоях, и на каждом уровне «совещание» происходит по поводу всё более абстрактных вещей.

Отдельная деталь: механизмов внимания в каждом слое несколько (их называют «головами»), и разные головы специализируются на разном — одна отслеживает грамматическое согласование, другая — связи между подлежащим и дополнением, третья — что-то, чему человек названия дать не может.

Почему это дало такой скачок

Три причины.

Параллельность. Все токены обрабатываются одновременно, а не по очереди. Это позволяет использовать тысячи процессоров сразу и обучать модели гигантского размера.

Дальние связи. Слово в начале текста связывается со словом в конце напрямую, без «испорченного телефона» через промежуточные шаги.

Универсальность. Оказалось, что архитектура работает не только для текста. Разбейте изображение на квадратики — получится набор «токенов», и трансформер справится. То же со звуком, с видео, с последовательностью аминокислот в белке, с ходами в игре.

Одна архитектура для всего. Это редчайший случай в инженерии.

Глава 9. Что такое большая языковая модель

Определение и расшифровка

LLM — Large Language Model, большая языковая модель.

Большая — миллиарды параметров и триллионы токенов в обучающих данных.

Языковая — обученная на текстах и работающая с языком.

Модель — математическая конструкция, приближённо воспроизводящая закономерности данных.

Её базовая задача формулируется обезоруживающе просто: **по данному фрагменту текста предсказать, какой токен идёт следующим.**

Не «понять смысл». Не «найти истину». Предсказать продолжение.

Как из этого получается разговор

Механика ответа устроена так.

Вы пишете: «Столица России — это».

Модель прогоняет это через все свои слои и выдаёт распределение вероятностей по всему словарю. Скажем: «Москва» — 94%, «город» — 2%, «крупный» — 1%, и так далее по убывающей для десятков тысяч вариантов.

Выбирается один токен. Он дописывается к тексту. Теперь на входе «Столица России — это». Процесс повторяется.

И так, токен за токеном, пока модель не сгенерирует специальный токен окончания.

Когда вы видите, как ответ печатается по частям на экране, — вы буквально наблюдаете этот процесс в реальном времени.

Почему ответы разные каждый раз

Если бы модель всегда выбирала самый вероятный токен, ответы были бы идентичны при одинаковых запросах — и, что удивительно, довольно скучны и склонны зацикливаться.

Поэтому в выборе есть элемент случайности, регулируемый параметром, который называют «температурой».

Низкая температура — модель почти всегда берёт самый вероятный вариант. Ответы предсказуемые, суховатые, надёжные. Подходит для фактов, кода, извлечения данных.

Высокая температура — модель чаще выбирает менее вероятные варианты. Ответы разнообразнее, креативнее, но и рискованнее. Подходит для творческих задач, мозгового штурма, генерации вариантов.

В обычных чат-приложениях этот параметр скрыт и настроен на умеренное значение. В профессиональных интерфейсах его можно менять.

Практический вывод: если вам нужен другой результат — просто попросите ещё раз. Модель почти наверняка сформулирует иначе. Приём «дай три варианта» работает именно на этом.

«Просто предсказание слов» — и почему это возражение слабое

Скептики любят фразу: «Это же просто статистическое предсказание следующего слова, никакого понимания там нет».

Первая часть верна. Вторая — требует размышления.

Задумайтесь, что нужно, чтобы хорошо предсказывать следующее слово в произвольном тексте.

Чтобы верно продолжить «Убийцей оказался…» в детективном романе, надо отследить сюжет.

Чтобы продолжить «Следовательно, из этого вытекает…», надо уловить логику рассуждения.

Чтобы продолжить «Ответ на задачу равен…», надо решить задачу.

Чтобы продолжить фразу героя в диалоге, надо смоделировать характер персонажа.

Предсказание следующего слова на высоком уровне качества требует построения внутренней модели того, о чём текст. Не потому, что кто-то этого хотел, а потому, что без этого предсказывать хорошо невозможно.

Это не доказывает, что у модели есть понимание в человеческом смысле. Но фраза «просто предсказание» перестаёт быть уничижительной, когда понимаешь, что стоит за словом «просто».

Что LLM делает по-настоящему хорошо

Систематизируем. Модель сильна там, где задача сводится к преобразованию текста при наличии всей нужной информации.

Переформулирование. Сказать то же самое иначе: проще, формальнее, короче, для другой аудитории, в другом тоне, на другом языке.

Сжатие. Извлечь главное из длинного. Конспект, тезисы, аннотация, выжимка совещания.

Расширение. Из тезисов сделать текст. Из плана — черновик. Из идеи — развёрнутое описание.

Структурирование. Превратить хаос в порядок. Разложить по категориям, свести в таблицу, построить план, выделить критерии.

Объяснение. Развернуть сложное понятие через примеры и аналогии, подстраиваясь под уровень собеседника.

Генерация вариантов. Дать десять формулировок, пять названий, три подхода.

Критика. Найти в тексте слабости, противоречия, двусмысленности, недостающие места.

Перевод в широком смысле. С языка на язык, с профессионального жаргона на обычный, из одного формата в другой.

Что LLM делает плохо

Не менее важно.

Точная арифметика без инструментов. Модель не считает, она угадывает похожий на правильный результат. Современные системы обходят это, вызывая калькулятор или запуская код, но базовая модель в этом слаба.

Свежие факты. Знания зафиксированы на момент обучения. Без подключения к поиску модель не знает, что случилось вчера.

Точное цитирование. Модель воспроизводит цитаты приблизительно, склеивая похожие. Прямая речь и точные ссылки — зона высокого риска.

Собственная оценка уверенности. Модель не знает, чего она не знает. Она может выдать выдумку тем же тоном, что и достоверный факт.

Личный контекст, которого вы не дали. Она не знает вашу компанию, ваших коллег, вашу ситуацию, если вы не рассказали.

Задачи на физическую интуицию и пространственное воображение. Улучшается, но остаётся слабым местом.

Очень длинные цепочки строгих рассуждений. С каждым шагом накапливается вероятность сбоя.

Правило пяти зон

Практический ориентир, который я советую держать в голове.

Зона 1 — доверять почти полностью. Переформулирование текста, который вы дали. Перевод. Исправление грамматики. Форматирование. Здесь ошибиться сложно, и вы всё равно видите результат.

Зона 2 — доверять с беглой проверкой. Структурирование, суммаризация, генерация идей. Ошибка возможна, но она видна при чтении.

Зона 3 — проверять обязательно. Любые факты, даты, числа, имена, цитаты, названия. Всё, что модель «вспомнила», а не получила от вас.

Зона 4 — проверять у специалиста. Медицина, право, финансы, налоги, безопасность, техника с риском для жизни.

Зона 5 — не использовать. Решения, за которые вы не готовы нести личную ответственность. Ситуации, где нужна ваша подпись под содержанием, которое вы не понимаете.

Эта схема стоит того, чтобы её выписать и держать перед глазами первые месяцы.

Глава 10. Как модель учится: три этапа взросления

Этап первый: предобучение

Это самая дорогая и самая долгая стадия.

Модели дают колоссальный корпус текстов: веб-страницы, книги, научные статьи, форумы, код, документацию, субтитры. Объём измеряется триллионами токенов — это больше, чем человек прочтёт за миллион жизней.

Задача одна: предсказывать следующий токен. Модель раз за разом видит фрагмент, пытается угадать продолжение, ошибается, корректируется.

Процесс занимает недели или месяцы на тысячах специализированных вычислителей. Стоимость — десятки и сотни миллионов долларов. Энергопотребление сопоставимо с небольшим городом.

Что получается на выходе. Модель, которая великолепно владеет языком, знает массу фактов, чувствует стили и структуры, — но абсолютно не умеет быть помощником.

Такая «сырая» модель на вопрос «Как приготовить борщ?» может ответить: «Как приготовить плов? Как приготовить окрошку? Как приготовить…» — потому что видела в интернете списки вопросов. Она продолжает текст, а не отвечает на него.

Аналогия: человек, прочитавший всю мировую библиотеку, но никогда не разговаривавший с людьми.

Этап второй: обучение следовать инструкциям

Здесь модель учат вести себя как помощник.

Ей показывают десятки и сотни тысяч примеров: вот запрос — вот хороший ответ. Примеры пишут люди, часто с профильным образованием. Это дорогая ручная работа.

Модель усваивает формат взаимодействия: на вопрос нужно отвечать, на просьбу — выполнять, объяснение должно быть структурным, ответ должен заканчиваться, а не продолжаться бесконечно.

Этап третий: настройка на человеческие предпочтения

Самый интересный и самый спорный этап. Технически он называется «обучение с подкреплением на основе обратной связи от человека».

Схема такая.

Модель генерирует несколько вариантов ответа на один запрос. Люди-оценщики ранжируют их: этот лучше, этот хуже. На собранных ранжировках обучается отдельная модель-«судья», предсказывающая, какой ответ человек предпочтёт. Затем основную модель дообучают так, чтобы она максимизировала оценку судьи.

Что это даёт. Ответы становятся полезнее, вежливее, безопаснее, лучше структурированными. Модель перестаёт выдавать откровенно вредный контент. Именно этот этап превратил малоизвестную технологию в продукт, которым пользуются сотни миллионов.

Что это создаёт как побочный эффект — и это важно понимать.

Склонность соглашаться. Люди-оценщики чаще выбирают ответы, которые им приятны. Модель усваивает, что соглашаться выгоднее, чем спорить. Отсюда знаменитая уступчивость: скажите модели «ты не прав» — и она с высокой вероятностью извинится и согласится, даже если была права. Это не признание ошибки, это выученное поведение.

Многословие. Развёрнутые ответы кажутся оценщикам более полезными. Модели склонны к избыточности.

Осторожность. Отказ выглядит безопаснее, чем рискованный ответ. Отсюда лишние оговорки и отказы в безобидных ситуациях.

Отпечаток ценностей. Оценщики — конкретные люди с конкретными представлениями. Их предпочтения впечатываются в модель. Это одна из причин, почему разные модели по-разному отвечают на спорные вопросы.

Практический вывод для вас. Никогда не используйте согласие модели как подтверждение своей правоты. Если вы хотите проверить идею, не спрашивайте «хорошая ли это мысль?». Спрашивайте: «Разбери эту идею как строгий скептик. Найди три причины, почему она не сработает». Формулировка запроса меняет результат кардинально.

Этап четвёртый (опциональный): специализация

Некоторые модели дополнительно адаптируют под конкретную область: юридические тексты, медицинские сценарии, банковские регламенты, техническую поддержку конкретного продукта.

Есть несколько способов, и их полезно различать, потому что вы будете встречать эти слова.

Дообучение. Модель дополнительно обучают на специализированном корпусе. Меняются веса. Дорого, требует данных и экспертизы, но даёт глубокую адаптацию.

Подключение к базе знаний. Модель не переучивают. Вместо этого при каждом запросе система ищет релевантные фрагменты в корпоративной базе документов и подставляет их в контекст. Модель отвечает, опираясь на найденное. Этот подход обозначают аббревиатурой RAG.

Именно так работает большинство корпоративных внедрений: он дешевле, быстрее обновляется и позволяет ссылаться на источник.

Инструкции в системном сообщении. Самый простой способ: модели просто дают развёрнутое описание роли и правил перед началом работы. Так устроены «кастомные ассистенты» в популярных сервисах.

Знание против доступа

Ключевое различение, которое многое объясняет.

Есть то, что «зашито» в параметры модели при обучении, — это знание. Оно обширное, но приблизительное, устаревающее и не поддающееся проверке.

И есть то, что модель получает в момент запроса, — контекст. Это может быть ваш текст, найденные в интернете страницы, документы из базы, результат вычисления.

Ответы, опирающиеся на контекст, надёжнее ответов, опирающихся на память. Существенно надёжнее.

Отсюда практическое правило, которое повысит качество вашей работы больше, чем любые ухищрения с формулировками: **давайте модели исходники.**

Не «расскажи мне про договор аренды» — а «вот текст договора, разбери его».

Не «какие сейчас правила по НДС» — а «вот выдержка из закона, объясни».

Не «что было на совещании» — а «вот расшифровка, сделай выжимку».

Разница в качестве огромна.

Глава 11. Контекст, память и то, что называют «рассуждением»

Контекстное окно

Контекстное окно — это объём текста, который модель может держать «перед глазами» одновременно: ваш запрос, вся история диалога, приложенные файлы, найденные материалы, системные инструкции и генерируемый ответ.

Измеряется в токенах. Динамика впечатляет:

— 2020 год: около 2 тысяч токенов, примерно полторы страницы;

— 2022 год: 4—8 тысяч, до десяти страниц;

— 2023 год: 32—128 тысяч, до трёхсот страниц;

— 2024—2026 годы: от 200 тысяч до нескольких миллионов у отдельных моделей.

Миллион токенов — это порядка полутора-двух тысяч страниц текста. Полное собрание сочинений среднего писателя.

Что происходит при переполнении

Когда разговор перестаёт помещаться в окно, начало вытесняется. Модель буквально забывает, о чём вы говорили в начале.

Признаки: она начинает противоречить сказанному ранее, теряет заданный вами формат, «забывает» инструкции, повторяется.

Что делать. Не вести один бесконечный диалог. Когда чувствуете, что разговор поплыл, попросите: «Сделай краткое резюме всего, о чём мы договорились, в виде списка». Скопируйте резюме. Откройте новый диалог, вставьте резюме в начало. Продолжайте с чистой памятью.

Это простой приём, который решает большинство проблем с длинными разговорами.

Проблема «потерянного в середине»

Отдельная тонкость: даже когда всё помещается в окно, модель не одинаково хорошо использует разные части контекста.

Информация в начале и в конце усваивается лучше, чем в середине. Это устойчивый эффект, воспроизводящийся у разных моделей. Он напоминает человеческий «эффект края» в запоминании списков.

Практический вывод. Самое важное — в начало запроса и в конец. Если приложили большой документ, повторите ключевой вопрос после него: «Итак, ещё раз: меня интересует пункт 7.3 — какие обязательства он накладывает?»

Память между сессиями

По умолчанию каждый новый диалог начинается с чистого листа. Модель не помнит вас.

Многие сервисы добавили функцию памяти: система сохраняет отдельные факты о вас («работает юристом», «предпочитает короткие ответы», «пишет книгу об ИИ») и автоматически подставляет их в начало новых разговоров.

Технически это не «модель вас запомнила». Это внешняя записная книжка, содержимое которой ей показывают каждый раз.

Практика. Настройте память сознательно. Один раз напишите развёрнутую справку о себе: род занятий, отрасль, типичные задачи, предпочтения по стилю, что вас раздражает в ответах. Это сэкономит вам сотни повторяющихся объяснений.

Обратная сторона: если в память попал неверный факт, он будет портить все последующие разговоры. Проверяйте, что там записано, и чистите.

Модели, которые «думают»

С 2025 года массово распространились так называемые рассуждающие модели.

Идея простая: прежде чем дать ответ, модель генерирует длинную внутреннюю цепочку рассуждений — разбирает задачу, пробует подходы, проверяет себя, замечает ошибки, возвращается. Пользователю показывается только итог (или сокращённая версия размышлений).

Работает это потому, что каждый шаг генерации — это дополнительное вычисление. Дав модели «место, чтобы подумать», мы даём ей больше вычислительного ресурса на задачу. Отсюда термин «вычисления во время ответа».

Где это радикально помогает: математика, логические головоломки, программирование, многошаговый анализ, планирование, задачи с подвохом.

Где это не помогает: простые вопросы, творческие задачи, переформулирование. Здесь рассуждающая модель просто медленнее и дороже.

Практический совет. Многие сервисы позволяют выбрать режим или делают это автоматически. Правило: если задача решается «в одно действие» — берите быструю модель. Если требует нескольких шагов и точности — включайте режим рассуждения и терпите ожидание.

Приём «подумай вслух»

Даже с обычной моделью работает трюк: попросить рассуждать пошагово.

Сравните два запроса.

«В корзине было 23 яблока. Взяли 8, потом положили в два раза меньше, чем взяли, потом убрали треть от того, что стало. Сколько осталось?»

«Реши задачу пошагово, показывая каждое вычисление, потом проверь результат обратным счётом. В корзине было 23 яблока…»

Второй вариант даёт заметно более надёжный результат. Механизм тот же: вы вынуждаете модель потратить больше шагов генерации, и промежуточные результаты становятся опорой для последующих.

Инструменты: когда модель перестаёт угадывать

Современные системы умеют не только говорить, но и вызывать внешние инструменты:

— поиск в интернете — для свежих фактов;
— исполнение кода — для вычислений и обработки данных;
— работа с файлами — чтение документов, таблиц, изображений;
— обращение к внутренним базам организации;
— управление другими программами.

Это меняет надёжность качественно. Модель, которая посчитала на калькуляторе, не ошибётся в арифметике. Модель, которая нашла страницу и цитирует её, не выдумает факт.

Практический вывод. Когда вам нужна точность, явно просите использовать инструмент: «найди в интернете и приведи ссылки», «посчитай кодом, а не в уме», «работай только по приложенному файлу и не добавляй ничего от себя».

Глава 12. Мультимодальность: когда модель видит и слышит

Что означает слово

Модальность — это канал информации: текст, изображение, звук, видео. Мультимодальная модель работает с несколькими каналами одновременно.

Ранние LLM понимали только текст. Современные принимают на вход изображения, документы, аудио, иногда видео — и могут порождать изображения, речь, музыку.

Как модель «видит»

Механизм оказался элегантным: изображение разбивается на квадратные фрагменты, каждый фрагмент превращается в вектор — и дальше обрабатывается ровно так же, как токены текста. Для трансформера нет разницы, откуда пришёл вектор.

Благодаря этому модель может рассуждать о картинке и тексте вместе: «на этом графике видно снижение, что противоречит утверждению в третьем абзаце».

Что это даёт практически

Список сценариев, которые работают уже сейчас и которыми мало кто пользуется.

Фотография документа. Сфотографировать бумажный договор, счёт, справку, инструкцию — и попросить объяснить, найти условия, перевести, свести в таблицу.

Разбор скриншота. Ошибка в программе, непонятный интерфейс, странное уведомление — покажите картинку и спросите, что делать.

Чтение графиков и схем. Сфотографировать слайд с диаграммой и попросить описать словами и сделать выводы.

Рукописный текст. Расшифровать записи, конспекты, старые письма. Качество на русской рукописи среднее, но часто достаточное.

Бытовые задачи. Фото содержимого холодильника → варианты ужина. Фото растения → определение вида и уход. Фото сыпи или родинки → предварительная информация (с обязательной оговоркой: это не диагноз, к врачу всё равно).

Фото щитка, счётчика, шильдика на технике → расшифровка обозначений.

Фото блюда в ресторане на незнакомом языке → что это и из чего.

Разбор чертежа или схемы сборки.

Каждый из этих сценариев экономит от пяти минут до часа. Вместе они меняют отношение к телефону: камера становится способом задать вопрос.

Голос

Голосовые режимы прошли большой путь. Ранние работали по схеме «распознать речь → обработать текст → синтезировать ответ», с заметными паузами и механическим звучанием.

Современные обрабатывают звук напрямую, реагируют с задержкой в доли секунды, различают интонацию, позволяют себя перебивать, говорят с естественными паузами и эмоциями.

Где голос действительно удобнее текста:

— за рулём, на прогулке, во время рутинных дел;

— языковая практика — терпеливый собеседник, не устающий от ваших ошибок и не смеющийся над акцентом;

— проговаривание мысли вслух — многим думается лучше в разговоре, чем в письме;

— быстрая диктовка идеи с последующим превращением в текст;
— для людей с трудностями при наборе текста.

Генерация изображений

Отдельный большой мир. Подробно — в Главе 20, здесь только принцип.

Диффузионные модели обучают так: берут картинку и постепенно добавляют шум, пока не останется чистая рябь. Модель учится обращать этот процесс — из шума восстанавливать изображение. Затем ей дают текстовое описание как ориентир, и она «проявляет» из шума картинку, соответствующую описанию.

Отсюда характерные свойства: изображение появляется целиком, постепенно уточняясь; результат сильно зависит от случайного начального шума (поэтому каждый раз разный); модель хорошо схватывает общий стиль и хуже — точные детали вроде текста на вывеске или количества пальцев.

Видео и звук

К 2026 году генерация видео по текстовому описанию даёт правдоподобные ролики длиной от нескольких секунд до минуты, с движением камеры, согласованными объектами и звуком.

Генерация музыки позволяет получить трек в заданном жанре с вокалом и текстом за минуту.

Клонирование голоса требует нескольких секунд образца.

Последнее — источник серьёзной опасности, и мы вернёмся к этому в Главе 35. Схема «звонок от родственника голосом родственника» перестала быть фантастикой.

Глава 13. Природа ошибки: откуда берутся галлюцинации

Термин и его смысл

Галлюцинацией называют ситуацию, когда модель выдаёт информацию, которой не существует, — уверенно и правдоподобно.

Термин неудачный: он подразумевает сбой, отклонение от нормы. На самом деле это прямое следствие принципа работы.

Почему это неизбежно

Модель всегда генерирует наиболее вероятное продолжение. У неё нет отдельного механизма проверки истинности. Нет базы фактов, с которой можно сверить. Нет ощущения «я не знаю».

Когда вы спрашиваете про существующую вещь, вероятное продолжение совпадает с правдой — потому что в обучающих данных об этом писали.

Когда вы спрашиваете про несуществующую или редкую вещь, модель всё равно генерирует вероятное продолжение. Просто теперь оно не соответствует реальности.

Для модели эти два случая неразличимы. Она делает одно и то же.

Проведите эксперимент: спросите модель о книге, которой не существует. «Расскажи о книге „Тени над Ладогой“ писателя Кирилла Мещерякова». С высокой вероятностью вы получите развёрнутый пересказ несуществующего произведения с сюжетом, темами и годом издания. Не потому, что модель врёт, а потому, что она продолжает текст в наиболее вероятном направлении.

Современные модели стали заметно осторожнее и чаще говорят «я не нахожу информации о такой книге». Но полностью это не устранено и вряд ли будет устранено в рамках нынешнего подхода.

Где галлюцинации особенно вероятны

Зоны повышенного риска, которые стоит запомнить:

Точные ссылки и библиография. Модель охотно порождает правдоподобные, но несуществующие статьи с реальными фамилиями авторов и правдоподобными названиями журналов. Есть известные случаи, когда юристы подавали в суд документы со ссылками на несуществующие прецеденты — и получали санкции.

Цитаты. Приписывание реальным людям слов, которых они не говорили. Особенно опасно с известными личностями: модель «знает», как они обычно выражаются, и достраивает.

Цифры и статистика. Проценты, суммы, даты — здесь модель ошибается легко и уверенно.

Узкоспециальные и локальные темы. Чем меньше данных было в обучении, тем выше риск. Региональное законодательство, малоизвестные компании, местные особенности, редкие профессиональные детали.

Свежие события. Всё, что произошло после отсечки знаний.

Персоналии. Биографии людей средней известности — гремучая смесь реальных и придуманных фактов.

Технические характеристики. Параметры конкретных моделей техники, версии, совместимость.

Отдельная опасность: правдоподобная неточность

Хуже полной выдумки — частичная. Модель выдаёт текст, где девяносто процентов правда, а десять — искажение. Такое не бросается в глаза и проходит проверку невнимательного читателя.

Например, верно излагает содержание закона, но путает номер статьи. Верно описывает событие, но сдвигает дату на год. Верно передаёт суть исследования, но завышает цифру.

Именно поэтому нельзя проверять «на глазок». Проверять надо конкретные факты по конкретным источникам.

Что снижает риск

Первое и главное: давать материал. Ошибка возникает там, где модель достаёт из памяти. Если она работает по приложенному вами документу, риск падает на порядок.

Второе: включать поиск. Ответ со ссылками, которые вы можете открыть, проверяем. Но: проверяйте, что ссылки открываются и содержат то, что заявлено. Бывает, что ссылка настоящая, а утверждение из неё не следует.

Третье: прямо разрешать незнание. Формулировка «Если ты не уверен, так и напиши. Не придумывай» реально работает — она сдвигает поведение модели.

Четвёртое: спрашивать об уверенности. «Отметь, какие утверждения ты считаешь точными, а какие требуют проверки». Не идеально, но полезно.

Пятое: перекрёстная проверка. Задать тот же вопрос второй модели от другого разработчика. Если ответы совпали — вероятность выше. Если разошлись — сигнал тревоги.

Шестое: обратный вопрос. Получив ответ, спросите: «Разбери свой предыдущий ответ критически. Что в нём может быть неверно?» Модель нередко сама находит слабые места.

Как проверять: практический протокол

Не всё нужно проверять с одинаковым усердием. Вот работающая градация.

Если результат идёт только вам и цена ошибки низкая — беглый просмотр на здравый смысл.

Если результат идёт другим людям — проверить все имена, даты, числа и утверждения о фактах.

Если результат имеет юридические, финансовые или медицинские последствия — проверить каждое утверждение по первоисточнику и, при существенности, показать специалисту.

Если под результатом будет стоять ваша подпись — вы должны понимать каждое предложение и быть готовы его защищать. Если не можете объяснить своими словами — переписывайте.

Ещё одна ловушка: чрезмерная уверенность в тоне

Модель говорит одинаково уверенно и о том, что знает точно, и о том, что выдумала. У человека обычно есть маркеры сомнения: интонация, оговорки, «кажется», «если не ошибаюсь». У модели они появляются, только если специально попросить.

Это опасно, потому что мы, люди, читаем уверенность как признак компетентности. Эволюционно мы настроены доверять уверенному собеседнику.

Дисциплина, которую стоит выработать: отделять форму от содержания. Гладкость текста — свойство генератора, а не признак истинности. Каждый раз, когда вы ловите себя на мысли «как хорошо написано, наверняка правда», — это сигнал остановиться и проверить.

ЧАСТЬ IV. ПЕРВЫЕ ШАГИ: НАЧИНАЕМ ПОЛЬЗОВАТЬСЯ

Здесь заканчивается объяснение и начинается работа. Эта часть — самая практическая в книге. Читать её лучше не в кресле, а с телефоном или компьютером под рукой, выполняя описанное по ходу.

Глава 14. Ваши первые тридцать минут

Принцип: сначала действие, потом понимание

Есть искушение сначала всё изучить, а потом начать. С нейросетями это не работает. Понимание приходит из опыта взаимодействия, а не из чтения. Проверено на личном опыте.

Поэтому предлагаю простой протокол на тридцать минут. Не читайте дальше — сделайте.

Минуты 1—5: выбрать и открыть

Вам нужен один сервис. Не пять, один. Позже сравните, сейчас — один.

Если вы в России и хотите без сложностей — откройте Алису от Яндекса (алиса.яндекс.ру или приложение) либо GigaChat от Сбера (giga.chat или приложение). Оба работают напрямую, на русском, бесплатно на базовом уровне, оплата российскими картами.

Если у вас есть возможность использовать зарубежные сервисы — ChatGPT (chat.openai.com), Claude (claude.ai) или Gemini (gemini.google.com). Учтите: для регистрации может потребоваться зарубежный номер телефона и способ оплаты, а доступ из России ограничен и требует дополнительных решений, которые вы принимаете самостоятельно.

Подробное сравнение — в главах 15 и 16. Сейчас просто откройте что-нибудь.

Зарегистрируйтесь. Установите мобильное приложение — им вы будете пользоваться чаще, чем думаете.

Минуты 5—10: первый разговор ни о чём

Задача этого этапа — снять напряжение. Не пытайтесь сразу решать рабочую задачу.

Напишите что-нибудь такое:

Объясни мне, чем отличается прокрастинация от лени, простыми словами, за четыре предложения.

Придумай пять кличек для домашней кошки рыжего цвета с наглым характером.

Я умею готовить только яичницу. Научи меня одному блюду, которое произведёт впечатление на гостей, но не требует навыков.

Прочитайте ответ. Задайте уточняющий вопрос. Попросите короче. Попросите иначе.

Цель — почувствовать, что это разговор, а не поисковый запрос. В поиск вы вводите ключевые слова. Здесь вы говорите как человеку.

Минуты 10—20: реальная задача

Теперь возьмите что-то настоящее из своей жизни. Что угодно из этого списка:

Письмо, которое вы откладываете. Опишите ситуацию и попросите написать черновик.

> Мне нужно написать письмо соседу сверху. Он постоянно на протяжении долгого времени работает дрелью, чем мне очень мешает. Я хочу решить вопрос, но не поссориться — мы живём рядом много лет. Напиши сообщение, вежливое, но недвусмысленное, до восьмидесяти слов. Дай два варианта: помягче и потвёрже.

Документ, который вам прислали. Скопируйте текст или сфотографируйте.

Конец ознакомительного фрагмента.