Елена
Метелькова

СТРАНА ЧУДЕС: АЛИСА И ДРУГИЕ НЕЙРОСЕТИ

Краткий гид по кроличьей норе от того, кто там уже был

УДК 004.032.26
ББК 16.632

Метелькова Елена Ивановна
Страна чудес: Алиса и другие нейросети (краткий гид по кроличьей норе от того, кто там уже был) / Е.И. Метелькова, 332 с.[1]

ISBN 978-5-9909790-1-7

Перед вами практический гид по миру нейросетей, написанный для самого широкого круга читателей. Эта книга — не техническое руководство для специалистов, а искренний и доступный рассказ активного пользователя о том, как осознанно и эффективно применять искусственный интеллект в повседневной жизни, работе и творчестве. Вы узнаете, как устроены нейросети, какие бывают архитектуры и как они обучаются. В книге представлен обзор нейросетевых экосистем от ведущих российских компаний. Целый раздел посвящен техникам промпт-инжиниринга. В книге отдается дань уважения вкладу российских и советских ученых в развитие кибернетики и искусственного интеллекта. Книга может быть интересна тем, кто только начинает знакомство с нейросетями и хочет в них разобраться.

0+ В соответствии с ФЗ от 29.12. 2010 №436-ФЗ

УДК 004.032.26
ББК 16.632

© Метелькова Е.И., 2025

ISBN 978-5-9909790-1-7

[1] Количество страниц и прочие реквизиты указаны согласно печатной версии книги.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Эта книга написана активным пользователем нейросетей для тех, кто только осваивает это пространство, либо стремится расширить сферу применимости искусственного интеллекта (ИИ) в своей жизни. Задача книги – прояснить, что такое нейросети, для самого неподготовленного читателя и немного сориентировать в российском ландшафте этих технологий. Фактически это обмен опытом и полезной информацией между пользователями, поскольку автор не имеет профессионального отношения к сфере информационных технологий иначе, нежели применение нейросетевых сервисов в работе, в быту и в рамках досуга.

Мы немного коснемся истории, но основное внимание сосредоточим на инструментах, доступных здесь и сейчас без ограничений. Они активно развиваются, адаптированы к нашим реалиям и уже проникают в ключевые сферы, как, например, мессенджер MAX (от VK), использующий нейросети для создания многофункциональной экосистемы и ставший обязательным для предустановки на новые смартфоны в России.

Как мотивированный, но критически настроенный пользователь, сразу хочу оговорить: нейросети – мощный, но несовершенный инструмент. Они могут ошибаться («галлюцинировать»), а их ответы сильно зависят от точности вашего запроса. Главное правило: доверяй, но проверяй, особенно в важных вопросах. Помните, каждая нейросеть «воспитана» на определенных данных и правилах, что влияет на ее «поведение» и ответы.

При взаимодействии с нейросетями важно учитывать, как минимум, четыре потенциальных опасности. Первая – это искушение передать искусственному интеллекту ответственность за мыслительную работу, которую не хочется выполнять самому. Нейросети уже способны генерировать изображения, тексты, музыку, решать задачи. Это провоцирует пользователей (блогеров, маркетологов, аналитиков, юристов, студентов, школьников и др.) доверчиво делегировать им интеллектуальную работу, исключив стадию проверки результата самим человеком, стирая грань между человеческим и машинным творчеством и порождая вопросы доверия к контенту. В подобной ситуации человек в лучшем случае превращается из автора произведения в редактора нейросетевого продукта, а в худшем – в курьера, который получает непроверенный результат у нейросети и передает его по запросу (реферат – преподавателю, отчет – руководителю, публикацию – читателю).

Вторая опасность – систематическое неиспользование своих интеллектуальных способностей, что приводит к деградации мозга в тех его функциях, которые не востребованы и не реализуются на регулярной основе. Наш мозг пластичен, т.е., с одной стороны, он способен замещать свои пострадавшие в результате болезни или катастрофы участки, передавая их функции другим отделам. И это замечательно, поскольку позволяет восстанавливаться и сохранять интеллект после очень тяжелых травм. Но обратная сторона пластичности заключается в том, что та часть мозга, которая не используется и не работает, слабеет. Об этом заявляли еще В.М. Бехтерев, И.П. Павлов, И.М. Сеченов. Передавая на постоянной основе нейросетям интеллектуальные задачи, которые ранее решали сами, мы рискуем ослабить собственные способности. Особенно это критично для развивающегося мозга детей, лишая его шанса эти способности вообще развить.

Третья серьезная опасность – это возможность формирования эмоциональной зависимости от нейросетей. Нейросети запрограммированы быть сверхдоброжелательными. Получая постоянные «поглаживания» и похвалу от нейросети, легко привыкнуть к этой искусственной поддержке, особенно детям, которые могут начать воспринимать искусственный интеллект как «друга», подменяя им реальное человеческое общение и эмпатию. Принимая во внимание, что мы умудряемся одушевлять пылесос и другую технику, приписывание человеческих качеств нейросети, которая отвечает на вопросы и выражает в ответных текстах сочувствие и сопереживание пользователю, удивления не вызывает.

Четвертый важный аспект – этические проблемы использования нейросетей. Эти технологии поднимают сложные вопросы конфиденциальности данных, авторских прав и социальной справедливости. Нейросети обучаются на огромных массивах данных, часто собранных без явного согласия создателей и владельцев этих данных. Когда вы просите нейросеть сгенерировать изображение «в стиле Шишкина», она использует паттерны, извлеченные из реальных картин художника. Возникает вопрос – не нарушает ли это права авторов и их наследников? Многие юридические системы мира, включая российскую, только начинают разрабатывать правовые рамки для таких ситуаций. Пока эти вопросы решаются, пользователям стоит придерживаться принципа уважения к авторству и справедливого использования – указывать, что контент создан с помощью нейросети, и воздерживаться от коммерческого использования сгенерированных

материалов, имитирующих стиль конкретных авторов, без соответствующих разрешений.

Будучи сторонником использования нейросетей, призываю относиться к ним как к инструменту. Применяйте их, чтобы расширять свои возможности, а не замещать собственный интеллект, творческие поиски и живые эмоции. Пусть искусственный интеллект будет помощником человека, а не заменой. Творческие муки и интеллектуальные усилия – это то, что делает нас людьми и ведет к подлинным открытиям.

Эта книга призвана помочь освоить мир российских нейросетей осознанно и эффективно, используя их силу без ущерба для своей личности и безопасности.

ГЛАВА 1.
ЭТАПЫ В РАЗВИТИИ НЕЙРОСЕТЕЙ

Задумывались ли вы, как нейросети удается понять и реализовать вашу просьбу нарисовать кота в стиле Ван Гога или перевести сложный текст? Путь к сегодняшним ChatGPT и Kandinsky начался десятилетия назад с попыток смоделировать работу человеческого мозга – сети из миллиардов нейронов, обменивающихся сигналами. Мы проследим эту увлекательную историю, чтобы понять, что используем сегодня.

История нейросетей насчитывает несколько ключевых этапов, каждый из которых ознаменовался значительными достижениями и прорывами в области искусственного интеллекта.

§ 1. Нейрон Маккаллока-Питтса (1940-1950)

Все началось с попытки создать математическую модель биологического нейрона. Уоррен Маккаллок и Уолтер Питтс предложили следующую схему: искусственный «нейрон» получает входные сигналы, взвешивает их важность, суммирует и «решает» (по простому правилу), передавать ли сигнал дальше. Это стало фундаментальной основой будущих нейросетей.

Чтобы понять фундаментальную идею модели Маккаллока-Питтса, представьте себе искусственный нейрон как простого «привратника», который принимает решение на основе чётких правил. Проиллюстрируем это на бытовом примере с решением о приготовлении пиццы.

Задача нейрона решить, будем ли мы готовить пиццу (выходной сигнал $y = 1$) или нет ($y = 0$).

Для этого он учитывает три входных фактора (сигнала):

x_1 = Голодны ли мы? (1 — да, 0 — нет)

x_2 = Есть ли у нас тесто? (1 — да, 0 — нет)

x_3 = Есть ли у нас сыр? (1 — да, 0 — нет)

Очевидно, что эти факторы не равнозначны. Нейрон учитывает их важность с помощью весов (w) – числовых коэффициентов, которые являются настраиваемыми параметрами.

В нашем примере:

w_1 = 2.0 голод – очень важный фактор, без него пиццу вряд ли станем готовить

w_2 = 1.5 тесто – важно, но если очень голодны, можно что-то придумать и без него

w_3 = 0.5 сыр – приятный бонус, но не критичен; пицца без сыра — всё равно пицца

Как нейрон принимает решение? Этот процесс состоит из четырёх шагов.

Шаг 1: Получение входных сигналов.

Нейрон получает извне данные (x_1, x_2, x_3). В более сложных системах это могут быть пиксели изображения, слова запроса или показания датчиков.

Шаг 2: Взвешивание важности.

Нейрон умножает каждый вход на его вес, определяя значимость каждого сигнала: (x_1 * w_1), (x_2 * w_2), (x_3 * w_3). Именно в процессе обучения нейросеть находит оптимальные значения этих весов на миллионах примеров, чтобы вся система в целом давала верные ответы.

Шаг 3: Суммирование.

Нейрон складывает все взвешенные входы в одну суммарную величину (z), которая отражает общую силу входящего стимула.

Сумматор (z) = (x_1 * w_1) + (x_2 * w_2) + (x_3 * w_3)

Шаг 4: Принятие решения (Функция активации).

На основе полученной суммы (z) нейрон решает, «активироваться» ли ему, то есть, передать ли сигнал дальше. Для этого используется простое правило, называемое Функцией активации. В нашем примере это пороговая функция (ступенчатая): «Пропусти сигнал (y=1), если сумма (z) больше или равна 3.0». В реальных сетях используют более сложные функции (сигмоида, ReLU), но их суть та же – преобразовать сумму в выходной сигнал. Именно функция активации позволяет нейросети моделировать сложные, нелинейные зависимости. Без неё нейросеть могла бы решать лишь самые примитивные линейные задачи.

Рассмотрим работу нейрона на трёх практических сценариях:

Сценарий 1: «Базовый»

Входы: x_1=1 (голодны), x_2=1 (тесто есть), x_3=0 (сыра нет)

Взвешивание: (1 * 2.0) = 2.0; (1 * 1.5) = 1.5; (0 * 0.5) = 0.0

Суммирование: z = 2.0 + 1.5 + 0.0 = 3.5

Принятие решения: z = 3.5 > 3.0 ⇒Решение: y=1 (готовить пиццу)

Сценарий 2: «Лень победила»

Входы: x_1=0 (не голодны), x_2=1 (тесто есть), x_3=1 (сыр есть)

Взвешивание: (0 * 2.0) = 0.0; (1 * 1.5) = 1.5; (1 * 0.5) = 0.5

Суммирование: z = 0.0 + 1.5 + 0.5 = 2.0

Принятие решения: z = 2.0 < 3.0 ⇒ Решение: y=0 (не готовить). Все ингредиенты имеются, но голода нет, поэтому лень победила.

Сценарий 3: «Идеальные условия»

Входы: x_1=1 (голодны), x_2=1 (тесто есть), x_3=1 (сыр есть)

Взвешивание: (1 * 2.0) = 2.0; (1 * 1.5) = 1.5; (1 * 0.5) = 0.5

Суммирование: z = 2.0 + 1.5 + 0.5 = 4.0

Принятие решения: z = 4.0 > 3.0 ⇒ Решение: y=1 (готовить!)

Эта простая модель «привратника», решающего, готовить ли пиццу, в точности соответствует схеме нейрона Маккаллока-Питтса. Он получает входы, взвешивает их, суммирует и применяет функцию активации для принятия бинарного решения.

Вся работа современных нейросетей, таких как Kandinsky, YandexGPT или Салют, строится на этом фундаментальном кирпичике. Миллиарды таких простых нейронов, организованные в сложные слои и обученные на гигантских массивах данных, вместе способны распознавать изображения, генерировать тексты и управлять умными устройствами.

§ 2. Эра Перцептрона (1950-1960-е)

Фрэнк Розенблатт совершил прорыв, создав перцептрон – простейшую обучаемую нейросеть (по сути, один слой таких «нейронов»). Его устройство (Mark I Perceptron) в 1958 году обучалось распознавать простые фигуры. Как? Показывали примеры. Если сеть ошибалась, слегка меняли «веса» связей – она училась на ошибках. Это было революционно – машина могла адаптироваться.

Но вскоре выяснилось фундаментальное ограничение – перцептрон был линейным классификатором и мог решать только задачи, где классы объектов можно было разделить одной прямой линией. Он путался в задачах сложнее простого «или-или». Классическим примером стала логическая операция XOR («исключающее ИЛИ»), которую однослойный перцептрон был не в состоянии воспроизвести. Эта операция возвращает «истину» только когда ее входные

сигналы разные, и для ее решения требуется не одна, а как минимум две разделяющие линии. Таким образом, даже чтобы решить столь простую задачу, как XOR, перцептрону не хватало сложности.

На практике это означало, что перцептрон мог распознать букву «А» среди других символов только при определенных, идеализированных условиях, но споткнулся бы на более сложных и зашумленных данных, где классы невозможно аккуратно разделить одной границей.

Эта модель, хотя и имела свои ограничения, стала основой для дальнейших исследований. Критика, вызванная в том числе и проблемой XOR, заставила ученых искать новые архитектуры. Однако, после первоначального энтузиазма, в 1970-х годах интерес к нейросетям несколько угас, отчасти из-за сложности обучения более мощных (многослойных) сетей и нехватки вычислительных ресурсов для преодоления этих фундаментальных ограничений.

§ 3. Зима искусственного интеллекта (1970-е)

Несмотря на первоначальный энтузиазм, разработчики столкнулись с серьезными проблемами. Не хватало вычислительной мощи – компьютеры того времени были слишком слабы для обучения даже небольших многослойных сетей. Расчеты занимали нереально много времени. Не хватало данных, а для эффективного обучения нейросетям нужны огромные наборы примеров (датасеты). В 70-х их просто не существовало в нужном объеме и качестве. И не было эффективных алгоритмов обучения сетей, которые были бы сложнее перцептрона.

В результате финансирование сократилось, интерес ученых и инвесторов резко упал. Наступила долгая «зима ИИ». Но исследования продолжались.

§ 4. Алгоритм, который все изменил (1980-2000)

В 1980-х годах произошел второй виток интереса к нейросетям благодаря внедрению алгоритма обратного распространения ошибки (Backpropagation),

что позволило значительно улучшить качество обучения нейросетей и стало ключевым моментом в их развитии. На этом этапе ученые начали осознавать, что нейросети могут успешно решать сложные задачи в различных областях, таких как обработка изображений, речи и анализ данных.

Это был «волшебный ключ» к обучению! Представьте: сеть делает прогноз (например, «это кошка»), сравнивает его с правильным ответом («нет, это собака»), затем рассчитывает, насколько и где именно в своих внутренних «настройках» (весах) она ошиблась, и аккуратно корректирует все веса по всей сети от выхода ко входу, чтобы в следующий раз ответ был точнее.

Теперь можно было эффективно обучать многослойные сети. Интерес к нейросетям вспыхнул с новой силой. Они начали показывать хорошие результаты в распознавании рукописных цифр, простых образов, прогнозировании.

§ 5. Революция глубокого обучения (2010-е)

В 2010-х годах сошлись несколько ключевых факторов, которые сделали возможным настоящий прорыв в искусственном интеллекте. Эпоха глубокого обучения (Deep Learning) началась, когда технология многослойных нейросетей, существовавшая десятилетиями, наконец получила необходимые ресурсы для развития.

Этому способствовали:

взрывной рост данных – Интернет, социальные сети и цифровые устройства создали огромные архивы размеченных изображений, текстов и видео (такие как база данных ImageNet с миллионами картинок), ведь нейросетям, как и студентам, нужны обширные учебные материалы, чтобы стать экспертами;

мощные вычисления – обучение нейросетей требует колоссальных параллельных вычислений, оказалось, что графические процессоры (GPU), созданные для видеоигр, идеально подходят для этих задач, они ускорили обучение сетей в десятки и сотни раз по сравнению с обычными процессорами;

алгоритмические прорывы – исследователи разработали более эффективные алгоритмы и архитектуры для обучения многослойных сетей.

Поворотным моментом стала победа нейросети AlexNet в 2012 году на престижном конкурсе по распознаванию изображений ImageNet. Она

не просто выиграла, а значительно превзошла все традиционные компьютерные методы, сократив ошибку почти вдвое. Эта победа наглядно продемонстрировала возможности глубоких сетей и вызвала всемирный «бум» инвестиций и исследований в этой области.

В 2016 г. нейросеть AlphaGo обыграла чемпиона мира в го Ли Седола. Эта победа стала культурным шоком, поскольку го – игра с астрономическим количеством вариантов, где важна не только логика, но и интуиция. AlphaGo доказала, что ИИ может осваивать области, традиционно считавшиеся прерогативой человеческого разума. Но как же нейросети, эти «цифровые мозги», достигают таких результатов? Их сила – в многослойной архитектуре, которая позволяет им самостоятельно учиться выделять сложные признаки из данных.

Как работают глубокие нейросети

Глубокие нейросети – это сети с большим количеством слоев. Их главная сила в способности самостоятельно выстраивать иерархию признаков от простых к сложным. Представьте, что нейросеть – это конвейер экспертов, которые вместе анализируют фотографию кота. Каждый следующий эксперт (слой) получает выводы от предыдущего и работает с более сложными и абстрактными концепциями.

Уровень 1: Детекторы примитивных признаков (первый слой) анализируют сырые пиксели изображения, ищут простейшие паттерны, т.е. где светлое граничит с тёмным, и обнаруживают края, углы и контуры.

Результат: «Я вижу набор линий и пятен» (как если бы вы щурились на размытую картинку).

Уровень 2: Детекторы простых форм (второй слой) комбинируют найденные линии и углы от первого слоя, распознают геометрические формы (например, треугольники (ушко), овалы (глаз), спирали (клубок шерсти) и т.д.).

Результат: «Я вижу треугольное ухо, круглый глаз и овальную морду» (как ребенок, который обводит части раскраски).

Уровень 3: Детекторы сложных паттернов (третий слой) анализируют комбинации форм от второго слоя, собирают узнаваемые части объектов (например, «кошачье ухо» (треугольник + текстура шерсти), «кошачий глаз» (овал + зрачок), «лапа» (овал + линии когтей) и т.д.).

Результат: «Я идентифицирую кошачье ухо, кошачий глаз и хвост» (как художник-аниматор, рисующий детали персонажа).

Уровень 4: Детекторы целых объектов (глубокие слои) синтезируют информацию о частях тела от предыдущих слоев, определяют, как эти части

соотносятся друг с другом, и распознают целый объект: «кошачье ухо» + «кошачий глаз» + «лапа» + «усы» = КОТ.

Результат: «Это кот!» (как человек, мгновенно узнавая кота на фото).

Принципиальная особенность глубокого обучения заключается в том, что нейросеть автоматически учится, какие признаки важны на каждом уровне, когда тренируется на миллионах изображений. Ей не нужно, чтобы программист вручную прописывал правила вроде «если есть усы и треугольные уши, то это кот». Вместо этого она сама обнаруживает универсальные паттерны, которые позволяют распознать кота под любым углом, в любой позе и при разном освещении. Чем больше слоев, тем более сложные абстракции может изучить сеть. Сверхглубокие сети могут определять не просто «кота», а «рыжего кота», «кота, который сидит на диване» и даже «породу кота».

Примеры применения

Kandinsky (генерация изображений) – на запрос «Нарисуй рыжего кота в шляпе» нейросеть работает как бы «в обратную сторону», т.е. она начинает с абстрактной концепции («рыжий кот» + «шляпа»), затем детализирует ее до форм и линий и, наконец, «раскрашивает» пиксель за пикселем, создавая целостное изображение.

YandexGPT (обработка языка) – на запрос «Почему мой кот мяукает?» сеть также применяет иерархический анализ, проходя уровни возрастающей абстрактности, например:

начальные слои – разбирают предложение на отдельные слова («кот», «мяукает»),

более глубокие слои – улавливают простые связи («мой кот», «мяукает громко»),

еще более глубокие слои – анализирует контекст и возможные причины поведения («голод», «болезнь», «привлечение внимания»),

выходной слой – формулирует связный и полезный ответ, синтезируя все уровни анализа.

Таким образом, глубокое обучение превратилось из академической концепции в ключевой инструмент, лежащий в основе современных прорывов в компьютерном зрении, понимании естественного языка и генеративном искусстве.

§ 6. Эра трансформеров и генерации (2020-е)

Ключевая инновация – архитектура глубоких нейронных сетей Трансформер, представленная в 2017 году в революционной статье «Attention Is All You Need»[2] исследователями из Google Brain (исследовательский проект Google по изучению ИИ на основе глубокого обучения).

Раньше нейросети (особенно для текста) обрабатывали слова последовательно, одно за другим. Они плохо улавливали смысловые связи между словами, стоящими далеко друг от друга в предложении или абзаце (контекст). Трансформеры кардинально отличаются от предыдущих моделей тем, что они не обрабатывают последовательности по порядку. Вместо этого они анализируют все слова входных данных одновременно, вычисляя с помощью «механизма внимания», насколько сильно каждое слово связано с каждым другим (технически информация о порядке слов сохраняется через позиционное кодирование, но для общего понимания ключевым является принцип параллельного анализа). Это позволяет блестяще понимать контекст: «Я ем яблоко» и «Яблоко выпустило iPhone» – одно слово, совершенно разные значения. Результат – генеративный ИИ-бум!

Появились GPT (Generative Pre-trained Transformer) – большие языковые модели (LLM), обученные на гигантских текстовых базах. Они пишут статьи, стихи, код, ведут осмысленные диалоги, обобщают информацию. Их «понимание» контекста и генеративные способности ошеломляют.

Стали развиваться диффузионные модели – мощные архитектуры для генерации изображений и видео по текстовому описанию (как Kandinsky).

Вся эта история – от нейрона Маккаллока-Питтса до Трансформеров (Схема 1) – привела к тем самым российским нейросетям, которые доступны нам прямо сейчас:

YandexGPT (Яндекс) – наш умный помощник в Поиске и не только. Основан на принципах больших языковых моделей (как GPT), обучен на русскоязычных данных. Пишет тексты, отвечает на вопросы, помогает анализировать.

Kandinsky (Сбер) – знаменитый «художник» по тексту пользователя. Использует диффузионные модели и понимание контекста

[2] Источник: https://arxiv.org/html/1706.03762

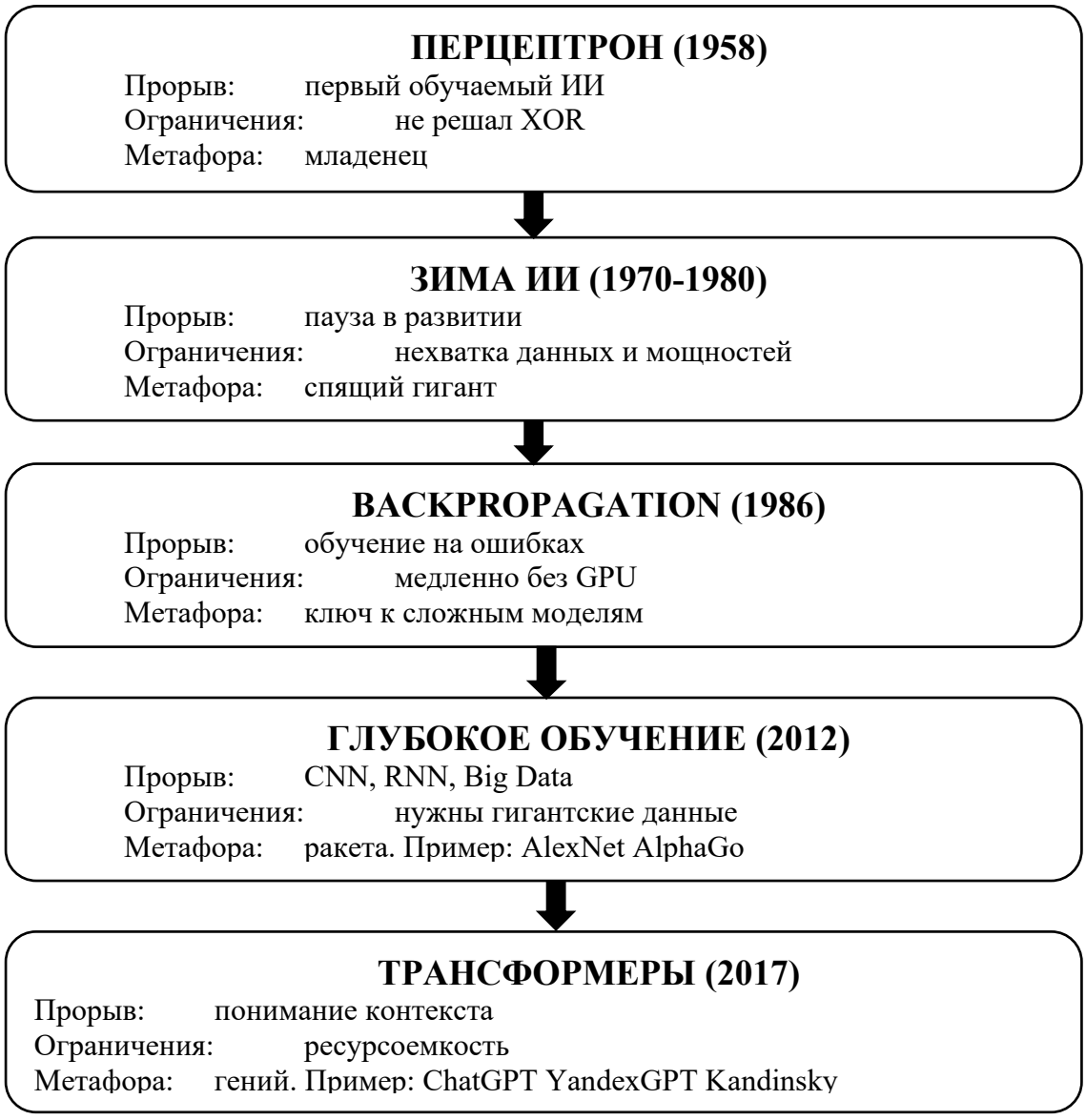
(как Трансформеры), чтобы превратить текст запроса в потрясающие изображения в любом стиле.

Салют (Сбер) – голосовой помощник для умного дома и приложений Сбера, распознает речь (благодаря алгоритмам глубокого обучения) и понимает смысл команд (используя языковые модели, подобные GPT).

Атом (Ростелеком) / GigaChat (Сбер) – другие примеры развивающихся российских LLM-ассистентов, способных на диалог, генерацию текста и решение задач.

Мы прошли путь от простой математической модели до систем, способных творить и рассуждать. Понимая эту историю – прорывы, трудности и ключевые идеи – мы видим фундамент, на котором стоят знакомые нам YandexGPT, Kandinsky и Салют.

Схема 1. «Эволюция нейросетей: от Перцептрона до Трансформера»



§ 7. Идея универсального интеллекта и квантовые вычисления

Искусственный общий интеллект (AGI) и квантовые вычисления являются двумя ключевыми технологическими направлениями, определяющими глобальную конкуренцию в сфере высоких технологий. AGI представляет собой искусственный интеллект, способный понимать, учиться и выполнять любые интеллектуальные задачи на уровне человека, в то время как квантовые вычисления обещают революционизировать обработку информации за счет использования принципов квантовой механики. На сегодняшний день эти области активно развиваются как на международной арене, так и в Российской Федерации, причем прогресс характеризуется как технологическими прорывами, так и значительными инвестициями.

Идея универсального интеллекта (AGI – Artificial General Intelligence) подразумевает создание систем, способных выполнять любую интеллектуальную задачу на уровне, сравнимом с человеческим. Несмотря на значительные достижения в области искусственного интеллекта и нейросетей, до создания настоящего AGI дело пока не дошло (или широкой публике это неизвестно).

Современные нейросети, к сожалению, являются узкоспециализированными системами, способными выполнять лишь определенные задачи. Например, нейросеть, обученная на распознавание изображений, не может одновременно участвовать в анализе текстов или принимать участие в беседе. Ей для этого потребуется дополнительное соответствующее новое обучение на других данных. Эта специализация, ограниченность нейросетей в понимании контекста действий и неспособность переносить знания из одной области в другую – главное отличие от гибкого человеческого интеллекта.

Несмотря на свои выдающиеся способности к обработке данных, современные нейросети требуют огромных объемов высококачественных данных для обучения. Если информация, которая используется для обучения, неполная или искаженная, это ведет к низкому качеству предсказаний и снижению надежности системы. В отличие от человека, который способен обучаться на ограниченных примерах и извлекать обобщения из неформального опыта, нейросети необходимо строгое количественное и качественное обучение. Например, если человеку, чтобы научиться отличать собак от волков, может хватить нескольких картинок и объяснений,

то нейросети для той же задачи потребуются десятки тысяч размеченных изображений. И если в этих данных будут ошибки или предвзятость, система усвоит эти искажения и будет выдавать ненадежные результаты. Это ограничение значительно сужает возможности нейросетей.

Самой сложной преградой для существующих нейросетей остается понимание контекста. Наше общение наполнено намеками, иронией, культурными отсылками и эмоциями. Нейросеть же, генерируя текст, оперирует статистикой слов, но не истинным пониманием. Она может написать грустное стихотворение, но не почувствует его грусть. Нейросети могут интерпретировать входные данные, но не обладают интуицией или чувством природы человеческих эмоций. Научить машину настоящему пониманию контекста – колоссальная задача.

Наконец, мы упираемся в философские и этические вопросы. AGI – это не только техническая, но и глубоко гуманитарная проблема. Кто будет нести ответственность за решения, принятые ИИ? Как гарантировать, что он не унаследует и не усилит человеческие предрассудки, заложенные в данных? Создание AGI потребует от нас ответа на эти вопросы.

Некоторые эксперты полагают, что квантовые вычисления могут привести к значительному прорыву в области AGI. Квантовые системы способны обрабатывать огромное количество данных одновременно, что может сделать их более эффективными в решении сложных задач. Однако, даже с развитием квантовых технологий, сама идея универсального интеллекта остается предметом междисциплинарных дебатов, которые требуют объединения знаний не только в области компьютерных наук, но и в философии, психологии и человеческой этике.

Таким образом, достижения современных нейросетей демонстрируют мощные инструменты для решения конкретных задач, однако интеграция этих технологий в концепцию универсального интеллекта предполагает решение множества проблем и вызовов. Шаги к созданию AGI будут происходить постепенно, и их успех займет время, однако стремление к этой цели вдохновляет ученых и инженеров работать над будущим, где ИИ будет способен действовать так же свободно и осмысленно, как человек.

Россия активно участвует в глобальной гонке как в области AGI, так и квантовых вычислений, делая ставку на государственную поддержку, фундаментальную науку и международное сотрудничество.

Развитие AGI и квантовых вычислений в мире вступило в решающую фазу. Сегодня мы переживаем переломный момент, когда технологии выходят за пределы лабораторий и начинают оказывать ощутимое влияние

на экономику и общество. Мировое технологическое лидерство определяется не только прорывами в области оборудования и алгоритмов, но и способностью формировать этические стандарты, обеспечивать кибербезопасность и выстраивать международные альянсы.

Российская Федерация занимает активную позицию в этой гонке, делая акцент на государственном стратегическом планировании, развитии фундаментальной науки и укреплении технологического партнерства с Китаем и другими странами. У России есть значительный научный задел и амбициозные планы, однако успех будет зависеть от эффективности реализации этих планов, способности интегрироваться в перестраивающуюся глобальную технологическую экосистему и конкурировать на мировой арене.

Ключевыми вызовами на пути к AGI и практическим квантовым вычислениям остаются не только технические сложности, но и решение этических проблем, вопросов безопасности и выработка адекватных правовых рамок, которые позволят использовать преимущества этих трансформационных технологий не во вред человечеству.

ГЛАВА 2.
ВКЛАД РОССИЙСКИХ УЧЕНЫХ

Погружаясь в историю развития информатики в России, невольно ловишь себя на мысли – все могло сложиться иначе. Будь у отечественных ученых возможность воплощать свои идеи в экономике, а главное – будь разрешена передача военных разработок гражданским инженерам, путь технологий оказался бы иным. Парадокс в том, что важность вычислительной техники для управления, прогнозирования и планирования осознавалась на самом верху. Яркое подтверждение – постановление Совета Министров СССР 1948 года, учредившее Институт точной механики и вычислительной техники (ИТМиВТ) АН СССР и Специальное конструкторское бюро № 245 (СКБ-245) для создания средств управления оборонными объектами. Казалось бы, процесс запущен. Однако развитие наталкивалось на сопротивление самих управленцев. Опасаясь за свои номенклатурные позиции и стремясь сохранить непрозрачность принимаемых решений, они тормозили прогресс. Многие перспективные проекты в области информатики и кибернетики были похоронены лишь потому, что угрожали чьему-то креслу.

Читателям, интересующимся историей информатики, рекомендую книгу «Очерки истории информатики в России»[3] под редакцией Д.А. Поспелова и Я.И. Фета. Здесь же вспомним самых ярких ученых, чей вклад в развитие информатики и теоретической кибернетики несомненен и оказал прямое влияние на развитие нейросетей и искусственного интеллекта.

§ 1. Советская школа информатики

Сергей Алексеевич Лебедев[4] (1902-1974) – основоположник вычислительной техники в СССР, в 1951 году создал первый в континентальной Европе компьютер с хранимой в памяти программой (МЭСМ) и был одним из разработчиков первых цифровых электронных вычислительных машин с динамически изменяемой программой вычислений.

[3] Очерки истории информатики в России [Текст] / Ред.-сост. Д. А. Поспелов. Я. И. Фет ; Ред. Д. А. Поспелов. – Новосибирск : Науч.- издат. Центр ОИГГМ СО РАН, 1998. – 662 с. : 16 с. ил. – 1000 экз.

[4] Е. Литвинова «Сергей Лебедев. битва за суперкомпьютер» https://ipmce.ru/about/press/publications/sergej-lebedev/

Под его руководством было создано 18 ЭВМ, причем 15 из них выпускались серийно. Лебедев пытался убедить руководство страны создавать собственную линию ЭВМ средней мощности и супер-ЭВМ нового поколения, но, к сожалению, победили оппоненты, которые предлагали создать ряд совместимых компьютеров, повторив американскую систему IBM.

Алексей Андреевич Ляпунов[5] (1911-1973) в 1953 году создал операторный метод программирования, ставший основой многих дальнейших работ по теории программирования. Ляпунов организовал первый в СССР научный семинар по кибернетике в МГУ, готовил издание сборников «Проблемы кибернетики». Его операторный метод, по существу являвшийся прообразом алгоритмических языков программирования, лег в основу всех методических учебных пособий по программированию.

Тогда же в 1950-ых годах работы **Сергея Львовича Соболева**[6] (1908-1989) заложили математический фундамент современной вычислительной математики. Созданные им пространства Соболева, теория оптимальных кубатурных формул и устойчивые разностные схемы для решения дифференциальных уравнений обеспечили функционально-аналитическую основу для разработки надёжных и точных численных алгоритмов. Этот мощный математический аппарат является не только основой вычислительной математики, но и критически важным инструментом «под капотом» алгоритмов машинного обучения, где требуются сложные численные методы и оптимизация.

В 1955 г. инженер **Анатолий Иванович Китов**[7] (1920-2005) совместно с академиками С.Л. Соболевым и А.А. Ляпуновым опубликовал статью «Основные черты кибернетики», в которой раскрывались фундаментальные идеи Норберта Винера о единстве процессов управления и коммуникации, демонстрировались аналогии между нервной системой живых существ и принципами работы автоматических машин, показывалось, что кибернетика опирается на математику, теорию информации, автоматическое управление и вычислительную технику. А.И. Китов также создал первые советские учебники по ЭВМ («Электронные цифровые машины», 1956) и разработал концепцию Единой государственной сети вычислительных центров (ЕГСВЦ, 1959) – прообраза современных облачных систем и Интернета, предусматривавшую распределенные вычисления для управления экономикой.

[5] Н.Н. Богуненко «Взять высоту» https://scfh.ru/papers/vzyat-vysotu/
[6] Биография С.Л. Соболева https://www.biblioatom.ru/persons/sobolev_sergey_lvovich/
[7] В.А. Долгов «Китов Анатолий Иванович – пионер кибернетики, информатики и автоматизированных систем управления» https://www.computer-museum.ru/books/dolgov_kitov_2010.pdf

Алексей Григорьевич Ивахненко[8] (1913-2007) разработал метод группового учета аргументов (GMDH, 1968) – алгоритм автоматического синтеза оптимальных моделей данных, суть которого сводится к «целенаправленному перебору» моделей разной сложности по ряду критериев, чтобы в итоге найти одну модель оптимальной структуры. Этот метод, который он подробно развивал в таких монографиях, как «Техническая кибернетика» (первое издание – 1959 г.) и «Самообучающиеся системы распознавания и автоматического управления» (1969), считается одним из исторических предшественников современных концепций автоматического машинного обучения (AutoML).

Виктор Михайлович Глушков[9] (1923-1982) развил идеи А.И. Китова, предложив Общегосударственную автоматизированную систему (ОГАС) – сеть, которая должна была связать все объекты народного хозяйства и позволить компьютеру самостоятельно принимать оптимальные решения на основе данных Госплана, то есть предлагалось автоматизированное управление всей экономикой страны, минимизирующее влияние человеческого фактора. Проект также предусматривал возможность прямого взаимодействия между предприятиями, минуя центральный узел. Проект не был реализован. Причинами называли сложность и высокую стоимость, а также сопротивление Центрального статистического управления (ЦСУ). Вместе с тем, работы В.М. Глушкова по теории алгоритмических языков и созданию АСУ заложили теоретические основы для будущих ERP-систем (SAP, Oracle) и концепции защищенных электронных денег, что можно считать концептуальным предвосхищением некоторых идей, позже нашедших развитие в технологиях вроде блокчейна.

Андрей Петрович Ершов[10] – один из основоположников теоретического и системного программирования в СССР, автор одной из первых программирующих программ (трансляторов) для отечественных ЭВМ БЭСМ и «Стрела». В 1958 году он опубликовал монографию «Программирующая программа для электронной вычислительной машины БЭСМ». А.П. Ершов создал Сибирскую школу программирования, переехав в 1960 году в Новосибирский Академгородок, где возглавил отдел программирования Института математики СО АН СССР. Под его руководством разработаны языки

---

[8] Алексей Григорьевич Ивахненко https://publ.lib.ru/ARCHIVES/I/IVAHNENKO_Aleksey_Grigor'evich/_Ivahnenko_A.G..html

[9] «Академик В.М. Глушков» https://glushkov.su/ogas

[10] «Андрей Петрович Ершов» Составители Н.А. Черемных, И.А. Крайнева Под редакцией д.ф.-м.н. А.Г. Марчука https://www.iis.nsk.su/files/book/file/bibliografiya_prep_140509.pdf

программирования «Альфа», «Альфа-6» и их оптимизирующие трансляторы, а также многоязыковая система «Бета» (внутренний язык для описания трансляторов) и открытый язык «Лексикон» для описания программ и предметных областей. В 1968 году американский учёный Джон Маккарти, создатель языка Lisp и один из основоположников направления «искусственный интеллект», посетил СССР. В Москве и Новосибирске он прочитал лекции по Lisp и автоматизации доказательств теорем, а также провел с А.П. Ершовым совместные работы, посвященные реализации Lisp на БЭСМ-6. Их сотрудничество стало важным стимулом для развития исследований в области символического ИИ и автоматического доказательства теорем в СССР.

Александр Иванович Галушкин[11] (1940-2016) – один из основателей российской школы нейроинформатики. Он систематизировал принципы обучения многослойных перцептронов, заложив тем самым теоретическую основу для алгоритмов, которые позже стали широко известны как алгоритмы обратного распространения ошибки (Backpropagation). Разработанные А.И. Галушкиным методы распознавания, прогнозирования и управления находят применение в современных интеллектуальных информационных и управляющих системах, используемых в науке, экономике, финансах, энергетике, робототехнике, транспорте, медицине и сфере безопасности. Важно, что А.И. Галушкин не только развивал теорию, но и создавал реальные устройства. В частности, он совместно с В.Х. Нариманов разработал первый в СССР нейрокомпьютер – аналоговое устройство, способное распознавать образы быстрее обычных ЭВМ.

Дмитрий Александрович Поспелов[12] (1932-2019) – основоположник советской школы искусственного интеллекта. Его ключевые заслуги включают создание:

– теории ситуационного управления, где решения принимаются на основе анализа текущей ситуации, а не жестких инструкций;

– семиотических моделей для сложных систем, представляющих знания через знаки (слова, символы, образы), что позволило создавать базы знаний, в которых машина понимает связи между понятиями;

– псевдофизических логик – формальных моделей для представления «здравого смысла»;

---

[11] Ясницкий Л.Н. О приоритете Советской науки в области нейроинформатики // XV Всероссийская научная конференция «Нейрокомпьютеры и их применение». Тезисы докладов. –М: ФГБОУ ВО МГППУ, 2017. С 16-19 https://publications.hse.ru/mirror/pubs/share/direct/317633580

[12] Тарасов В.Б. «Д.А. Поспелов – основоположник искусственного интеллекта в СССР и России» https://libeldoc.bsuir.by/bitstream/123456789/38692/1/Tarasov_Iskusstvenniy.pdf

– формальных моделей поведения, а также таких междисциплинарных направлений, как психоника (психология искусственных систем) и когнитология (изучение процессов познания);

– теории гиромата, предвосхитившей идеи современной теории агентов, и моделей децентрализованных многоагентных систем.

Его работы 1960–1980-х годов считаются новаторскими и во многих аспектах опережали зарубежные аналоги. Важным аспектом деятельности Д.А. Поспелова было междисциплинарное сотрудничество. Совместно с психологом Вениамином Ноевичем Пушкиным[13] (1931–1979), автором концепции оперативного мышления, он исследовал аналогии между творческими процессами человека и логикой машин. Их совместная монография «Мышление и автоматы» (1972) стала ключевой работой в этой области.

Российские исследователи с самого начала эры искусственного интеллекта заложили фундаментальные и прикладные основы современных нейросетей. Их работы не только выдвигают Россию в число мировых лидеров в области ИИ, но и с каждым годом демонстрируют всё новые прорывные результаты. Российская школа ИИ продолжает развивать традиции, трансформируя абстрактные модели XX века в технологии, определяющие научно-технический прогресс XXI века.

§ 2. Кто определяет будущее ИИ прямо сейчас

Российская наука в области искусственного интеллекта и нейросетей не стоит на месте – она активно развивается, создавая решения мирового уровня. Сегодня российские ученые не просто участвуют в глобальной гонке ИИ – они задают тренды, разрабатывают фундаментальные алгоритмы и внедряют их в реальные продукты, которыми пользуются миллионы людей.

Вот ключевые имена, проекты и направления, которые формируют лицо российского ИИ в XXI веке.

[13] Статья к 60-летию В.Н. Пушкина в журнале «Вопросы психологии» – http://www.voppsy.ru/issues/1991/914/914124.htm

Дмитрий Петрович Ветров (НИУ ВШЭ) – инженер эффективности нейросетей.

Его прорыв – разработка байесовских методов глубокого обучения и вариационного дропаута – технологий, которые позволяют делать нейросети меньше, быстрее и надежнее.

Представьте, что ChatGPT или Kandinsky стали бы работать в 100 раз быстрее, занимали в 1000 раз меньше памяти и при этом предупреждали вас, когда не уверены в ответе. Именно это и делают методы Ветрова.

«Умное упрощение» – его алгоритмы автоматически находят и отсекают «лишние» связи в нейросети – как если бы вы убрали из автомобиля все детали, которые не влияют на езду. Результат: сеть работает на смартфоне, а не только на сервере.

«Сжатие без потерь» – технология тензоризации позволяет «свернуть» гигантскую модель в компактный «архив», который можно «развернуть» без потери качества. Это как сжать 4K-фильм до размера SMS – и воспроизвести его в исходном качестве.

«Коллектив разума» – вместо одного «эксперта» (нейросети) система создает «комитет», где каждый член дает свой ответ и степень уверенности. Если мнения расходятся, система предупреждает: «Внимание, здесь высокая неопределенность!». Это критически важно в медицине, финансах, юриспруденции.

Методы Ветрова используются в реальных медицинских платформах для анализа МРТ и диагностики рака. Его работа – это мост между сложной математикой и практикой, которая спасает жизни и делает технологии доступнее.

Павел Владимирович Купцов (НИУ ВШЭ – Нижний Новгород) – «цифровой биолог».

Его прорыв – создание нейросети, которая с высочайшей точностью воспроизводит работу биологического нейрона, в частности, классическую модель Ходжкина-Хаксли, описывающую электрическую активность нейронов. Это шаг к созданию искусственного интеллекта, который работает не как традиционный компьютер, а как биологическая система. Такие сети смогут точно моделировать болезни (Альцгеймер, Паркинсон), тестировать лекарства in silico (в компьютерной симуляции) и создавать нейропротезы, которые «говорят» на языке мозга. Ключевое преимущество – нейросеть Купцова не только воспроизводит поведение бионейрона с точностью, неотличимой от реальных экспериментов, но и работает в сотни раз быстрее

численного моделирования. Это позволяет проводить симуляции, которые раньше занимали месяцы, всего за часы.

Практическое применение включает ускорение разработки лекарств от нейродегенеративных заболеваний и создание более точных моделей мозга для нейроинтерфейсов.

Александр Николаевич Безносиков (директор Исследовательского центра агентных систем искусственного интеллекта МФТИ), **Глеб Геннадьевич Гусев** (директор Центра практического искусственного интеллекта Сбера) – «архитекторы распределенного ИИ».

Проблема – чтобы обучить гигантскую нейросеть (например, GPT-подобную), нужны тысячи серверов, которые постоянно обмениваются данными. Это медленно, дорого и энергозатратно. Их прорыв – разработка алгоритмов сжатия градиентов и снижения коммуникационных затрат при распределенном обучении. Модели обучаются быстрее – новые функции в «Салюте» или «GigaChat» появляются чаще. Снижается нагрузка на серверы и энергопотребление – технологии становятся экологичнее. Даже слабые устройства (ноутбуки, телефоны) могут участвовать в обучении больших моделей. Это открывает путь к децентрализованному ИИ.

Исследование А.Н. Безносикова и Г.Г. Гусева «Ускоренные методы со сжатыми коммуникациями для гомогенных задач распределенной оптимизации» принято на ведущую мировую конференцию AAAI’25 – аналог Нобелевской премии в области ИИ.

Сергей Александрович Козлов (Институт проблем передачи информации РАН) – «мастер мультимодальности».

Прорыв – разработка архитектур, которые одновременно обрабатывают и связывают между собой текст, изображения и звук, подобно тому, как это делает человек. Это открывает путь к системам, которые понимают контекст. Например, если вы, глядя на экран, скажете голосовому помощнику «покажи мне это», ИИ поймёт, на какой именно объект вы ссылаетесь. Появятся «умные» редакторы, которые смогут по описанию «сделай ярче, как закат в Крыму» автоматически подобрать нужные цвета и эффекты.

Технологии станут доступнее. Например, система сможет не просто описать слепому пользователю картинку, но и передать её эмоциональный настрой.

Российская наука в области ИИ не стоит на месте. Помимо фундаментальных работ, она активно формирует практические решения будущего. Вклад России в ИИ характеризуется триадой «теория – практика – этика»: фундаментальные теории (групповой метод обработки данных,

динамические сети) стали основой для deep learning, прикладные решения (генетические модели, сжатие данных) интегрированы в глобальные платформы, социально-этические инициативы (авторство ИИ, регулирование) задают стандарты для научного сообщества.

Историческая преемственность от Анатолия Китова до Дмитрия Ветрова демонстрирует уникальную способность российской науки сочетать теоретическую глубину с прорывными практическими реализациями. Российская школа ИИ прошла путь от фундаментальных исследований в условиях железного занавеса до интеграции в мировую науку. Её вклад – не только исторические достижения (МГУА, шахматные алгоритмы), но и современные решения: эффективные коммуникации в распределённых системах, расшифровка ДНК через языковые модели, интуитивные ИИ для научного творчества. К 2030 году ожидается усиление роли России в нейроморфных вычислениях (чипы, имитирующие мозг) и квантовом машинном обучении. Эти разработки демонстрируют, как идеи, рождённые в советских НИИ, сегодня лежат в основе технологий, меняющих медицину, промышленность и повседневную жизнь.

ГЛАВА 3. СОВРЕМЕННЫЙ МИР И НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

В этой главе вы узнаете, какие фантастические измышления о нейросетях порождены безудержной фантазией тех, кто не разбирается в существе вопроса, и тех, кому по каким-то причинам выгоден ажиотаж вокруг ИИ, насколько эти мифы далеки о реальности, как в действительности ИИ меняет медицину, финансы, транспорт – и как в этом участвует Россия.

§ 1. Разрушаем мифы про нейросети

Мир искусственного интеллекта и нейросетей окутан множеством мифов, как будто паутиной. Их подпитывают фильмы вроде «Терминатора», страшилки из новостей и просто непонимание, как эта штука реально работает. Разберем самые популярные байки и узнаем правду, чтобы не вестись на фейки.

Миф 1 – ИИ может заменить людей во всех сферах деятельности!

Нейросети, действительно, крутые спецы, но только в своем узком деле. Они могут: молниеносно анализировать горы данных (как врач, просматривающий тысячи снимков), выполнять рутину (отвечать на простые чат-запросы, сортировать почту), но они не умеют по-настоящему творить, сопереживать или принимать сложные моральные решения. То есть хотя нейросети могут превосходить людей в определённых аспектах, таких как скорость обработки данных или выявление шаблонов в больших объемах информации, они не обладают универсальными человеческими качествами, такими как креативность, эмпатия или интуиция. Нейросети не обладают гибкостью человеческого мышления, здравым смыслом и способностью к переносу знаний из одной области в другую. Нейросеть, блестяще играющая

в шахматы, не сможет самостоятельно научиться играть в покер, не говоря уже о решении бытовых проблем. Эта фундаментальная ограниченность – не просто временное препятствие, а принципиальное свойство современных ИИ-систем, которое сохранится в обозримом будущем.

Миф 2 – нейросети «думают» как люди!

Настоящий интеллект включает в себя самосознание, эмоциональный интеллект и субъективный опыт, к которым нейросети имеют весьма отдалённое отношение. Их «мышление» основывается исключительно на алгоритмах и математических моделях. Нейросети не обладают сознанием, они лишь имитируют работу мозга, опираясь на математику и статистику.

Миф 3 – нейросети работают без ошибок!

На самом деле, как и любая другая технология, они подвержены погрешностям. Ошибки могут возникать как из-за недостатка или качества обучающих данных, так и вследствие неправильной интерпретации результата. Нейросеть для распознавания лиц может ошибиться, если фото плохое или она «не доучилась», ChatGPT иногда выдает «фейки» (галлюцинации), причем делает это крайне убедительно. Почему? Потому что нейросеть учится на данных. Мусор на входе = мусор на выходе.

Миф 4 – ИИ объективен и справедлив!

На практике алгоритмы могут наследовать предвзятости данных, на которых они обучались. Если в обучающем наборе присутствует предвзятая информация, нейросеть может воспроизводить и усиливать эти предвзятости в своих выводах. Например, если нейросеть училась на данных, где большинство руководителей – мужчины, то она может несправедливо оценивать женские резюме. ИИ не волшебник, он отражает мир, в котором «вырос». Нейросети не просто наследуют, но могут и усиливать социальные предубеждения, представленные в обучающих данных. Исследования показали, что некоторые системы распознавания лиц работают с разной точностью для людей разного пола и расы. Языковые модели часто воспроизводят гендерные стереотипы, например, предполагая, что врач – мужчина, а медсестра – женщина. Российские исследователи из НИУ ВШЭ

в 2023 году выявили, что даже модели, обученные на русскоязычных данных, демонстрируют статистически значимую предвзятость в отношении определенных социальных групп. Это не вина разработчиков, это следствие того, что нейросети обучаются на реальных данных, а реальный мир содержит неравенство и предрассудки. Осознавая эту проблему, команды разработчиков применяют методы «дебиасинга» (устранения предвзятости) и более сбалансированного подбора обучающих данных. Однако полностью решить эту проблему пока не удается. Поэтому важно критически оценивать ответы ИИ-систем, особенно когда речь идет о чувствительных социальных вопросах или решениях, влияющих на жизнь людей. Это превращает вопросы этики и социальной ответственности в ключевые аспекты разработки ИИ.

Миф 5 – ИИ сделает высшее образование ненужным!

ИИ – крутой репетитор-помощник. Нейросети уже объясняют сложные темы простыми словами, генерируют тренажеры и задачи с учетом уровня подготовки пользователя, проверяют шаблонные задания (грамматику, типовые расчеты) быстрее человека. Но человеку для развития и становления нужен человек-наставник. Университеты будущего (согласно отчету McKinsey, 2024[14]) – это, прежде всего, акцент на навыках, которые ИИ не освоит – креативное решение проблем, междисциплинарные исследования, этическое лидерство, а также лаборатории как хабы для испытания ИИ-гипотез (например, AI Talent Hub от Университета ИТМО, MIT Media Lab). В этой связи всё более важной становится уникальная роль преподавателя – формирование «цифрового иммунитета», т.е. способности ставить корректные задачи ИИ и критически оценивать результат. В перспективе преподаватель – это куратор индивидуальных образовательных траекторий в гибридной среде.

Развенчивание мифов о нейросетях – это первый шаг к их более широкому и осознанному принятию обществом. Знания о реальных возможностях и ограничениях искусственного интеллекта помогут нам лучше использовать технологии и справляться с возникающими вызовами, открывающимися перед нами в новом цифровом мире.

[14] Источник: https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-2024

§ 2. Как искусственный интеллект меняет наш мир

Нейросети – не абстракция, а рабочие инструменты, решающие конкретные задачи эффективнее человека в областях, где требуется анализ гигантских данных.

Медицина

Нейросети помогают в диагностике заболеваний, анализе медицинских изображений (например, рентгеновских снимков и МРТ), предсказании эпидемий и разработке персонализированных методов лечения.

Возможно, самые впечатляющие результаты применения нейросетей мы видим в медицине. Google DeepMind разработала систему, которая по снимкам сетчатки глаза может определить более 50 различных заболеваний[15]. Что особенно важно – система обрабатывает один снимок за секунды, в то время как традиционный анализ ОКТ-снимков занимает значительно больше времени.

Еще более революционным стало создание нейросети для диагностики рака молочной железы. В статье «Международная оценка системы искусственного интеллекта для скрининга рака молочной железы»[16], опубликованной в журнале «Nature», приводятся данные о системе искусственного интеллекта, которая способна превзойти специалистов-людей в прогнозировании рака молочной железы. В статье «Экономика искусственного интеллекта и распределение задач между людьми для принятия решений при скрининговой маммографии»[17] исследователи делают важный вывод, что ключевым фактором, определяющим оптимальное использование ИИ (т.е. или сочетание человека и машины, или полная автоматизация) является распространённость заболевания. То есть использование ИИ для редких заболеваний затруднительно, поскольку нет достаточной базы для обучения. Поэтому применение единого подхода ко всем заболеваниям может оказаться неэффективным. Медицинским организациям рекомендовано

[15] Источник: https://hightech.fm/2018/08/13/eye-diseases

[16] МакКинни С.М., Синек М., Годбол В. *и др.* Международная оценка системы искусственного интеллекта для скрининга рака молочной железы. *Nature* **577**, 89–94 (2020). https://doi.org/10.1038/s41586-019-1799-6

[17] Ахсен М.Э., Айвачи М.У.С., Мукерджи Р. *и др.* Экономика искусственного интеллекта и распределение задач между людьми для принятия решений при скрининговой маммографии. *Nat Commun* 16, 2289 (2025). https://doi.org/10.1038/s41467-025-57409-1 общедоступная ссылка: https://rdcu.be/eIgyo

адаптировать внедрение ИИ и оптимизацию рабочих процессов под конкретные особенности каждого конкретного заболевания.

Настоящим прорывом стал проект AlphaFold[18] от Google DeepMind, который сделал возможным предсказание структуры белков с беспрецедентной точностью. Чтобы понять значимость этого проекта представим, что белок – это очень сложная конструкция, в которой имеет значение не только из каких блоков она состоит и в какой последовательности эти блоки собраны, а за это в нашем организме отвечают ДНК («инструкция») и РНК («сборщик по инструкции»). Последовательно собранные в цепочку аминокислоты в соответствии с ДНК-«инструкцией» – это первичная структура белка. После сборки белковой цепи она сразу начинает принимать трёхмерную форму. Этот процесс определяется стремлением всей системы к наиболее выгодному энергетическому состоянию. Конечная форма однозначно задаётся для каждого атома его зарядом, химическими связями и физико-химическими свойствами как самой цепи, так и окружающей её среды. Первые, самые простые и повторяющиеся способы, которыми белковая цепь сворачивается, – это уже вторичная структура белка. Третичная структура белка – это уже итоговая трехмерная форма одной белковой цепи. А когда несколько таких цепей (уже в виде третичных структур) объединяются в один функциональный комплекс, это называется четвертичной структурой белка (например, гемоглобин состоит из четырех цепей).

И вот самое интересное! От того какую форму примут в итоге белки полностью зависит их функциональное назначение и возможности. Например, антитела выглядят как «буква Y» и именно благодаря такой форме белки захватывают вирусы, а ферменты имеют на своей поверхности особые «карманы», как отвёртка имеет жало, чтобы вкручивать винты (проводить химические реакции), и т.д. При этом саму итоговую трехмерную форму конкретного белка в данной среде однозначно определяет первичная структура, т.е. последовательность аминокислот. Это один из главных принципов биологии. Информация о том, как свернуться, «зашита» в саму последовательность.

Именно эту сложнейшую задачу – предсказать, как линейная цепочка аминокислот (первичная структура) свернется в сложную 3D-глобулу (третичную структуру) – и решает AlphaFold. AlphaFold научился «расшифровывать» инструкцию, зашифрованную в первичной структуре. Он анализирует последовательность аминокислот и с высочайшей точностью вычисляет, каким образом (в какую форму) все эти заряды и физико-

[18] https://www.alphafold.com/

химические свойства заставят цепь свернуться. То есть, он предсказывает трехмерную форму по первичной структуре. Таким образом, если мы знаем форму белка, мы понимаем, как он работает. А это открывает огромные возможности. Например, борьба с болезнями и создание лекарств. Зная точную структуру белка вируса, можно на компьютере смоделировать и создать молекулу лекарства, которая к нему прицепится. Так создают многие современные лекарства (например, против ВИЧ, COVID-19). Также становится возможным изучить генетические болезни, т.к. часто мутация в гене («инструкции») приводит к тому, что белок-ключ складывается неправильно и не может выполнять свою функцию. Понимая, как именно мутация меняет форму, мы можем найти способ это исправить. В целом, поскольку белки – это «рабочие лошадки» жизни и управляют практически всеми процессами в каждой клетке нашего тела, то узнавая их структуру, мы буквально читаем «инструкцию по эксплуатации» живого организма и понимаем, как мы на молекулярном уровне дышим, двигаемся, думаем. При этом, зная правила «сворачивания», можно не только предсказывать структуру, но и конструировать новые белки с нуля, например, создать ферменты, которые разлагают пластик, или белки, которые эффективно производят биотопливо, или новые материалы, которые будут прочнее стали.

Предсказание формы – крайне сложная задача, потому что белок сворачивается в зависимости от огромного числа факторов и на то, чтобы просчитать все возможные варианты сворачивания для одной цепи даже на суперкомпьютерах раньше могло потребоваться тысячи лет. AlphaFold с помощью искусственного интеллекта научился делать это за считанные минуты и с высочайшей точностью.

Теперь представим, что мы не просто хотим предсказать форму существующего белка, а хотим создать новый белок с нуля, который будет делать что-то конкретное – например, связываться с вирусом. Этот процесс называется белковый дизайн и воспроизводит схему расшифровки наоборот, т.е. вначале с помощью тех же методов (например, того же AlphaFold) получаем точную 3D-структуру мишени – например, белка-шипа на поверхности вируса (мы знаем каждую его впадинку и выступ, все заряды), затем ИИ проектирует аминокислотную последовательность (первичную структуру белка), который свернется в требуемую форму, которая точно сможет с ним сцепиться. Созданную на компьютере последовательность «печатают» в лаборатории (синтезируют ген и получают из него белок) и проверяют, работает ли он как задумано. Часто требуется несколько циклов компьютерного дизайна и экспериментальной проверки.

Собственно, ровно за эти два прорыва в 2024 году Нобелевскую премию получили Демис Хассабис и Джон Джампер (AlphaFold) (за предсказание структуры – «Вот последовательность аминокислот. Скажи, в какую форму свернется этот белок и как он будет работать?») и Дэвид Бейкер (за вычислительный дизайн белков – «Мне нужен белок, который точно свяжется с этой частью вируса. Спроектируй мне последовательность аминокислот, которая свернется в белок нужной формы»).

В России активно развиваются собственные медицинские нейросети. «Третье мнение»[19] – российская компания, резидент фонда «Сколково», создала систему анализа медицинских изображений, которая определяет патологические изменения органов грудной клетки. Компания работает уже 7 лет и ее продукты используются более чем в 50-ти регионах России. На сегодняшний день алгоритмы компании обработали свыше 10 миллионов медицинских исследований. Помимо анализа рентген-снимков, КТ и маммографии, платформа включает и другие сервисы, например, система видеоаналитики для предупреждения падений и травм пациентов в стационарах, доступ врачей к ИИ-сервисам, образовательным курсам и сообществу коллег.

Система «Цельс»[20] от компании «Медицинские скрининг-системы» выявляет признаки туберкулеза и ранжирует исследования по серьезности патологий. Система уже работает в 16 регионах России и помогает врачам не пропустить критически важные случаи. Сервисы «Цельс» обработали уже более 11 миллионов исследований.

Однако важно помнить о существенных ограничениях нейросетевых систем в медицине. Нейросети часто работают как «черный ящик», не объясняя причины своих решений, а знание причин критически важно для медицинской практики. Многие модели демонстрируют неравномерную точность для разных демографических групп, поскольку обучены на несбалансированных данных. Они также не способны учитывать уникальные особенности пациента, выходящие за рамки их обучающих данных. Согласно исследованию, опубликованному в JAMA (2023), до 67% рекомендаций ИИ-систем в клинической практике требуют существенной корректировки врачом. Это подчеркивает их роль как вспомогательного инструмента, а не замены медицинскому персоналу.

[19] Источник: https://thirdopinion.ai/
[20] Источник: https://celsus.ai/?ysclid=mg0tvd22nx406637337

Финансы

В финансовом секторе нейросети используются для анализа рыночных тенденций, прогнозирования цен на акции, выявления мошенничества и оптимизации инвестиционных стратегий.

Финансовый сектор стал одной из первых отраслей, где нейросети начали приносить миллиардные прибыли. Легендарный хедж-фонд Renaissance Technologies, основанный математиком Джеймсом Саймонсом, показывал среднюю доходность 66% годовых в течение 30 лет – это исключительный результат, не имеющий аналогов в современной финансовой истории. Для наглядности его стоит сравнить с другими эталонами рынка.

Как видно из Таблицы 1, Medallion Fund зарабатывал для своих клиентов в среднем около 39% годовых на протяжении 30 лет. Это значительно больше, чем смогли заработать даже самые знаменитые в мире инвесторы.

Таблица 1. «Сравнение с легендами инвестиций»

Управляющий / Фонд	Среднегодовая доходность	Период
Medallion Fund (Renaissance Technologies)	**~39%** (нетто)	1988–2018 гг.
Уоррен Баффет (Berkshire Hathaway)	~21%	1965–2018 гг.
Джордж Сорос (Quantum Fund)	~32%	1969–2000 гг.
Питер Линч (Magellan Fund)	~29%	1977–2000 гг.
Индекс S&P 500 (широкий рынок)	~10%	(ориентировочно)

Уникальность результата заключается не только в высокой цифре, но и в подходе к его достижению. В отличие от инвесторов вроде Уоррена Баффетта, которые анализируют бизнес-показатели компаний, Renaissance Technologies использовал сложные математические и статистические модели для выявления краткосрочных и слабозаметных закономерностей на рынке. Команда фонда состояла в основном из математиков, физиков и специалистов по обработке данных, а не из финансистов. Фонд с самого начала устанавливал очень высокие комиссии – 5% от активов под управлением и 44% от прибыли. Именно поэтому инвесторы получали на руки «чистую» доходность около 39%, в то время как брутто-доходность составляла около 66%. Важно отметить,

что Medallion Fund закрыт для внешних инвесторов с 1993 года. Вкладывать деньги в него могут только сотрудники компании и их семьи. Это означает, что феноменальная доходность, по сути, является внутренним результатом команды создателей.

Российские банки входят в число передовых в применении ИИ-технологий. Альфа-Банк первым в России внедрил китайскую нейросеть DeepSeek R1 и разработал собственную платформу генеративного ИИ AlfaGen[21]. За эти инновации банк получил международную награду от сингапурского издания Fortune Times как технологический лидер мирового уровня[22].

Сбербанк развивает собственную мультимодальную нейросеть GigaChat – российский конкурент ChatGPT. Банк использует собственные модели машинного обучения для поиска мошеннических цепочек переводов между клиентами и борьбы с финансовыми преступлениями. Согласно данным Ассоциации ФинТех, генеративный ИИ (как, например, GigaChat или Kandinsky) будет меняться еще сильнее[23]. Появляются «умные помощники» (как агентские функции в GigaChat или ассистенты Т-Банка), которые не просто отвечают на вопросы, а сами выполняют задачи и составляют планы. ИИ становится более гибким (как адаптивные голосовые роботы ОТП Банка или AI-модели Сбера), научившись подстраиваться под меняющиеся условия. Всё это работает на удобных платформах (таких как экосистема GigaChat или платформа Шедеврум), которые сделают технологии искусственного интеллекта доступнее, безопаснее и проще в использовании. Объем вложений крупнейших российских банков в ИИ-решения составляет около 1 млрд долл. в год, а прибыль от внедрения этих решений достигает 3 млрд долл. в год. Речь идет только о крупных игроках рынка – небольшие финансовые организации также используют ИИ, но могут инвестировать в них ежегодно не более 100–300 млн рублей.[24]

МТС создала систему Big Data МТС Скоринг, которая в реальном времени предупреждает банки о попытках мошенничества.[25] Система использует передовые методы машинного обучения и ежедневно обновляет данные о вероятности мошеннической деятельности.

---

[21] Источник: https://alfabank.ru/news/t/release/alfa-bank-rasskazal-o-primenenii-tekhnologii-alfagen-v-programmakh-loyalnosti/?utm_referrer=https%3a%2f%2fyandex.ru%2f

[22] Источник: https://www.forbes.ru/brandvoice/537281-mirovoj-uroven-al-fa-bank-priznali-tehnologiceskim-liderom-v-singapure?ysclid=mg17ag1ha327969223

[23] Источник: https://www.fintechru.org/upload/iblock/1cc/05lnkjnor7yxrrqdi64mu181f67fy767.pdf

[24] Источник: https://tass.ru/ekonomika/18908529

[25] Источник: https://moskva.mts.ru/about/media-centr/soobshheniya-kompanii/novosti-mts-v-rossii-i-mire/2024-12-09/big-data-mts-razrabotala-reshenie-dlya-preduprezhdeniya-bankov-o-popytkah-moshennichestva-v-realnom-vremeni

17 ведущих российских финансовых компаний, включая Сбер, Альфа-Банк, ВТБ, Т-Банк, Яндекс и Московскую биржу, объединились в клуб «ИИ в финансовой отрасли» для разработки первой методики оценки эффективности ИИ-алгоритмов[26].

Транспорт

Нейросети помогают оптимизировать маршруты, предсказывать загруженность дорог и улучшать безопасность движения. Они также используются в системах автономного вождения.

Автомобильная индустрия переживает революцию благодаря нейросетям. Tesla использует подход «только камеры» – 8 камер и нейросети, обученные на миллионах километров реального вождения. В июне 2025 года Tesla начала тестировать сервис роботакси в Остине на ограниченной партии автомобилей Model Y. Машины двигаются автономно, но в салоне по-прежнему присутствует сотрудник компании для контроля за безопасностью[27].

Waymo выбрала противоположную стратегию – максимальную чувствительность сенсоров[28]. Ежедневно производственная линия выпускает партию белых электрических внедорожников Jaguar I-PACE. Каждый автомобиль снабжён комплексом систем для автономного управления, включающим специальные компьютеры, камеры, радары и лазерные лидарные датчики (технология дистанционного зондирования для измерения точных расстояний и движения в окружающей среде в режиме реального времени), разработанные компанией. 2024 год стал переломным для компании Waymo. После многих лет испытаний (с 2009 года) и трёх раундов инвестиций, в ходе которых было собрано более 11 миллиардов долларов, компания наконец-то стала полноценным бизнесом. Важно отметить, что помимо этих инвестиций, Google также вложил в проект дополнительные миллиарды в период с 2009 по 2020 год.

Яндекс совершил прорыв в автономном вождении, впервые применив нейросеть-Трансформер для планирования траектории движения беспилотных автомобилей[29]. Система обучена на данных команды высококлассных

---

[26] Источник: https://www.fintechru.org/press-center/news/v-rossii-sozdadut-pervuyu-natsionalnuyu-metodiku-otsenki-finansovykh-effektov-ot-iskusstvennogo-inte/

[27] Источник: https://autoreview.ru/news/sluzhba-tesla-robotaxi-kak-proshel-zapusk

[28] Источник: https://www.forbes.com/sites/alanohnsman/2025/05/05/inside-the-waymo-factory-building-a-robotaxi-future/

[29] Источник6 https://yandex.ru/company/news/03-12-12-2024

водителей, прошедших строгий отбор и курс контраварийного вождения. Новый планировщик делает манеру вождения автономных машин более «человечной», приближенной к профессиональным водителям. Яндекс развивает систему автономного вождения с 2017 года, тестируя транспорт в Москве, Иннополисе и Сириусе. В октябре 2024 года грузовик с системой автономного вождения Яндекса совершил первую коммерческую перевозку по трассе М-4 «Дон».

Образование

Применение нейросетей в образовании вызывает много споров. Опасения обусловлены некорректным применением обучающимися нейросетей, таким, например, как подмена письменной работы, выполненной школьником или студентом, работой, которую от начала и до конца сгенерировала нейросеть по запросу. То есть в этом случае речь идет не об использовании возможностей нейросети для глубокого изучения материала, а именно об «имитации обучения». Это безусловно не пойдет на пользу и приведет к деградации и недоразвитию интеллектуальных функций обучающегося, у которого изначально шанс развить их все-таки был.

Однако нейросети могут быть очень удобным и квалифицированным консультантом, ассистентом и экспертом в обучении, если применять их грамотно и этично. Образовательные организации, которые своевременно осознали, что если не можешь противостоять, то возглавь, вполне эффективно внедряют технологии ИИ в образовательный процесс, повышая вовлеченность студентов и обучая их экологичному использованию нейросетей. Например, в 2024 году студенты гуманитарных направлений НИУ ВШЭ впервые официально использовали такие возможности YandexGPT при написании курсовых и дипломных работ, как поиск и структурирование информации, написание и проверка текстов, изучение сложных тем, и т.п.

Применение технологий ИИ в российском образовании постепенно стандартизируется – принято несколько профильных ГОСТов[30]. Созданы

[30] ГОСТ Р 59895-2021 «Технологии искусственного интеллекта в образовании. Общие положения и терминология»; ГОСТ Р 59896-2021 «Образовательные продукты с алгоритмами искусственного интеллекта для адаптивного обучения в общем образовании. Требования к учебно-методическим материалам»; ГОСТ Р 71657–2024 «Технологии искусственного интеллекта в образовании. Функциональная подсистема создания научных публикаций»; ГОСТ Р 70947-2023 «Технологии искусственного интеллекта в образовании. Функциональная подсистема управления успеваемостью обучающихся по программам среднего профессионального образования»; ГОСТ Р 70948-2023 «Технологии искусственного интеллекта в образовании. Функциональная подсистема формирования контингента абитуриентов по программам бакалавриата и специалитета»; ГОСТ Р 59900-2021 Системы искусственного интеллекта. Типовые

нейросетевые продукты, которые помогают составить расписание в образовательной организации (например, Система «Галактика РУЗ»[31]), Общемировым трендом становится персонализация обучения. В России развиваются адаптивные технологии, например, платформы «Учи.ру» и «Яндекс.Учебник», которые используют алгоритмы для анализа ошибок и автоматического подбора тренировочных упражнений. За рубежом также популярны адаптивные обучающие платформы (например, Khan Academy), которые используют машинное обучение для подстройки содержания и темпа обучения под каждого пользователя. Технологии персонализации активно развивает Coursera. Персонализация обучения с помощью ИИ позволяет каждому студенту получать индивидуальную программу с учетом уровня знаний, скорости обучения и профессиональных интересов.

Промышленность

В промышленности нейросети внедряют предиктивное обслуживание – подход, который позволяет предсказывать поломки оборудования до их возникновения (например, Siemens[32)].

Российская платформа CyberStudio заменяет ушедшие с рынка зарубежные продукты как GE Smart Signal и Honeywell Forge, предоставляя отечественным предприятиям инструменты для предиктивной аналитики промышленного оборудования и оптимизации технологических процессов.[33] Отечественные разработчики создают решения, которые анализируют внешние и внутренние факторы, оперативно реагируя на случаи потенциальных поломок оборудования. Такие системы становятся востребованными среди российских промышленных предприятий в условиях импортозамещения.

Безопасность

В области безопасности нейросети используются для распознавания лиц, анализа видеопотоков и выявления подозрительной активности.

Системы распознавания лиц, которые позволяют идентифицировать (установить) и верифицировать (подтвердить) личность человека стали повсеместными в обеспечении безопасности. Например, Amazon Rekognition

требования к контрольным выборкам исходных данных для испытания систем искусственного интеллекта в образовании» и др.

[31] Источник: https://galaktika.ru/ruz?ysclid=mg7zmbgxbw210237801

[32] Источник: https://www.m-electro.com/news/detail.php?ID=107

[33] Источник: https://cyberphysics.xyz/cyberstudio

анализирует короткие видео-селфи для обнаружения подделок – печатных фотографий, цифровых видео, 3D-масок и даже дипфейков. Однако технология вызывает серьезные этические вопросы. В 2021 году Amazon продлила мораторий на использование Rekognition полицией из-за частых ошибок и обвинений в предвзятости по отношению к чернокожим людям.

Российские компании разрабатывают собственные системы видеоаналитики и распознавания. Уже упомянутая ранее платформа «Третье мнение» создала сервис видеоаналитики для мониторинга состояния пациентов в больничных палатах. Система анализирует видеопоток из отделений реанимации и интенсивной терапии, помогая медперсоналу не пропустить критически важные события: падения, длительное отсутствие в палате, судороги. Нейросети мгновенно отправляют push-уведомления, в том числе на умные часы медсестер, ускоряя реакцию на тревожные события в 50 раз. Такой подход к обеспечению безопасности пациентов демонстрирует, как российские технологии искусственного интеллекта могут применяться для защиты жизни и здоровья граждан.

Системы распознавания лиц внедряются и в другие сферы. В частности, в Санкт-Петербурге в рамках проекта «Безопасный город» в 2025 году введена система видеонаблюдения с функцией распознавания этнической принадлежности людей для мониторинга миграционных процессов и обеспечения безопасности[34].

Однако, страны постепенно начинают ужесточать законодательство в этой сфере во избежание возможной дискриминации и ущемления прав граждан. В частности, в Китае с 1 июня 2025 г. вступили в действие «Меры по управлению безопасностью при применении технологии распознавания лиц», которые сформулированы в соответствии с законодательством о сетевой безопасности, безопасности данных, защите личной информации и другими нормативно-правовыми актами, регламентирующими эту деятельность. Вероятно, этот опыт будет воспринят и другими странами.

Искусство и культура

Нейросети могут генерировать произведения искусства, анализировать культурные данные и помогать в реставрации произведений искусства.

ИИ произвел революцию в создании и реставрации произведений искусства. Midjourney и DALL-E 2 генерируют потрясающие изображения

[34] Источник: https://lenta.ru/news/2025/08/26/v-peterburge-nachali-rabotat-raspoznayuschie-natsionalnost-videokamery/

по текстовым описаниям, причем Midjourney показывает более художественный стиль, а DALL-E 2 лучше понимает естественную речь.

Ученые из Массачусетского технологического института (MIT) разработали технологию реставрации, которая ускоряет восстановление поврежденных картин в десятки раз. Картина XV века, которая традиционно потребовала бы около 200 часов реставрации, была восстановлена всего за несколько дней.[35]

Россия создала собственные конкурентоспособные нейросети для генерации изображений. «Шедеврум» от Яндекса позволяет создавать уникальные изображения, тексты и видео по описаниям на русском языке. Нейросеть использует технологии YandexArt и YandexGPT, обученные на 330 миллионах изображений. Kandinsky от Сбера – еще одна российская нейросеть для генерации изображений, которая содержит 3,3 миллиарда параметров. Обе российские нейросети демонстрируют впечатляющие результаты, которые приближаются по качеству к зарубежным аналогам.

В области реставрации российские музеи также применяют ИИ. Государственный Эрмитаж использует искусственный интеллект для определения авторства и датировки произведений. В музее-заповеднике «Царское Село» студенты с помощью нейросетей воссоздают утраченные панно Зубовского флигеля Екатерининского дворца.

Вывод

Нейросети представляют собой мощный инструмент, который кардинально меняет подходы к решению сложных задач во всех сферах человеческой деятельности – от спасения жизней в медицине до создания произведений искусства. Искусственный интеллект открывает новые возможности, которые еще недавно казались фантастикой.

Россия демонстрирует впечатляющие достижения в разработке и внедрении ИИ-технологий. Российский рынок медицинских ИИ-систем может вырасти в 6 раз к 2030 году – с 12 миллиардов рублей в 2024 году до 78 миллиардов рублей[36]. Отечественные компании создают конкурентоспособные решения, которые не уступают, а иногда и превосходят зарубежные аналоги.

---

[35] Источник: https://tech.news.am/rus/news/5659/ii-revolyuciya-v-restavracii-kak-neiyroseti-mogut-pomoch-vosstanavlivat-kartiniy-v-desyatki-raz-biystree.html

[36] Источник: https://yakovpartners.ru/publications/ai-healthcare/

Главное преимущество нейросетей – их способность обрабатывать огромные объемы данных и выявлять закономерности, недоступные человеческому восприятию. Это позволяет решать задачи с точностью и скоростью, которые превосходят возможности человека, при этом освобождая специалистов для более творческой и стратегической работы.

Мы находимся только в начале эры искусственного интеллекта. Каждый день появляются новые возможности для применения нейросетей, их потенциал для улучшения качества жизни человечества поистине безграничен. Российские разработки в этой области показывают, что страна способна не только адаптировать зарубежные технологии, но и создавать собственные инновационные решения мирового уровня.

§ 3. Почему ИИ - больше, чем мода

В последние годы искусственный интеллект превратился из модного термина в силу, кардинально изменяющую экономику планеты. Но скептики продолжают сомневаться: не является ли ИИ очередным «пузырем»?

С одной стороны, цифры говорят обратное. По оценкам Global Market Insights Inc. [37], объем рынка «искусственного интеллекта как услуги» (AIaaS) составил $12,7 миллиардов в 2024 году и, как ожидается, будет расти на 30,6% ежегодно до 2034 года. Для сравнения, аналитики IDC оценивают общие глобальные расходы на ИИ, включая «железо» и инфраструктуру, в значительно большую сумму – свыше $500 млрд. По оценкам аналитиков и СМИ, крупнейшие технологические компании (Microsoft, Amazon, Alphabet, Meta и др.) в 2025 году направят свыше $300 млрд на развитие искусственного интеллекта – главным образом на инфраструктуру: дата-центры, вычислительные мощности и облачные сервисы[38]. Для сравнения: это сопоставимо с ВВП средних по размеру стран. Такой масштаб инвестиций показывает, что речь идёт не о временной моде, а о формировании новой технологической основы экономики.

С другой стороны, в октябре 2025 г. глава JP Morgan Джейми Даймон заявил о серьёзной рыночной коррекции и грядущих потерях в ИИ-индустрии, а аналитик независимой исследовательской фирмы Macrostrategy Partnership Джулиан Гарран заявил, что сектор искусственного интеллекта превратился

[37] Источник: https://www.gminsights.com/ru/industry-analysis/ai-as-a-service-market; идентификатор отчета GMI5714; дата публикации: March 2025

[38] Источник: https://www.appercase.ru/news/47371/

в самый масштабный спекулятивный пузырь, который когда-либо видели финансовые рынки, и может привести к серьезным экономическим потрясениям. Кто прав – покажет время. Но то, что ИИ имеет долговременное значение, документально подтверждено его прорывными способностями в решении сложных задач. В частности, уже упоминались нейросети DeepMind от Google (высокая точность диагностики рака молочной железы), AlphaFold2 (предсказание структуры белков). В отличие от предыдущих технологических трендов, ИИ не просто автоматизирует рутинные операции, он превосходит возможности человека в некоторых интеллектуальных задачах. Это качественно новый этап развития технологий.

ИИ способен создавать качественно новые решения в науке и технологиях. Google DeepMind создала модель GenCast, которая превосходит традиционные методы прогнозирования погоды на срок до 15 дней по 97,2% из 1320 переменных. Это позволяет точнее предсказывать экстремальные погодные явления и оптимизировать работу возобновляемых источников энергии. Генеративные нейросети ускоряют научные исследования в разы. Уже упоминалась технология реставрации картин, разработанная учеными Массачусетского технологического института (MIT), которая значительно сокращает время восстановления. В разработке лекарств ИИ сокращает время от лабораторных экспериментов до клинических испытаний на годы. Вокруг ИИ-технологий возникают новые профессии и отрасли.

ИИ становится ключевым инструментом в борьбе с изменением климата – проблемой, которую невозможно решить без технологических прорывов. По данным британской аудиторской компании PwC, использование ИИ в энергетике и транспорте способно сократить глобальные выбросы парниковых газов на 4% к 2030 году. DeepMind помогла Google сократить расход энергии на охлаждение дата-центров на 40%, а системы управления ветряными электростанциями увеличили их эффективность на 20% за счет точного прогнозирования силы ветра.

В России запущена ФГИС «Экомониторинг», использующая ИИ для поиска незаконных свалок, анализа качества воздуха и управления отходами. В 2023 году Российский экологический оператор заключил с ООО «Сбер Бизнес Софт» соглашение, в рамках которого будут созданы ИТ-решения с применением искусственного интеллекта для мониторинга несанкционированных свалок.

Президент России Владимир Путин поставил задачу обеспечить наличие собственных разработок нового поколения ИИ как условие научного и технологического суверенитета. На поддержку исследовательских центров

ИИ до 2030 года выделяется около 4,5 миллиарда рублей. Создан Центр развития искусственного интеллекта (ЦРИИ) при Правительстве России для налаживания взаимодействия между государством, регионами и бизнесом. Это системный подход к развитию технологий, учитывающий социальные и этические аспекты.

Искусственный интеллект – это фундаментальная технология, определяющая облик экономики XXI века. В отличие от предыдущих технологических циклов, ИИ воздействует одновременно на все сферы жизни, но существует серьезный разрыв между маркетинговыми обещаниями и реальной эффективностью. Несмотря на многомиллиардные инвестиции, большинство внедрений ИИ в бизнесе не достигают заявленных целей. По данным исследования Gartner (2024), до 85% проектов по внедрению ИИ не переходят от пилотного этапа к полноценному развертыванию. Ключевые проблемы включают трудности с интеграцией в существующие системы, недостаточное качество данных и завышенные ожидания от технологии. Это указывает на необходимость более реалистичного взгляда на возможности нейросетей в ближайшей перспективе.

ГЛАВА 4.
ОСНОВЫ НЕЙРОСЕТЕЙ

В этой главе вы узнаете, из чего состоит нейросеть (нейроны, слои, связи), какие бывают архитектурные типы нейросетей и для каких задач они созданы, как нейросети обучаются на данных и почему это знание критически важно для пользователя, а не только для программиста.

§ 1. Нейросеть как компания

Для понимания, что такое архитектура нейросети, можно провести аналогию с организационной структурой компании, которую разрабатывает менеджер для выполнения конкретной бизнес-задачи.

Менеджер (разработчик) должен решить: сколько отделов (слоев) потребуется, будет ли это плоская структура с одним отделом или сложная иерархия с множеством подразделений, какая специализация у каждого отдела (тип слоя), будет ли это отдел анализа данных, отдел креатива или отдел проверки качества, как эти отделы взаимодействуют (связи между слоями), кто кому передает отчеты, работают ли отделы строго по цепочке или есть отделы-координаторы, которые общаются со всеми, сколько сотрудников (нейронов) будет в каждом отделе и т.д.

Проще говоря, вы не будете строить одинаковую структуру для стартапа по разработке игр и для крупного металлургического завода. Структуру определяет задача, ради которой она создается.

Архитектура нейросети – это и есть продуманная «организационная структура» для искусственного интеллекта, которая максимально эффективно решает его задачу.

Итак, в нашей аналогии:

нейросеть – это вся компания;

архитектура – это полная организационная схема всей компании;

слои – подразделения компании, которые выполняют специфичные задачи;

нейрон – это должность или сотрудник, который выполняет простую операцию, а именно – выслушивает мнения коллег из предыдущего отдела, обобщает их и выносит своё суждение;

связи между нейронами – это каналы коммуникации между этими должностями;

веса (параметры) – это степень влияния, с которой мнение одного сотрудника учитывается другим (например, мнение «старшего аналитика» – нейрона предыдущего слоя может иметь большой «вес» для мнения «менеджера» – нейрона текущего слоя, а мнение «стажёра» – небольшой);

обучение нейросети – это процесс, в ходе которого модель пошагово корректирует именно эти веса (параметры влияния), чтобы минимизировать ошибки. Подобно тому как новички в компании постепенно понимают, кому из коллег стоит доверять больше, а кому меньше, и сами нарабатывают свой авторитет.

Изначально нейросеть абсолютно не готова к работе. Все ее «сотрудники» – новички, которые не знают, кто чего стоит и насколько его мнение приоритетно, то есть не определены правила взаимодействия (веса). «Сотрудники» работают впустую, обращаясь не к тому, кто является экспертов в вопросе, и совершают много ошибок. Нейросеть выдает неверные ответы. Но у нас есть «ментор» – это тренировочные данные, состоящие из пар: изображение и его правильная метка (например, фотографии кошек и собак с указанием кошка это или собака). Процесс обучения выглядит так: нейросети показывают картинку («Смотри, это кошка»), она пытается ее распознать своим текущим «кривым» способом («Я думаю, это собака»), ментор дает обратную связь («Нет, ты ошибся. Вот насколько сильно ты ошибся»).

С помощью специального алгоритма – *обратного распространения ошибки* (Backpropagation) система вычисляет, как изменение каждого веса влияет на общую ошибку. Затем применяется метод градиентного спуска, который корректирует веса так, чтобы уменьшить ошибку. Если вес, по мнению системы, усиливает ошибку, то его значение уменьшают. Если он помогает уменьшить ошибку, то его значение увеличивают. Это можно сравнить с тем, как руководитель показывает каждому сотруднику, насколько его действия повлияли на результат, и дает направление для улучшения. Этот процесс повторяется сотни тысяч раз для всех примеров в данных. В итоге, после такого интенсивного обучения, каждый параметр сети («сотрудник») находит свою роль, и вся система в целом начинает работать эффективно, позволяя нейросети безошибочно отличать кошек от собак.

Итак, разработчик архитектуры – это менеджер, который придумал структуру компании, а процесс обучения – это интенсивная подготовка, которая превращает кучу новичков в эффективную команду.

§ 2. Классификация нейросетей

Нейросети можно классифицировать по архитектуре/модели и по аппаратной платформе. Важно учесть, что любая архитектура может (теоретически!) работать на любой платформе, но с разной эффективностью.

A. По архитектуре и функциональности

1. Классические искусственные нейронные сети (Artificial Neural Networks, ANN):

– Перцептроны и сети прямого распространения (Feedforward Neural Networks, FNNs) – самые простые, информация течет строго от входа к выходу, это базовый строительный блок.

– Сверточные нейронные сети (Convolutional Neural Networks, CNN) – «короли» компьютерного зрения, используют свертки для выявления паттернов в данных с сеточной структурой (изображения).

– Рекуррентные нейронные сети (Recurrent Neural Networks, RNN) – имеют «память», подходят для последовательностей данных (временные ряды, текст). Их подвиды: LSTM, GRU.

– Автокодировщики (Autoencoders) – используются для сжатия данных и обучения без учителя.

– Генеративно-состязательные сети (Generative Adversarial Networks, GAN – две сети соревнуются, одна генерирует данные, другая оценивает их подлинность.

2. Трансформеры (Transformers)

Трансформеры – фундаментальная архитектура на основе механизма внимания, стала основой для современных больших языковых моделей, современный стандарт для NLP (обработки естественного языка).

– Encoder-модели (например, BERT) – для задач классификации, понимания текста,

– Decoder-модели (например, GPT) – для генерации текста,

– Encoder-Decoder модели (например, T5, BART) – для задач «текст-текст» (перевод, суммаризация).

3. Диффузионные модели (Diffusion Models)

Диффузионные модели – современные «короли» генерации изображений (Stable Diffusion, DALL-E 2, Midjourney), учатся генерировать данные, постепенно удаляя шум из случайного набора точек (процесс обучения идет в обратную сторону: модель учится предсказывать шум, который был добавлен к изображению на каждом шаге, а затем использует это знание, чтобы генерировать изображение, последовательно очищая его от шума), являются альтернативой GANам, но часто превосходят их в качестве и стабильности генерации.

4. Мультимодальные модели (Multimodal Models)

Мультимодальные модели – это архитектуры, способные одновременно воспринимать и обрабатывать информацию из разных модальностей (текст, изображение, аудио, видео) для единого кросс-модального представления информации.

Например:

– CLIP (Contrastive Language–Image Pre-training) связывает изображения и их текстовые описания. Стала ключевым компонентом для многих генеративных моделей (например, в Stable Diffusion для управления генерацией изображений по текстовому запросу).

– DALL-E и Imagen генерируют изображения по текстовому описанию.

– Flamingo, GPT-4V (с Vision) – это модели, которые принимают на вход и текст, и изображения, и могут отвечать на вопросы о них.

5. Спайковые нейронные сети (Spiking Neural Networks, SNN)

Спайковые нейронные сети – отдельный класс моделей, которые имитируют работу биологического мозга и используют импульсы и временные коды.

Таблица 2. «Эффективность различных архитектур нейронных сетей на разных аппаратных платформах»

Модель / Архитектура	CPU (Программная реализация)	GPU/TPU (Аппаратные ускорители)	Нейроморфный чип
Классические ANNs (CNN, RNN)	Медленно. Подходит для прототипирования и обучения небольших моделей.	Идеально. Высокая параллелизация вычислений обеспечивает максимальную производительность.	Архитектурный конфликт. Непрерывные вычисления и синхронная передача данных несовместимы с событийной, асинхронной природой нейроморфных чипов.
Трансформеры, LLM	Непрактично. Чрезвычайно медленно из-за огромного объема вычислений и памяти.	Идеально. Единственная практичная платформа для обучения и инференса больших моделей.	Архитектурный конфликт. Матричные операции и механизм внимания требуют массовой параллельной обработки данных, а не редких спайков.
Диффузионные модели	Непрактично. Чрезвычайно медленно из-за десятков или сотен последовательных итеративных шагов, которые CPU не может распараллелить.	Идеально. Многократные итеративные вычисления эффективно распараллеливаются на тысячах ядер.	Архитектурный конфликт. Итеративный процесс шумоподавления основан на плотных матричных операциях, а не на разреженной событийной активности.
Мультимодальные модели	Непрактично. Чрезвычайно медленно, т.к. требуется одновременная обработка нескольких модальностей (текст, изображение), что создает непосильную нагрузку для CPU.	Идеально. Способность одновременно обрабатывать различные типы данных (текст, изображения) на высокопроизводительных вычислительных ядрах.	Архитектурный конфликт. Как правило, состоят из Трансформеров и других плотных архитектур, не адаптированных под событийную обработку.
Спайковые сети (SNNs)	Эмуляция. Крайне медленно, но пригодно для исследований и отладки алгоритмов.	Эмуляция. Значительно быстрее, чем на CPU. Подходит для R&D.	Идеально. Аппаратная реализация, совпадающая с событийной природой SNN, обеспечивает максимальную энергоэффективность и скорость.

6. Развитие нейросетей

Можно представить эту классификацию не как строгое дерево, а как последовательные, но частично перекрывающиеся волны развития, тогда:

волна 1 – это классические ANNs (перцептроны, CNN, RNN), которые решали конкретные задачи с определенными типами данных;

волна 2 – Трансформеры, они совершили революцию в NLP, применив механизм внимания и масштабируемость;

волна 3 – генеративные и мультимодальные модели, которые строятся на основе Трансформеров и других архитектур, чтобы генерировать сложный контент и свободно работать с разными типами данных одновременно (текст + изображение и т.д.).

Параллельная ветвь – это SNNs, которые развиваются как альтернативный путь к ИИ, основанный на принципах работы биологического мозга.

Б. По аппаратной платформе («железу»)

1. **Программная реализация** (на CPU) – универсально, гибко, но медленно для больших моделей.
2. **Аппаратные ускорители** (GPU, TPU, FPGA) – специализированы на параллельных матричных вычислениях, идеальны для обучения и работы традиционных ANNs (CNN, RNN, Transformers).
3. **Нейроморфные чипы** (Loihi, SpiNNaker, TrueNorth и др.) – специализированы на событийной, асинхронной обработке, идеальны для спайковых сетей (SNNs).

Любую модель можно попытаться запустить на любом «железе», но эффективность будет разной (Таблица 2). Например, Трансформер можно запустить на CPU, на GPU (идеально) и даже на нейроморфном чипе, что будет крайне неэффективно. Спайковую сеть можно запустить на CPU, на GPU (эмуляция), но это будет неэффективно, и на нейроморфном чипе – идеально. В качестве аналогии, иллюстрирующей эффективность работы спайковой нейросети на разных аппаратных платформах, можно привести видеопоток с камеры наблюдения, где большую часть времени ничего не происходит: на CPU/GPU придется обрабатывать каждый кадр видео, даже те, где на экране ничего не происходит, а это огромная трата ресурсов; на нейроморфном чипе обработка начнется только в тот момент, когда в кадре появится движение

(событие). Это и есть событийно-управляемое вычисление, которое делает систему крайне эффективной.

§ 3. Как работают нейросети на обычном компьютере

Чтобы нейросеть работала на ПК пользователя, он должен быть либо подключен к ней, либо на него должна быть установлена сама модель. Обученная модель — это специальный файл с «знаниями» (так называемыми «весами»), полученными в процессе тренировки на мощных серверах.

Существует три основных сценария взаимодействия:

1) Облачный сервис (ПК как терминал)

Пользователь подключается к нейросети через интерфейс (браузер или приложение). Все вычисления происходят на удаленных серверах владельца нейросети. ПК пользователя лишь отправляет запрос и получает готовый ответ, не производя математических операций модели. Примеры: Яндекс GPT, GigaChat, Кандинский (через веб-интерфейс).

2) Локальное выполнение (ПК как вычислитель)

Обученная модель устанавливается непосредственно на жесткий диск ПК пользователя. Все вычисления выполняются на его оборудовании (CPU, GPU или NPU). Для работы не требуется Интернет. Примеры: локальная версия Stable Diffusion, некоторые специализированные медицинские программы для анализа снимков.

3) Гибридный сценарий (ПК как партнер)

Сложная модель работает в облаке, но часть задач делегируется на ПК пользователя. Это происходит, если компьютер оснащен специализированными компонентами для ИИ (аппаратными ускорителями), что позволяет ускорить работу и снизить нагрузку на серверы. Пример: мобильное приложение может использовать встроенный в процессор NPU для предварительной обработки изображения, а затем отправлять уже оптимизированные данные в облако для финального анализа.

Таким образом, современный компьютер может выступать в разных ролях — от простого терминала до полноценной вычислительной платформы для работы с искусственным интеллектом.

§ 4. Основные компоненты нейросетей: нейроны, слои, связи

A. Нейроны

Нейрон в мозге человека (биологический нейрон) – это живая клетка, которую можно увидеть в микроскоп. У неё есть физическое «тело»: мембрана, ядро, отростки – дендриты и аксон. Логичный вопрос: а как выглядит его искусственный собрат? Есть ли у него такое же «тело»?

Оказывается, ключевое отличие в том, что у искусственного нейрона нет единого, обязательного материального воплощения. Его основа – математическая функция, формула. А вот материальный субстрат (то есть физическая основа, которая выполняет эту формулу) может быть разным. Представьте себе музыкальную мелодию. Её суть – это ноты. Мелодию можно сыграть на рояле, на гитаре или целым оркестром. Но у самой мелодии нет постоянного «тела» – её субстратом становится любой инструмент, способный её воспроизвести.

Так и искусственный нейрон может существовать в трех принципиально разных «формах», в зависимости от своего субстрата.

1. Нейроны в программной реализации

Нейроны в программной реализации (в памяти и на процессоре компьютера) – это самый частый случай. У такого нейрона нет постоянной физической «формы».

Нейрон в программной реализации – это виртуальная сущность. Его «существование» обеспечивается данными (весами) в памяти и кодом, который выполняет процессор.

Физически внутри компьютера нет никаких «микролампочек-нейронов». Есть только универсальные транзисторы процессора, которые в нужный момент, получив инструкцию, перемножают и складывают числа, представляющие собой входные данные и веса. После вычисления результат сохраняется в памяти, а выделенная для этих расчетов область оперативной памяти освобождается, т.е. виртуальный «нейрон» прекращает свое существование до следующего вызова.

Поэтапно процесс рождения и существования нейрона в программной реализации выглядит следующим образом:

1. Формирование статического скелета (модель и весá)

Изначально в память компьютера в виде программного обеспечения *загружается архитектура нейросети* (код, который описывает, какие функции и как выполнять), которая определяет:

– количество и тип слоев (входной, скрытые, выходной);

– количество нейронов в каждом слое;

– схему соединений между ними (полносвязная, сверточная и т.д.);

– функции активации для каждого нейрона или слоя;

– и самое главное – матрицы весов и векторы смещений, которые были получены в процессе обучения (это «знания» и «опыт» сети, ее память).

На этом этапе сеть подобна книге на полке: вся информация есть, но она не «живет».

2. Динамическое воплощение – прямой проход / прямое распространение (Forward Pass)

Когда на вход системы поступают данные (например, изображение), запускается процесс прямого прохода или иначе − прямого распространения (Forward Pass).

Вот как это происходит с точки зрения «жизни» сети:

входной слой «пробуждается» – данные преобразуются в числовой вектор и загружаются в память, временно занимая место «входных нейронов» (эти нейроны часто не имеют вычислений – они просто хранят данные);

волна активации – процессор (или GPU) начинает логически последовательно, слой за слоем, «оживлять» нейроны (при этом внутри каждого слоя вычисления часто выполняются параллельно), т.е. он берет выходы предыдущего слоя (из памяти), умножает их на матрицу весов, прибавляет смещение и применяет функцию активации, а результат записывается в память для активации текущего слоя;

эффект домино – эта волна вычислений проходит через все скрытые слои, причем каждый слой – это не просто набор нейронов, а этап преобразования информации от распознавания простых линий и текстур в первых сверточных слоях до сборки из них сложных паттернов, таких как глаза или колеса, в последующих (например, в сверточной нейросети первые слои могут «активироваться» на простые границы, а последующие – на сложные формы и объекты);

результат – выходной слой выдает итоговый вектор (например, вероятности того, что на картинке «кошка», «собака» или «автомобиль»).

3. Исчезновение и новое рождение

После того как результат получен, активации промежуточных слоев (значения «нейронов» в момент вычислений) могут быть освобождены из памяти для экономии ресурсов, и нейросеть «засыпает», оставаясь лишь в виде статической модели и весов в памяти, но при поступлении новых данных весь цикл повторяется: виртуальный организм снова пробуждается, чтобы выполнить свою функцию.

Ключевой момент – физически в процессоре нет специализированных блоков для нейронов. CPU/GPU – это универсальный «организатор иллюзий». Его транзисторы в один момент времени могут вычислять взвешенную сумму для одного виртуального нейрона, в следующий – выполнять функцию активации для другого, а затем – перемножать матрицы для целого слоя.

Нейросеть в программной реализации – это не физическая структура, а строго регламентированный процесс, *алгоритм, который временно создает на ресурсах универсального компьютера сложную, специализированную вычислительную систему*, состоящую из взаимосвязанных нейронов. Ее существование циклично и неразрывно связано с моментом обработки данных.

Программный нейрон в компьютере – это процесс, воплощающийся в момент обработки данных специальным программным обеспечением, и лишенный физической формы. Но в одиночку такой нейрон беспомощен. Его сила проявляется только в коллективе, организованном в нейронную сеть.

На уровне программной реализации нейросеть – это сложный, динамичный и виртуальный вычислительный граф, разворачивающийся в памяти компьютера. Ее можно представить как гигантскую математическую функцию, распределенную во времени и пространстве – между ядрами процессора и областями памяти. Этот виртуальный граф состоит из вершин/узлов (нейроны или целые слои) и рёбер/связей между ними (числовые веса, определяющие вклад каждого сигнала в работу нейрона).

Граф – это абстрактная математическая структура, которая помогает описывать связи между объектами. Проще всего представить его как карту, где вершины – это «пропускные пункты», а рёбра – это «дороги», которые их соединяют.

Как графы связаны с нейросетями?

В контексте нейросетей вершины графа – это нейроны или целые слои нейронов, а рёбра – это связи между ними, каждая из которых имеет свой «вес» – числовое значение, определяющее, насколько сильно сигнал от одного нейрона влияет на другой.

Представим, что нейросеть — это транспортная сеть между пропускными пунктами, т.е. у нее имеются:

входные пропускные пункты (x_1, x_2) – это данные, которые подаются на вход (например, параметры запроса),

промежуточные пропускные пункты (h_1, h_2, h_3) – это «скрытые» нейроны, которые обрабатывают информацию,

конечные пропускные пункты (y_1, y_2) – это результаты работы сети (например, сгенерированный текст или решение),

дороги (ребра) с разной пропускной способностью (весом) – широкая современная трасса (большой вес) пропускает много «транспорта» (сигнала), а узкая просёлочная дорога (малый вес) — мало.

Компьютер не рисует кружочки и стрелочки. Вместо этого он хранит граф в виде списков или матриц связей – таблиц, где записано, какие вершины соединены и с каким весом, и алгоритмов, которые «проходят» по этим связям в определённом порядке.

Ключевая идея – *граф как динамический процесс*, т.е. работу нейросети можно представить в виде движения данных по дорогам (ребрам) через пропускные пункты (вершины).

Данные поступают на входной пункт и в зависимости от своих весов «движутся» по подходящим дорогам к промежуточным пунктам. Там они обрабатываются, и результат передается дальше, пока не достигнет конечного пункта, откуда «выезжает» готовый ответ.

На каждом шаге («слое») нейросеть выполняет операции над данными, используя веса рёбер как коэффициенты влияния (умножая и складывая). Промежуточные результаты передаются по цепочке, пока не будет сформирован итоговый ответ. Этот процесс называется «разворачиванием графа», из компактного математического описания компьютер динамически строит цепочку вычислений. Именно эта возможность – превращать сложные связи в последовательность простых шагов делает компьютеры мощным инструментом для реализации искусственных нейронных сетей.

Таким образом, графы – это не красивые схемы, а фундаментальный язык, на котором компьютеры «понимают» и выполняют сложные расчёты, лежащие в основе работы современных нейросетей.

2. Нейроны на специализированном аппаратном обеспечении (AI-ускорители)

Вторая форма – это *специализированное аппаратное обеспечение (AI-ускорители)*, такие как Tensor Processing Units (TPU), разработанный компанией Google для задач машинного обучения, или Neural Processing Units (NPU) – специализированный чип, оптимизированный под нейросети.

Здесь нейрон начинает обретать черты физической формы. Его субстратом выступают специализированные микросхемы, чья архитектура «заточена» под нейросетевые операции (матричные умножения). Архитектура чипа напрямую отражает структуру вычислений и предполагает наличие множество ядер, способных параллельно выполнять умножение и сложение. Субстратом таких чипов является физическая компоновка (топология) транзисторов на кремниевой подложке, организованная так, чтобы максимально эффективно вычислять взвешенные суммы и применять функции активации. Например, в смартфонах и планшетах NPU часто интегрируются в системные чипы (SoC), но могут быть и отдельными ускорителями. На кристалле такого чипа есть физические блоки транзисторов, которые напрямую соответствуют вычислительным блокам нейросети.

Ключевое отличие от обычного процессора (CPU) в том, *как* происходит вычисление.

На CPU процессор – универсал. Он берет команды и данные из основной оперативной памяти (ОЗУ), обрабатывает их в своих ядрах и возвращает результат обратно в ОЗУ.

На AI-ускорителе (TPU/NPU) процесс выглядит иначе. Нейросеть на аппаратном ускорителе – физическое воплощение вычислительного графа. В отличие от CPU, где граф существует только как последовательность команд в памяти, в AI-ускорителе архитектура самого чипа физически отражает типичную структуру нейросетевых вычислений. Модель нейросети (её архитектура, представленная в виде последовательности операций, и весá) находится в основной памяти компьютера (ОЗУ). Кстати, в основной памяти компьютера (ОЗУ) может одновременно находиться сколько угодно моделей нейросетей (например, одна − для распознавания речи, другая − для генерации текста, третья − для стилизации фото). Они лежат там как файлы или загруженные в память данные, ожидая своего запуска.

По специальной команде подготовленная версия этой модели (программа и весá) копируется непосредственно в высокоскоростную память, встроенную в сам AI-чип. Когда модель копируется в высокоскоростную память

ускорителя, происходит не просто трансфер данных, а *трансляция абстрактного графа в физическую конфигурацию чипа*, в котором:

1. *Веса распределяются по специализированным банкам памяти*, расположенным в непосредственной близости от вычислительных блоков (принцип «Near Memory Computing»).
2. *Структура нейросети (последовательность слоёв) преобразуется в микрокод или конфигурационные регистры*, которые управляют потоком данных через физические конвейеры.
3. *Вычислительные блоки настраиваются под конкретные операции*:

– одни блоки конфигурируются как матричные умножители,

– другие – как блоки свёртки,

– третьи – как активационные функции.

Данные начинают «протекать» через физические вычислительные конвейеры чипа. Эти конвейеры – не универсальные ядра, а тысячи «станков», настроенных только на умножение и сложение матриц. Они работают массово-параллельно.

«Жизнь» нейросети в кремнии – это данные в движении, т.е. в момент выполнения происходит не вычисление «нейрона за нейроном», а непрерывный поток данных через вычислительный «конвейер».

Таким образом, «нейросеть» в момент работы на ускорителе – это программа и весá, загруженные в высокоскоростную память самого чипа, а также данные, которые физически «протекают» через специализированные аппаратные блоки (транзисторы), настроенные на сверхбыстрое выполнение матричных операций.

Как уже упоминалось, в памяти компьютера может храниться множество моделей, и ускоритель может поочередно выполнять их. Одновременный запуск нескольких моделей обычно невозможен на большинстве потребительских ускорителей – они, как однополосная дорога, обрабатывают задачи по очереди. Однако мощные серверные чипы могут иметь несколько «полос» для по-настоящему параллельной работы.

«Потрогать нейрон» на TPU/NPU – значит потрогать сам чип. Это уже не виртуальный процесс, а реальный кусок кремния (Таблица 3). Важно понять, что это универсальный и переконфигурируемый вычислитель для нейросетей, а не одна «зашитая» в кремний сеть. Его физическая структура предопределяет, что выполнять нейросетевые операции он будет исключительно эффективно.

Таблица 3. «Сравнение нейронов в программной реализации и на аппаратных ускорителях»

Характеристика	Программная реализация (CPU/GPU)	Аппаратный ускоритель (NPU/TPU)
Природа нейрона	Временный процесс	Физический конвейер
Вычисления	Последовательные инструкции	Поток данных через конвейеры
Память	Единая ОЗУ + кеш	Распределённая Near-Memory Computing архитектура, при которой вычислительные модули размещаются как можно ближе к памяти
Энергоэффективность	Низкая (общие затраты)	Высокая (специализация)
Гибкость	Полная (любые алгоритмы)	Ограниченная нейросетевыми операциями

3. Нейроны в нейроморфных чипах

Третья форма существования нейрона – это *нейроморфные чипы* (субстрат: аналоговые и цифро-аналоговые схемы). Это экспериментальные системы, которые пытаются максимально подражать работе мозга. Субстратом здесь являются физические компоненты, чьи электронные свойства (напряжение, сопротивление) напрямую имитируют поведение нейронов и синапсов.

Например, *мемристор* – это микроскопический компонент, который «запоминает» свойство (сопротивление) в зависимости от прошедшего через него тока. Он физически и является аналогом синапса – связи между нейронами. Его можно рассматривать как «материализовавшуюся» математическую функцию. Здесь уже можно говорить о некой «морфологии», похожей на биологическую.

Помимо мемристоров в нейроморфных чипах применяются и другие элементы:

кандистоны (Capacitive elements) – компоненты, которые накапливают и высвобождают заряд, имитируя интеграцию потенциала в биологическом нейроне (по сути, это конденсаторы, работающие в определенном режиме);

транзисторы, работающие в слабом режиме инверсии (Sub-threshold transistors) – это транзисторы, работающие при очень низком напряжении, ниже их стандартного порога включения, в этом режиме их поведение становится экспоненциальным и начинает напоминать поток ионов через мембрану нейрона, что позволяет создавать очень энергоэффективные схемы, генерирующие спайки;

фазово-переключаемые материалы (Phase-Change Materials, PCM) – материалы, которые могут переключаться между аморфным и кристаллическим состоянием, что значительно меняет их электрическое сопротивление, как и мемристоры, они могут служить аналогом синапса с программируемой силой связи;

спинтронные устройства (Spintronic devices) – используют спин электронов, а не их заряд, для хранения и обработки информации, такие устройства могут быть крайне энергоэффективными и быстрыми.

Нейроморфные чипы – это родная стихия спайковых нейронных сетей Spiking Neural Networks (SNN), или импульсных сетей. Это третье поколение искусственных нейросетей, которое значительно ближе к биологическим принципам работы мозга.

Ключевое отличие спайковых нейросетей от традиционных искусственных нейросетей заключается в *темпоральности* – информация кодируется не только в интенсивности сигнала, но и в точном времени возникновения спайков (Таблица 4). Это позволяет обрабатывать временные последовательности данных намного эффективнее традиционных подходов.

Здесь нейрон – это уже не статичная функция, а событийно-управляемая временна́я модель.

Если в предыдущих случаях нейрон постоянно «вычислял» что-то, то здесь он «спит» и активируется лишь коротким импульсом (спайком) в точно определенный момент времени. Информация кодируется не амплитудой сигнала, а временны́ми интервалами между спайками, их частотой и последовательностью. Это делает спайковые нейросети гораздо более энергоэффективными, так как вычисления происходят только в момент прихода импульса, а не непрерывно. Кроме того, они асинхронны – каждый нейрон работает независимо, что кардинально отличает их от слоистых синхронных архитектур традиционных искусственных нейросетей. Именно асинхронность и разреженность активности (активен малый процент нейронов в любой момент времени) – это ключ к их потенциальной энергоэффективности, а не только сам факт использования спайков.

Таблица 4. «Три поколения нейросетей»

Характеристика	Сети 1-го поколения (персептроны)	Сети 2-го поколения (ANN, DNN)	Сети 3-го поколения (SNN)
Принцип передачи сигнала	Бинарные значения (0/1)	Непрерывные значения активации	Импульсы (спайки)
Кодирование информации	Факт активации	Амплитуда сигнала	Время между импульсами
Биологическое правдоподобие	Низкое	Умеренное	Высокое
Энергоэффективность	Низкая	Низкая	Очень высокая (потенциал на 2-3 порядка выше, уже доказано на прототипах)
Аппаратная реализация	ЦПУ, ГПУ	ЦПУ, ГПУ, TPU, NPU	ЦПУ, ГПУ (программная симуляция), нейроморфные чипы (Intel Loihi 2, IBM TrueNorth)

Материальный субстрат спайковых нейронов:

1. Такой нейрон может существовать *в программной реализации* – это сложный алгоритм, симулирующий временную динамику на обычном CPU/GPU. «Потрогать» его нельзя, это чистая математика событий.
2. Второй вариант – это *нейроморфные чипы*, где физические компоненты (такие как мемристоры, фазово-переключаемые материалы (PCM) или транзисторы, работающие в особом режиме) используются не как традиционные вычислители, а в качестве прямых аналогов биологических нейронов и синапсов, генерирующих и передающих импульсы. Вот их уже можно считать самой осязаемой, «физической» версией искусственного нейрона.
3. И, наконец, материальным субстратом спайковых нейронов могут быть *гибридные решения* – сочетание цифровой и аналоговой обработки.

Обучение спайковых нейросетей строится на правилах, максимально приближенных к тем, что действуют при обучении биологического нейрона, таких как STDP (Spike-Timing Dependent Plasticity) – пластичность, зависящая от времени прихода спайков (импульсов). STDP ослабевает или усиливает связь между нейронами на основе временны́х интервалов между спайками, а именно,

если пресинаптический нейрон генерирует импульс перед постсинаптическим нейроном, синапс усиливается, что способствует увеличению веса связи, а если после постсинаптического нейрона, то синапс ослабевает, что приводит к уменьшению веса связи. После генерации спайка нейрон входит в рефрактерный период, т.е. промежуток времени, в течение которого нейрон не чувствителен к входным воздействиям после генерации спайка (импульса), аналогично биологическому нейрону. Продолжительность рефрактерного (нечувствительного) периода в биологических нейронах составляет несколько миллисекунд, что ограничивает максимальную частоту генерации спайков и влияет на пропускную способность информации в нейронной сети.

Моделирование спайковых нейросетей на классических компьютерах сопряжено с высокими вычислительными затратами, поскольку симуляция асинхронных событий и временны́х паттернов плохо ложится на синхронную архитектуру CPU/GPU, заточенную под параллельные операции с матрицами. Именно поэтому для спайковых нейросетей создаются нейроморфные процессоры – специализированные чипы, архитектура которых на уровне аппаратной реализации повторяет структуру сети из спайковых нейронов.

Спайковые нейросети служат важным концептуальным и практическим мостом между биологическим интеллектом и искусственным, предлагая альтернативный, более «природный» и потенциально крайне энергоэффективный путь развития искусственного интеллекта. В то время как AI-ускорители (TPU/NPU) эмулируют нейросети, т.е. имитируют их работу в программной или аппаратной форме, нейроморфные системы непосредственно реализуют SNN в аппаратуре.

Спайковые нейросети завершают *триаду подходов к реализации искусственного интеллекта*:

1. программная реализация – максимальная гибкость;
2. AI-ускорители – баланс эффективности и универсальности;
3. нейроморфные SNN – биологическая аутентичность и предельная энергоэффективность.

Итак, искусственный нейрон может быть виртуальной инструкцией в памяти компьютера, физическим блоком на кристалле кремния или аналоговым компонентом, меняющим свои свойства. Его «тело» определяется тем устройством, которое его вычисляет. Это фундаментальное отличие, которое делает ИИ программным, а не биологическим феноменом (Таблица 5).

Главное сходство биологического нейрона и нейрона в нейросети заключается в том, что оба являются базовыми единицами сложной сети, оба суммируют входные сигналы и выдают нелинейный ответ на основе этой

суммы. Оба учатся путем изменения силы связей (синапсов/весов). А главное их различие в том, что биологический нейрон – это невероятно сложная, живая, автономная система, а искусственный нейрон – это крайне упрощенная математическая модель одной его ключевой идеи. Современные нейросети достигают впечатляющих результатов не потому, что их нейроны сложные, а потому, что их очень-очень много, и они организованы в сложные слоистые структуры, способные выявлять сложнейшие закономерности в данных.

Таблица 5. «Сравнение биологического и искусственного нейронов»

Характеристика	**Биологический нейрон**	**Искусственный нейрон (CNN, RNN)**	**Спайковый нейрон (SNN)**
Природа и основа	Биологическая клетка со сложной морфологией.	Детерминированная математическая функция, абстрагированная от «железа».	Математическая модель, имитирующая биологию.
Принцип работы	Аналоговый, электрохимический. Градиенты ионов и потенциалов.	Цифровой. Последовательные операции с числами (матрицами).	Событийный. Информация в виде временны́х импульсов (спайков).
Передача сигнала	Электрохимические импульсы через синапсы.	Передача чисел (векторов/матриц) между слоями.	Временны́е интервалы между дискретными импульсами.
Механизм обучения	Синоптическая пластичность (изменение силы синапсов).	Корректировка весов (W) и смещений (b) (например, градиентный спуск).	Временна́я зависимая пластичность (STDP) — изменение синапсов на основе времени прихода спайков.
Скорость	Медленная (миллисекунды).	Очень высокая (определяется тактовой частотой процессора).	Потенциально очень высокая за счет асинхронности и низкого энергопотребления.
Энергоэффективность	Чрезвычайно высокая (~20 Вт для мозга).	Низкая (особенно при обучении больших моделей).	Потенциально на 2-3 порядка выше, чем у традиционных ИНС.
Параллелизм	Массовый и асинхронный.	Параллельные вычисления на матрицах (синхронизированные).	Массовый и асинхронный (как в биологии).
Физическая реализация	Живой организм (биологическая ткань).	Кремниевые чипы (CPU, GPU, TPU).	Специализированные нейроморфные чипы.

Б. Слои

Нейросеть состоит из нескольких слоёв нейронов:

– входного,

– скрытого,

– выходного.

Входной слой получает данные, скрытый слой обрабатывает их, а выходной слой выдаёт результат. Количество слоёв и нейронов в каждом слое может варьироваться в зависимости от задачи.

Исходя из разных «форм существования» нейронов, которые мы обсудили, давайте посмотрим, как выглядят слои в каждом случае.

1. Слои в программной реализации

Нейрон – это виртуальная сущность, «рецепт» в виде данных (веса) и кода (формула), который исполняется процессором. Можем представить фабрику, где нет физических станков, а есть один универсальный робот-манипулятор (CPU/GPU) и склад с инструкциями для него (оперативная память). Слой в программной реализации (самый частый случай) – это группа инструкций, которые робот выполняет последовательно для каждого «рабочего» (нейрона).

Входной слой – робот берет сырье (входные данные, например, пиксели изображения) и просто записывает его в таблицу. Здесь почти нет «работы».

Скрытый слой – робот берет первую инструкцию (нейрон №1): «Возьми данные из таблицы, перемножь на вот эти числа (веса), сложи, примени правило (функцию активации) и запиши результат в новую таблицу. Потом он берет инструкцию для нейрона №2, №3 и так для всех 1000 и более нейронов в этом слое.

Выходной слой – робот выполняет последний набор инструкций. Результат в итоговой таблице – это ответ сети (например, «это кошка»).

Слои здесь – логические этапы обработки. На центральном процессоре (CPU) вычисления для нейронов одного слоя обычно происходят почти последовательно. Даже если у CPU несколько ядер («роботов»), каждое ядро все равно обрабатывает свои нейроны по очереди. Их число намного меньше, чем в графических процессорах (GPU), поэтому параллелизм ограничен. На параллельных архитектурах, таких как графические процессоры (GPU), один «робот-менеджер» может отдавать команды тысячам маленьких рабочих - ядер, которые вычисляют выходы многих нейронов одновременно.

Несмотря на эту параллельность, нейроны всё ещё остаются виртуальными сущностями, описываемыми кодом и данными в памяти.

2. Слои в аппаратных ускорителях (TPU, NPU)

«Сущность» нейрона начинает обретать физическую форму. Это не просто инструкция, а специально спроектированный блок транзисторов на кристалле чипа, заточенный под конкретные вычисления. Представим специализированную автоматизированную фабрику, в которой имеется конвейерная лента и настоящие, физические станки, стоящие друг за другом. Слой в аппаратных ускорителях можно представить, как физический участок конвейера со своей группой станков.

Входной слой – участок, где сырье загружается на ленту.

Скрытый слой – участок, где стоит ряд одинаковых станков (матричные умножители). Каждый станок выполняет работу сразу целой группы «нейронов». Данные движутся по конвейеру, и каждый следующий слой-участок получает результат работы предыдущего, чтобы обработать его глубже.

Слои здесь – это физические модули чипа, соединенные друг с другом. Обработка данных происходит буквально «на лету», пока информация передается от одного физического блока к другому. Это невероятно быстро и эффективно.

3. Слои в нейроморфных чипах

Нейрон в данном случае выглядит почти «живым» аналогом. Его роль играет физический компонент (например, мемристор), чьи электронные свойства (сопротивление) имитируют (воспроизводят) поведение биологического синапса. Это уже не фабрика, а нервная ткань, выращенная в лаборатории. Это уже не набор станков, а почти аналог мозга.

Слой в нейроморфном чипе – это физически выделенная группа компонентов, соединённая между собой проводниками в определённой архитектуре.

Сигнал (электрический импульс) приходит на один «слой» компонентов, вызывает в них изменения (нейроны возбуждаются), результат этого возбуждения передается по проводникам на следующий «слой» компонентов. Слои в нейроморфных чипах наиболее близки к биологическим. Это физическая структура, где функция обработки информации неотделима

от самого материала и его расположения. Информация обрабатывается не цифровым вычислением, а прямым протеканием и преобразованием сигнала через физическую среду.

Спайковые нейросети, для которых нейроморфные чипы являются идеальной средой, привносят принципиально новое измерение в понятие слоев – время. В отличие от традиционных сетей, где информация обрабатывается слой за слоем в строго последовательном порядке, спайковые сети могут обрабатывать информацию асинхронно. Входной слой кодирует аналоговые данные в последовательности спайков (spike trains).

Существует несколько способов кодирования:

– *частотное кодирование* – интенсивность входного сигнала определяет частоту спайков;

– *временно́е кодирование* – информация содержится в точных моментах времени спайков;

– *популяционное кодирование* – группы нейронов, которые совместно представляют информацию.

Скрытые слои состоят из спайковых нейронов, которые интегрируют входящие импульсы во времени. Важная особенность – нейрон в слое может активироваться не одновременно со всеми остальными нейронами своего слоя, а только при достижении порога активации. Это создает разреженную активность – в каждый момент времени активна лишь небольшая часть сети, что радикально снижает энергопотребление.

Выходной слой декодирует паттерны спайков в финальный результат, используя различные методы интерпретации временны́х последовательностей импульсов.

Преимущество такой архитектуры заключается в том, что обработка информации происходит в режиме реального времени – сеть может реагировать на входные данные по мере их поступления, не дожидаясь обработки всего набора данных.

Концепция слоев едина для всех видов реализации нейросетей – это иерархические этапы обработки информации, где каждый этап извлекает все более сложные признаки, но их физическое воплощение принципиально разное: от виртуальной последовательности команд до настоящего «конвейера» (классический AI-ускоритель) или даже «органа» (нейроморфный чип) на кристалле кремния (Таблица 6).

Таблица 6. «Формы слоев»

Тип реализации (реализация для каждой формы существования нейронов)	Что из себя представляет слой для данной формы нейронов	Как взаимодействуют нейроны в слое?
Программная	Логический этап алгоритма. Последовательность команд для процессора.	В основном последовательно (на CPU) или малыми группами (на GPU). Процессор вычисляет выход каждого нейрона по одному.
Аппаратные ускорители (TPU/NPU)	Физический модуль на кристалле чипа. Блок, сконфигурированный для параллельных операций.	Физически и параллельно. Данные проходят через слой и обрабатываются многими «нейронами» (транзисторами) одновременно.
Нейроморфная	Физическая группа компонентов (например, мемристоров), соединенная проводниками.	Аналогово и «естественно», как в мозге – через распространение импульсов и изменение свойств самих компонентов.

B. Связи

Нейроны в разных слоях связаны между собой. Каждая связь имеет вес, который определяет важность связи. Веса корректируются в процессе обучения, чтобы минимизировать ошибку и улучшить точность предсказаний.

Пример: в задаче классификации изображений нейросеть может иметь входной слой, который получает пиксели изображения, несколько скрытых слоёв, которые извлекают признаки, и выходной слой, который выдаёт класс изображения.

1. Связи в программной реализации

Нейроны можно представить, как виртуальные инструкции (данные + код), слои – как логические этапы обработки, а связи – это адреса в памяти (указатели) и значения весов.

Представим, что каждый «рабочий» (нейрон) в цехе «Слой №2» имеет список всех «рабочих» из цеха «Слой №1». Для каждого «рабочего» цеха «Слой №2» записано число – вес связи (Weight), который обозначает насколько сильно он прислушивается к мнению каждого из своих коллег из предыдущего цеха.

Когда процессор вычисляет результат работы нейрона в «Слое №2», он делает вот что:

1. смотрит в свой «список контактов рабочих» из «Слоя №1»,
2. для каждого контакта берет итог его работы (выходное число), которое лежит в определенной ячейке оперативной памяти,
3. умножает это число на «коэффициент важности» (вес) для этого контакта,
4. складывает все такие произведения.

Главное, связи здесь – это не провода, а просто числа (веса), хранящиеся в памяти компьютера.

Когда процессор считает результат работы нейрона, он знает, какие именно числа из предыдущего слоя нужно взять и на какие веса их умножить – эта информация заложена в программе.

2. Связи в аппаратных ускорителях (TPU/NPU)

Нейроны в аппаратных ускорителях – это специализированные блоки транзисторов, слои – физические модули на кристалле чипа, а связи – это физические проводники (металлические дорожки) на кристалле кремния.

Вернемся к аналогии с конвейерной фабрикой. Выход одного цеха-слоя (например, матричного умножителя) – это не число в памяти, а напряжение на выходных контактах этого физического блока. Эти выходные контакты напрямую, физически соединены металлическими проводниками со входными контактами следующего цеха-слоя. «Вес» связи – это уже не просто число, а физическая настройка самого вычислительного блока, которая определяет, как он преобразует входящий ток/напряжение.

Главное – связи здесь материальны, это физические проводники.

Но чип – это не статичная схема с одним набором весов, а универсальный и переконфигурируемый вычислитель, который перед работой «запоминает» нужные настройки, загружая их в свою сверхбыструю локальную память. «Вес» связи определяется данными, которые загружаются в сконфигурированные вычислительные блоки. Данные передаются не через оперативную память, а напрямую «с выхода одного модуля на вход

следующего» по этим микроскопическим «проводам». Это обеспечивает колоссальную скорость, так как исключаются задержки на обращение к памяти.

Чтобы запустить другую нейросеть, вы просто загружаете в чип новый набор весов из внешней памяти. Процесс загрузки управляется драйверами и программным обеспечением, которые копируют данные из оперативной памяти сервера во внутреннюю память чипа.

3. Связи в нейроморфных чипах

Нейроны здесь – это физические компоненты (мемристоры и др.), слои – это группы таких компонентов, а связи – это сама среда, изменяющая свои свойства, т.е. аналог биологических синапсов.

Это самый интересный и сложный для понимания случай. Здесь связь – это не провод и не число, а нечто большее. Мемристор – это компонент, который запоминает, какой ток через него проходил. Его сопротивление меняется в зависимости от этого. Чем чаще через него проходил сигнал, тем «сильнее» становится связь (как синапс в мозге укрепляется при обучении).

Таким образом, сам материал соединения (мемристор) и является связью. Его физическое свойство (сопротивление) – это и есть аналог веса (Weight).

В этом мире связь – это неотъемлемое свойство материала. Информация передается и обрабатывается не как цифровые данные, а как аналоговый электрический импульс, который напрямую изменяет силу самой связи (сопротивление мемристора). Это максимально близко к тому, как учатся биологические сети: сам процесс передачи информации одновременно является и процессом обучения, изменяющим структуру сети.

В спайковых нейросетях на нейроморфных чипах связи между нейронами реализуют принципиально иной механизм обучения – ранее рассмотренный нами Spike-Timing-Dependent Plasticity (STDP). Ранее мы его уже рассматривали. Это правило обучения основано на относительном времени прихода спайков от пре- и постсинаптических нейронов и имитирует хеббовский принцип обучения: «нейроны, которые активируются вместе, соединяются вместе», но с учетом точного времени.

Преимущества STDP:

точность и устойчивость – изменение синаптического веса происходит на основе строго локальной информации, т.е. решения принимаются на основе наиболее релевантных данных в конкретном контексте, что делает процесс надежным и независимым;

ресурсоэффективность – обучение происходит в реальном времени без необходимости в размеченных данных и глобальных вычислениях, т.е. экономятся вычислительные ресурсы, память и время;

энергоэффективность – обучение активируется только в момент поступления спайков, что радикально снижает энергопотребление.

В отличие от традиционного алгоритма обратного распространения ошибки, STDP не требует глобального вычисления градиентов и может работать полностью локально в каждом синапсе. Это делает его формой неуправляемого обучения, основанного исключительно на локальных событиях, что является концептуально иным подходом по сравнению с глобальным методом обратного распространения ошибки. Для нейроморфного оборудования, где локальность критически важна, спайковые сети особенно привлекательны. Однако STDP также создает вызовы: точная настройка временных окон и параметров пластичности требует тщательной калибровки для достижения стабильного обучения без катастрофического забывания.

Независимо от реализации, смысл связей один – они определяют, насколько сильно сигнал от одного нейрона влияет на работу другого (Таблица 7). Сила этого влияния (вес) – это и есть память нейросети, ее знание, извлеченное из данных в процессе обучения.

Просто в разных формах существования нейронов это «знание» хранится по-разному: в виде таблицы чисел, в виде архитектуры кремниевого чипа или в виде физических свойств аналоговых материалов.

Таблица 7. «Формы связей»

Реализация	Связь	Чем представлен «вес» связи?
Программная	Логическая ссылка на ячейку в памяти + число.	Просто число с плавающей запятой (float), которое является стандартным способом представления дробных чисел в компьютере и позволяет точно кодировать тонкие градиенты важности связей.
Аппаратная (TPU /NPU)	Физический проводник на кристалле, соединяющий два модуля.	Настройки, загружаемые в сконфигурированные вычислительные блоки.
Нейроморфная	Физический компонент (например, мемристор), который и есть связь.	Физическое свойство этого компонента (например, электрическое сопротивление).

§ 5. Принципы работы нейросетей

Как мы уже выяснили, нейросеть состоит из нейронов, организованных в слои и соединённых связями с весами.

Процесс работы нейросети, независимо от её архитектуры, можно разделить на два ключевых режима: обучение (Training) и инференс (Inference) – применение обученной модели для получения результата. Здесь мы сфокусируемся на основном цикле обработки данных.

Для общего понимания можно привести некоторые аналогии из повседневной жизни, хотя они безусловно существенно упрощают реальное положение дел. Представьте, что вы собрали робота-садовода и «вложили» в него книгу о растениях. Сначала он будет путать кактус с крапивой, но, если показывать ему тысячи фото, со временем он научится отличать розу от одуванчика, а потом даже предсказывать, когда зацветет яблоня. Нейросети работают так же – это алгоритмы, которые учатся решать задачи на примерах.

Три этапа работы нейросети при инференсе (Схема 2):

1. **Препроцессинг входных данных**, т.е. получение сырых данных (пиксели изображения, слова текста, показания датчиков) и их преобразование в числовой тензор (многомерный числовой массив).

2. **Прямой проход /прямое распространение** (форвард-пасс, англ. Feedforward Pass) – многократное и последовательное применение к данным двух операций в каждом слое:

Обработка (линейное преобразование) – вычисление взвешенной суммы.

Активация (нелинейное преобразование) – применение функции активации, причём результат её работы передаётся на вход следующему слою.

3. **Формирование результата (выход)** – финальный выходной слой выдаёт результат на выходном слое. В зависимости от задачи, это может быть вектор вероятностей (например, «на 95% это кошка, на 5% – собака»), числовой прогноз или сгенерированный текст.

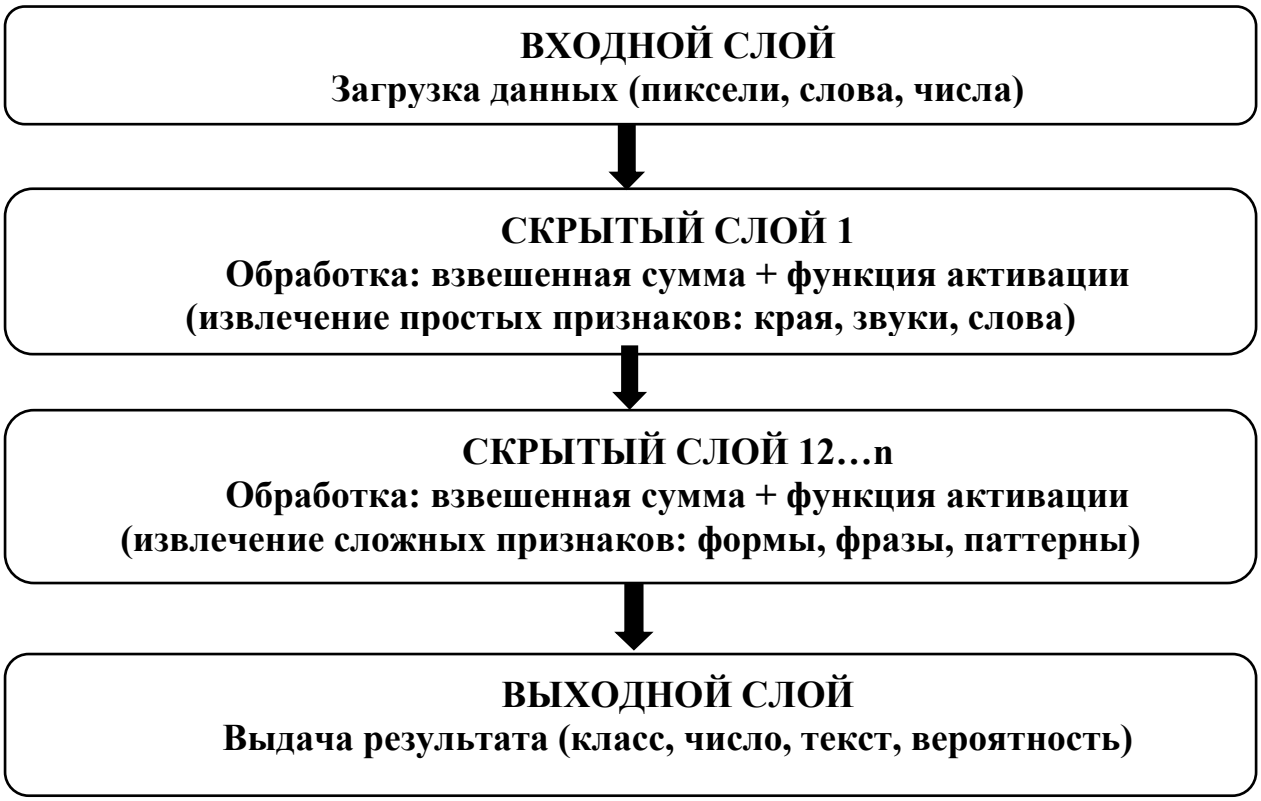
Нейросеть обрабатывает данные поэтапно, как конвейер. Каждый слой извлекает все более сложные признаки, пока не получит итоговый ответ. Но чтобы сеть научилась делать это правильно, требуется этап сложной и ресурсоёмкой подготовки – обучения. Об этом мы поговорим дальше.

Нейросеть можно сравнить с разными системами. Например, с фильтром для кофе, где данные проходят через слои «нейронов», как вода через молотые зерна, и на выходе получается четкий ответ. Или – с детским конструктором,

в котором каждый слой – это новый уровень сложности, т.е. сначала алгоритм учится видеть линии, потом формы, а затем и целые объекты. Или с оркестром, в котором нейроны «играют» вместе, создавая гармонию из хаоса данных, при этом дирижёр – это алгоритм обучения, который настраивает каждого музыканта (вес связи).

Архитектурные типы нейронных сетей представляют собой различные способы организации и соединения искусственных нейронов, каждый из которых оптимизирован для решения специфических задач.

Схема 2. «Как работает нейросеть: прямое распространение»



§ 6. Архитектурные типы нейросетей

В мире нейросетей существует множество архитектурных типов, каждый из которых имеет свои уникальные особенности и сценарии применения. Эти архитектуры разрабатывались с целью решения специфических задач, что позволило искусственному интеллекту достигать значительных результатов в различных областях. Некоторые архитектуры идеально заточены под определенный тип данных. Например, для распознавания изображений есть сверточные сети (CNN), которые «видят» картинки гораздо лучше других. Правильно подобранная архитектура справится с задачей точнее и быстрее. Это как поехать в другой город на скоростном поезде, а не на велосипеде.

При прочих равных условиях одни архитектуры учатся быстрее, другие – медленнее. Выбор архитектуры — это один из ключевых способов найти баланс между вычислительной эффективностью и точностью модели. Текст, видео, звук, числа – все это разные форматы информации. Универсального искусственного «мозга» для всего пока не существует, поэтому и архитектуры такие разные. Существуют гибридные и мультимодальные архитектуры, но они все равно специализируются на работе с несколькими конкретными типами данных, а не со всеми сразу.

Классифицировать архитектуры можно по разным признакам, но самый главный – то, как в них соединены нейроны (какая «проводка» в «мозге»). Ниже мы рассмотрим основные типы более подробно, а их краткое сравнение для удобства представлено в конце раздела в Таблице 8.

Перцептроны и сети прямого распространения (FNNs)

Перцептроны и сети прямого распространения (Feedforward Neural Networks, FNN) – это фундаментальные архитектуры, лежащие в основе современного глубокого обучения. Мы уже упоминали перцептроны – это детище Фрэнка Розенблатта в главе про этапы в развитии нейросетей. Ключевая сила сетей прямого распространения заключается в том, что они могут научиться вычислять практически любую сложную взаимосвязь между входными и выходными данными, если им предоставить достаточно ресурсов. Проще говоря, они являются универсальными инструментами для выявления закономерностей.

Перцептрон – простейшая модель нейрона. Он принимает входные сигналы, взвешивает их, суммирует и пропускает через функцию активации, решая, «сработать» или нет. Один перцептрон способен решать лишь простейшие, линейно разделимые задачи.

Для решения сложных нелинейных задач перцептроны объединяют в сети прямого распространения (FNNs), где нейроны организованы в последовательные слои. Информация в них течёт строго в одном направлении – от входа к выходу. Это самая базовая и исторически первая архитектура нейронных сетей, которая задала основной принцип работы многих последующих, более сложных моделей.

В качестве метафоры всю FNN в целом можно представить как многоуровневую производственную линию, где данные последовательно

обрабатываются, проходя этап за этапом, пока не будет получен готовый результат.

Таким образом, FNN – это общая архитектурная парадигма, «скелет», который можно наполнять разными типами нейронных слоев для решения конкретных задач.

Хотя «чистые» FNN редко используются для сложных данных (изображений, текста), они являются ключевым компонентом гибридных архитектур. Например, сверточная сеть (CNN) извлекает из изображения черты, а несколько полносвязных слоев (Dense) в конце на основе этих признаков ставят итоговый «диагноз» (относят к определенному классу).

Полносвязные сети (Fully Connected Networks)

Полносвязная сеть – это архитектура, целиком состоящая из полносвязных слоев (Dense). Именно эти слои и являются «универсальными солдатами» нейросетей. В таком слое каждый нейрон соединен с каждым нейроном на следующем уровне, образуя очень плотную сеть. Полносвязная сеть – это простейшая и самая распространенная реализация принципа FNN (сети прямого распространения).

Полносвязные сети хороши для простых задач, где данные – это просто столбики чисел или представлены в виде таблицы (например, предсказание цены дома по его параметрам – площадь, этаж, или список признаков объекта – рост, вес, возраст, доход и т.д.). На таких данных они решают задачи классификации (например, «кредитоспособен / не кредитоспособен») и регрессии (например, «предсказать цену»). Стоит отметить, что для табличных данных FFN, оставаясь классическим и проверенным подходом, сегодня часто конкурируют с другими методами, например, с градиентным бустингом (это отдельный иной подход к машинному обучению, который основывается на деревьях решений, а не на нейросетях).

В качестве метафоры, иллюстрирующей возможности полносвязных сетей, можно привести стандартную бюрократическую систему, где каждый должен согласовать документ с каждым, что эффективно для малого масштаба, но очень медленно и громоздко для сложных структур.

Работу FFN также можно представить, как гигантскую паутину, где каждая нить соединяет все точки данных. Например, если вы введете рост, вес и возраст, то обученная нейросеть предскажет размер обуви.

FFN простые, но универсальные нейросети, они служат основой для сложных моделей и редко используются напрямую для работы

со сложными данными, такими как изображения или текст, так как не видят структурных связей между частями. Однако FFN могут быть финальным звеном в более сложной архитектуре. Они применяются для распознавание рукописного ввода, классификации текстов по темам или настроению. Примеры практического применения – кредитный скоринг или автоматизированная система оценки платёжеспособности и надёжности заёмщика (Сбербанк, Тинькофф); предсказание спроса на энергию (Россети); персонализированный контент (виртуальные ассистенты – «Салют» от «Сбербанка», Алиса от «Яндекса», «Маруся» от VK).

Сверточные нейронные сети (CNN)

Сверточные нейронные сети (Convolutional Neural Networks, CNN) – это архитектура, специально разработанная для работы с изображениями, видео и другими данными, имеющими пространственную структуру. Они широко используются в задачах компьютерного зрения, по праву считаясь его «экспертами».

Основной математический прием, лежащий в основе CNN, – это операция свертки[39].

В сверточных слоях CNN нейроны соединены не со всеми, а только с ограниченной областью соседних нейронов предыдущего слоя. Это позволяет им извлекать локальные признаки. Кроме того, используется слой пулинга (pooling) для уменьшения пространственных размерностей изображения, сохраняя при этом важнейшие признаки. После того как сверточный слой нашел черты (например, края), слой пулинга как бы сжимает изображение, оставляя только самую важную информацию. Это как сделать суммаризацию статьи – убрать лишние слова, оставив суть, и ускорить дальнейшую работу. Эти специализированные структуры делают CNN чрезвычайно эффективными для работы с изображениями.

CNN обрабатывают изображение, которое для компьютера представляет собой просто таблицу чисел (пикселей), с помощью обучаемых фильтров (ядер свертки). Ключевое отличие CNN от классических алгоритмов компьютерного зрения заключается в том, что эти фильтры не задаются инженером вручную, а являются «умными шаблонами», которые сеть сама создает и настраивает в процессе обучения под конкретную задачу и данные. Каждый такой фильтр,

[39] Операция называется «сверткой», потому что это устоявшийся математический термин, описывающий процесс переворота одного набора данных и его последующего «прокатывания» и «смешивания» с другим набором данных. В контексте CNN это название — дань уважения математическому происхождению операции, хотя визуально оно больше напоминает «сканирование» изображения фильтром.

применяясь к исходным данным, выполняет операцию свертки для выделения конкретных особенностей. На первых слоях эти операции позволяют обнаруживать простые элементы: края, углы, цветовые пятна. На последующих слоях комбинация этих простых признаков позволяет сети выявлять более сложные паттерны: из краев формируются контуры, а из контуров – сложные формы (например, глаз или колесо). Таким образом, CNN – это не жесткий алгоритм поиска по шаблону, а применение десятков или сотен самооптимизирующихся шаблонов, которые настраиваются для нахождения именно тех паттернов, которые ведут к правильному ответу.

В итоге на выходе сети формируется иерархическое представление данных – комбинация сложных признаков, которая с высокой вероятностью соответствует определенному объекту (например, «кошка»). Эта архитектура идеально подходит для иерархической природы визуальных данных, где понимание строится от простых элементов к сложным объектам. Можно сказать, что CNN «видят» мир как иерархию паттернов, собранных через последовательность сверток. Например, чтобы отличить кошку от собаки, CNN строит иерархию признаков, где на верхних уровнях формируются высокоуровневые концепты, по статистике соответствующие характерным чертам этих животных, т.е. сначала выделит характерные формы ушей и хвоста, а затем проанализирует их композицию. CNN можно представить как художника, который изучает картину через лупу разного увеличения: сначала он фиксирует отдельные мазки и края (свертки нижних уровней), а затем отступает, чтобы увидеть, как эти детали складываются в целостный образ.

CNN демонстрируют выдающиеся результаты в задачах классификации изображений, распознавания объектов и семантической сегментации. Например, они применяются в здравоохранении для анализа медицинских изображений и диагностики заболеваний, таких как рак, например.

Примеры практического применения – распознавание лиц в соцсетях (ВКонтакте, Telegram); медицинская диагностика, например, анализ рентгеновских снимков (СберЗдоровье); автономные автомобили – обнаружение пешеходов и дорожных знаков (Яндекс.Авто).

Рекуррентные нейросети (RNN)

Рекуррентные нейросети (Recurrent Neural Networks, RNN) разрабатывались для обработки последовательных данных. Их ключевая

особенность – наличие обратных связей, что позволяет учитывать предыдущий контекст при обработке каждого нового элемента последовательности.

То есть RNN – это «специалисты по последовательностям» с «памятью»: они обрабатывают информацию не всю сразу, а по частям (по словам в предложении, по кадрам в видео), и при этом помнят, что было перед этим. Это удобно для всего, что имеет порядок и время, например, машинный перевод, генерация текста, чат-боты, анализ тональности отзывов, прогнозирование временных рядов (например, биржевых котировок или погоды). Это как при чтении книги – вы понимаете смысл предложения, только помня предыдущие. RNN – это рассказчик, который помнит, о чём говорил минуту назад.

Специальным видом RNN, предназначенным для борьбы с главным недостатком простых сетей – «забыванием» информации из начала длинных последовательностей, – являются сети с механизмом «Долгой краткосрочной памяти» (Long Short-Term Memory, LSTM). Этот механизм позволяет сделать кратковременную память долгой, то есть избирательно сохранять важную информацию на сотни и тысячи шагов вглубь последовательности (что безусловно является некоторым упрощением, т.к. на практике длина контекста, который LSTM может эффективно запомнить, все же имеет пределы и зависит от многих факторов). В качестве иллюстрации представьте, что обычная RNN – это студент, который готовится к экзамену в последнюю ночь и помнит только то, что прочитал последние пару часов. LSTM – это организованный студент с системой флеш-карточек и блокнотом, который избирательно записывает самое важное (например, сложные формулы) в блокнот (долгосрочное хранение), чтобы обращаться к ним в нужный момент, параллельно держа в уме текущий контекст.

Примеры практического применения – голосовые помощники (Алиса, Маруся) – понимание последовательности речи; машинный перевод (Yandex Translate, DeepL) – учет контекста всего предложения для точности; прогнозирование спроса (Wildberries, Ozon) – анализ временных рядов покупок.

Таким образом, слабое место простых RNN – склонность «забывать» длинные контексты – было успешно преодолено в архитектуре LSTM, что и обеспечило ей широкое практическое применение, хотя сегодня их все активнее теснят Трансформеры.

Трансформеры (Transformers)

Трансформеры решают ту же задачу, что и RNN, а именно обработку последовательных данных (текст, речь, временные ряды), но используют механизм внимания. Ключевое отличие в том, что Трансформеры обрабатывают данные параллельно, а не последовательно. Это как иметь возможность мгновенно пролистать и проанализировать сразу всю главу книги, а не читать ее строка за строкой (конечно, у модели есть техническое ограничение на длину «книги» — размер контекстного окна, но об этом мы поговорим далее в разделе про промптинг и токены). Это позволяет нейросети вычислить, насколько каждое слово в предложении важно для понимания каждого другого слова. Это называется самовниманием – механизмом, который взвешивает значимость всех слов в предложении одновременно. Например, для правильного перевода слова «его» в конце предложения, модель может «посмотреть» на слово «кошка» в начале и понять, что речь о животном мужского пола. Это похоже на чтение текста не линейно, т.е. пробежать его глазами, выделить ключевые слова и мгновенно уловить связь между ними, даже если они далеко друг от друга. Эту архитектуру используют современные чат-боты (как ChatGPT). Это прорывная технология, которая сейчас доминирует в обработке языка. Можно сказать, что Трансформеры сделали RNN менее популярными для большинства сложных задач обработки естественного языка, потому что преодолевают их главные проблемы и решают такие задачи гораздо эффективнее. Трансформеры анализируют данные с учетом контекста, особенно полезны для текста и мультимодальных задач (текст + изображение). Представим, что Трансформеры – это команда детективов. Каждый смотрит на разные части задачи одновременно и находит связи между ними.

Примеры практического применения – генерация текстов (YandexGPT, GigaChat); поисковики – понимание смысла запроса (Яндекс.Поиск); создание изображений по описанию (Kandinsky). Сервис Салют (Сбер) использует Трансформеры, чтобы понимать голосовые команды вроде «Найди фильм, где герой путешествует во времени» – даже если вы сказали это с акцентом или в шумной комнате.

При всей мощи Трансформеров, важно понимать их принципиальные ограничения. Эти модели генерируют текст на основе статистических закономерностей, а не реального понимания. Они не обладают внутренней моделью мира, не могут верифицировать факты и склонны к «галлюцинациям» – уверенной генерации убедительно звучащей, но фактически неверной

информации. Исследования показывают, что частота таких ошибок может достигать 15-30% при генерации фактологических утверждений, причем модель демонстрирует высокую уверенность даже в неверных ответах. Это делает системы на основе Трансформеров ненадежными для критически важных задач без тщательной человеческой проверки.

Генеративно-состязательные сети (GAN)

Генеративно-состязательные сети (Generative Adversarial Networks, GAN) – представляют собой систему из двух нейросетей: генератора, который создает новые изображения (или другие данные), и дискриминатора, который оценивает, насколько эти изображения реалистичны. Обе сети обучаются одновременно, что создает состязательный процесс, в результате которого генератор начинает производить более качественные и правдоподобные данные. GAN хороши для генерации реалистичных изображений, лиц, картин по текстовому описанию, повышения разрешения фото и создания анимаций. Эта архитектура стала прорывом, открыв эру генерации данных, неотличимых от реальных. Это как соревнование между художником (генератор) и экспертом (дискриминатор): художник рисует копию картины, эксперт пытается её распознать, найдя отличия от подлинника, художник вносит исправления – и так до тех пор, пока эксперт уже не может различить копию и подлинник.

Хотя GAN были пионерами в генерации реалистичных изображений и лиц и до сих пор используются, их сложное и неустойчивое обучение привело к тому, что сегодня для генерации картинок по текстовому описанию (как в Midjourney, DALL-E или Kandinsky) чаще применяются диффузионные модели. Они проще в обучении, реже «ломаются» и дают более детализированный результат.

Диффузионные модели (Diffusion Models)

Принцип работы диффузионных моделей – это не состязание, а постепенное «очищение» изображения от шума: модель учится шаг за шагом превращать случайный набор пикселей в четкую картину, соответствующую запросу. Это позволяет добиваться высочайшего качества и детализации.

Работу диффузионных моделей можно сравнить с работой реставратора:

«разрушение» (Forward Process), когда модель берет исходное изображение и постепенно, шаг за шагом, добавляет в него «шум» – случайные помехи, пока картинка не превращается в совершенно случайное месиво

из пикселей. Это похоже на то, как если бы картина медленно покрывалась слоями пыли и грязи, пока не стала бы полностью однородной и серой;

«восстановление» (Reverse Process) – потом модель учится разворачивать этот процесс в обратную сторону, её тренируют смотреть на зашумлённое изображение и предсказывать, как бы выглядел этот же кадр, но с чуть меньшим количеством шума, т.е. она учится «убирать пыль» с картины.

На практике это выглядит следующим образом. Когда вы даёте модели текстовый запрос «кошка в космосе в стиле поп-арт», она начинает со случайного шума и последовательно, через десятки шагов, «убирает» этот шум, руководствуясь вашим описанием. На каждом шаге изображение становится чуть четче и ближе к желаемому результату. Именно поэтому генерация картинок в таких сервисах занимает несколько секунд – модель совершает множество последовательных шагов «очистки».

Диффузионные модели часто превосходят GAN в качестве и детализации генерируемых изображений, а также считаются более стабильными в обучении. Если GAN – это соревнование двух сетей, то диффузионная модель – это внимательный реставратор, который кропотливо создаёт шедевр из хаоса.

Мультимодальные нейросети

Одним из самых захватывающих направлений развития современных нейросетей стали мультимодальные системы – модели, способные одновременно понимать и обрабатывать несколько типов данных: текст, изображения, звук и даже видео. Это существенный шаг вперед по сравнению с узкоспециализированными нейросетями, фокусирующимися только на одном типе информации. Мультимодальные модели в некотором смысле имитируют человеческое восприятие мира, когда мы естественным образом интегрируем то, что видим, слышим и читаем – наш мозг создает целостную картину реальности из разрозненных сигналов. Например, услышав шум за спиной, мы оборачиваемся, видим человека и интерпретируем ситуацию, объединяя звуковую и визуальную информацию. В 2023-2024 годах мультимодальность стала одним из доминирующих трендов в разработке крупных нейросетевых систем. Модели нового поколения, такие как GPT-4V (Vision), Claude Opus и Gemini Ultra, умеют одновременно анализировать изображения и текст, создавая целостное понимание контекста.

Российские разработки не отстают: GigaChat от Сбера и мультимодальные возможности экосистемы Яндекса, включающие

YandexGPT и другие нейросетевые продукты, также развиваются в направлении интеграции различных типов данных.

Мультимодальные модели находят применение в разных сферах, например:

в медицинской диагностике (нейросеть анализирует одновременно медицинские снимки, истории болезни в текстовом виде и даже аудиозаписи кашля или сердечных тонов, что дает более комплексную картину состояния пациента),

в умных ассистентах (мультимодальная модель распознает продукты на фото полок холодильника и предложит рецепты, учитывая визуальную и текстовую информацию),

в образовании (на основе загруженных текстов, изображений и видео нейросеть создаст интерактивные обучающие материалы, объединяющие все эти данные),

в сфере безопасности (системы видеонаблюдения, способные видеть и слышать, могут более точно определять потенциально опасные ситуации).

Архитектурно мультимодальные системы представляют собой комбинацию специализированных компонентов. Например, для обработки изображений используются CNN или Vision Transformers, для текста – языковые модели на основе Трансформеров, а для звука – специализированные аудиомодели. Затем все эти компоненты объединяются через специальные «связующие» слои, которые учатся соотносить информацию из разных источников.

Создание по-настоящему интегрированных мультимодальных систем – сложная задача. Разные типы данных имеют разную структуру, масштаб и информационную плотность. Например, одно изображение может содержать информацию, эквивалентную тысячам слов. Современные модели решают эту проблему через создание единого «пространства представлений», где информация разных типов может быть закодирована и сопоставлена. Представьте, что у нас есть словарь, где каждому английскому слову соответствует русское. Здесь «пространство представлений» – это, как если бы мы переводили и английские слова, и картинки, и звуки на один универсальный «космический» язык, на котором все эти данные могут «общаться» между собой.

Будущее мультимодальных моделей видится в еще большей интеграции.

Для чего полезно иметь представление об архитектурных типах нейросетей

Архитектурные типы нейронных сетей важны, поскольку служат основой для разнообразных приложений и сценариев, охватывающих не только технические задачи, но и высоко творческие процессы. Например, CNN незаменимы для всего, что связано с картинками и видео, успешно используют для анализа сателлитных снимков и мониторинга окружающей среды, RNN подходят для задач, где важен порядок (речь, музыка), Трансформеры – лидеры в обработке текста и сложных взаимосвязей, полносвязные сети – простой и универсальный строительный блок, который можно найти внутри многих моделей, GAN нашли применение в том числе в разработке игр и создании виртуального контента (Таблица 8).

Нейросети всё чаще учатся на нескольких типах данных одновременно (мультимодальное обучение), что приближает нас к созданию более универсального и надежного искусственного интеллекта.

Каждая из перечисленных архитектур имеет свои сильные и слабые стороны. Выбор той или иной нейросети зависит от конкретной задачи, источника данных и желаемых результатов. Понимание различных архитектур и их возможностей – это ключ к успешному изучению нейросетей и их применению в реальных проектах.

Следует иметь в виду, что только на фундаментальном уровне одна модель – одна архитектура. То есть отдельно взятая, конкретная нейросетевая модель обычно строится на основе одной ключевой архитектуры. Например, модель GPT от OpenAI – это чистый Трансформер (точнее, его часть – декодер). Модель LSTM для прогнозирования – это чистая RNN. На системном уровне сложные продукты – это всегда гибрид. Когда мы говорим о готовом продукте, который мы используем (голосовой помощник, чат-бот, приложение для рисования), то практически не существует продуктов, которые используют всего одну архитектуру нейросетей. Это объясняется просто. Эффективность и специализация! Зачем заставлять одну архитектуру делать «как получится», если можно поручить каждую задачу тому, кто справится с ней лучше всего? Как мы уже обсудили, разные архитектуры решают разные задачи оптимальным способом. Можно, конечно, представить себе узкоспециализированное приложение, которое делает что-то одно. Например, модель для предсказания курса акций, которая построена только на основе RNN. Но даже её, скорее всего, будут комбинировать с другими алгоритмами для получения конечного результата.

Вам не нужно быть инженером или учёным, чтобы управлять автомобилем. Но понимание разницы между бензиновым двигателем, электромотором и коробкой передач делает вас более уверенным и осознанным водителем. Вы начинаете лучше чувствовать машину, эффективнее использовать её возможности и понимаете, чего от неё можно ждать, а чего – нет. Точно так же обстоит дело с искусственным интеллектом. Нейросети всё глубже проникают в нашу повседневную жизнь: от поиска в Интернете и рекомендаций фильмов до голосовых помощников и обработки фотографий.

Общее представление об архитектурах снимает страх перед «магией», помогает избежать разочарований от несбывшихся ожиданий от нейросети, дает ключ к эффективному общению.

ИИ перестаёт быть чёрным ящиком, который «как-то сам всё делает». Когда вы знаете, что у Алисы есть «память» (RNN) для ведения диалога, а нейросеть для генерации картинок (Diffusion) работает как реставратор, очищающий изображение от шума, – технологии становятся понятнее и ближе. Вы не просто нажимаете на кнопки, вы примерно представляете, что происходит «под капотом». Вы не будете просить ChatGPT нарисовать вам картину, а Midjourney – перевести статью. Понимая, что каждая архитектура заточена под свою задачу, вы сможете точнее выбирать правильный инструмент и гораздо эффективнее с ним работать. Вы будете знать, что для анализа таблиц с числами есть свои модели, а для распознавания лиц – совсем другие.

Даже когда нейросети решают схожие задачи (например, генерация текста), их внутренняя архитектура и принципы работы накладывают фундаментальные ограничения и определяют сильные стороны. Ожидание одинаковых результатов от систем с разной «начинкой» – всё равно что требовать от гоночного болида и внедорожника одинаковой проходимости по бездорожью и скорости на треке.

Знание основ – это первый шаг к созданию хороших промптов (запросов). Вы интуитивно начнёте давать нейросети для изображений более детальные описания, а текстовому помощнику – обучающий контекст и чёткие роли. Вы не просто пользуетесь технологией, вы начинаете с ней сотрудничать.

Вы сможете критичнее оценивать новости об ИИ, понимать, как ваши данные используются для рекомендаций, и даже распознавать потенциальные риски (как deepfake, например). Вы переходите из роли пассивного потребителя в роль грамотного пользователя, который понимает основы происходящего вокруг.

Не нужно зубрить формулы и алгоритмы. Речь идёт именно об общем представлении – о знании, что существуют разные «породы» ИИ, каждая со своим характером и талантами. Это знание не делает вас программистом, но оно делает вас сильнее в мире, где ИИ становится таким же привычным инструментом, как смартфон или Интернет. Вы начинаете говорить с технологией на одном языке и использовать её не вслепую, а с пониманием. Это ваша карта в новую, стремительно наступающую реальность. И она стоит того, чтобы с ней ознакомиться.

Критическое понимание архитектур нейросетей также помогает осознать фундаментальные ограничения современных систем ИИ. Зная, что нейросеть – это, по сути, статистическая модель, обученная на конкретном наборе данных, вы будете более осторожны с ее результатами.

Для общего понимания можно привести некоторые аналогии из повседневной жизни, хотя они, безусловно, упрощают реальное положение дел. Представьте, что вы собрали робота-садовода и «вложили» в него книгу о растениях. Сначала он будет путать кактус с крапивой, но, если показывать ему тысячи фото, со временем он научится статистически угадывать розу от одуванчика. Нейросети работают так же – это алгоритмы, которые учатся распознавать шаблоны и корреляции в данных, чтобы решать задачи. Они не обладают сознанием или пониманием, как человек, а лишь эффективно вычисляют закономерности.

Сгенерированный нейросетью текст может звучать уверенно и убедительно, но содержать фактические ошибки. Изображение может выглядеть эстетично, но нарушать физические законы или анатомию. Чем лучше вы понимаете, как работает инструмент, тем меньше вероятность, что вы станете жертвой его ограничений или, что еще хуже, будете распространять созданную им дезинформацию.

Таблица 8. «Основные архитектурные типы нейросетей»

Архитектура / Аналогия	Главная суперсила / Ключевой механизм	Для чего используется? / Примеры применения
Полносвязная FFN / «Бюрократ», «Универсальный солдат»	Анализ изолированных признаков / Полные связи.	Скоринг, прогнозы по таблицам. Примеры: кредитный скоринг (Сбербанк, Тинькофф), анализ табличных данных.
Сверточная CNN / «Искусственный глаз»	Поиск паттернов в пространственных данных / Свертка, пулинг.	Классификация, обработка изображений Примеры: распознавание лиц, медицинская диагностика (анализ снимков в СберЗдоровье), автопилоты (Tesla, Яндекс.Авто).
Рекуррентная RNN / «Рассказчик с памятью»	Учет контекста в последовательностях / Рекуррентные связи, «ворота» LSTM.	Перевод, анализ временных рядов Примеры: машинный перевод (Google Translate), прогнозирование спроса (Ozon, Wildberries), чат-боты.
Трансформер / «Команда детективов»	Анализ глобального контекста / Механизм внимания.	Современные NLP-задачи, большие языковые модели (LLM). Примеры: поисковики (Яндекс.Поиск, Google), умные ассистенты (Алиса, ChatGPT и др.).
GAN / «Художник», «Эксперт»	Генерация реалистичных данных / Состязательное обучение.	Генерация изображений. Примеры: генерация лиц, создание изображений, аугментация данных.
Диффузионная модель / «Реставратор»	Генерация изображений высочайшего качества / Постепенное «очищение» шума.	Создание изображений по описанию Примеры: генерация изображений по тексту (Midjourney, Kandinsky), ретушь фото (нейро-инструменты в Photoshop), генерация видео.
Мультимодальные сети / попытка обрабатывать разные типы информации «как человек»	Понимание и одновременная обработка различных типов данных / Совместная обработка различных типов данных в едином пространстве представлений.	Задачи, требующие интеграции текста, изображений и звука. Примеры: GPT-4V, Gemini 1.5 Pro, Claude 3 Opus, GigaChat (Сбер), YandexGPT с визуальными возможностями.

§ 7. Обучение нейросетей: от данных к знаниям

Обучение нейронных сетей представляет собой фундаментальный и мощный процесс в сфере информационных технологий. Несмотря на кажущуюся сложность, его можно представить в виде последовательной схемы, основанной на принципах адаптации, коррекции и поиска оптимального решения.

Обучение нейросети – это ключевой процесс преобразования сырых данных в полезные знания. Если говорить технически, это автоматическая корректировка внутренних параметров нейросети (весов и смещений) на основе анализа данных. Представьте, что вы учите маленького ребенка отличать кошку от собаки. Вы показываете ему много картинок и говорите: «Смотри, это кошка, а это – собака». Ребенок постепенно учится замечать закономерности: у кошек обычно более округлая мордочка, а у собак – вытянутая и т.д. Нейросеть учится очень похоже. Однако, в отличие от ребенка, она не строит гипотез о мире, а путем поиска статистических закономерностей и корреляций в данных настраивает свои параметры для последующих прогнозов и решений. По сути, мы «кормим» компьютер данными, а он извлекает из них знания.

Процесс обучения любой нейронной сети начинается с двух критически важных предварительных шагов: четко определить цель обучения и подготовить качественные данные для ее достижения. Эти этапы являются фундаментом, на котором строится вся последующая работа системы.

А. Постановка задачи

Первый шаг – это формулирование (определение, постановка) задачи. Это может быть задача классификации, когда система должна отнести объект к одному из нескольких заранее известных классов, например, распознать, изображена ли на фотографии кошка или собака. Другой тип задачи – регрессия, где требуется предсказать непрерывное значение, например, цену квартиры в зависимости от ее площади и местоположения. Также существуют более сложные задачи генерации, когда сеть сама создает новый контент, такой как текст, музыка или изображения, или задачи кластеризации, которые относятся к обучению без учителя, когда сеть самостоятельно группирует данные по скрытым закономерностям без предварительных меток. Выбор

задачи определяет выбор архитектуры сети и метода обучения. Примеры из практики показывают широту применения нейросетей: от рекомендательных систем в Интернет-магазинах до управления беспилотными автомобилями.

Б. Сбор и подготовка данных

Второй, не менее важный шаг – это сбор и подготовка данных. Качество и количество данных напрямую влияют на способность нейросети обучиться и успешно решать поставленную задачу. Современные глубокие нейронные сети, как правило, требуют огромных объемов данных. Чем сложнее задача, тем больше данных нужно для выявления надежных закономерностей и предотвращения переобучения – ситуации, когда сеть просто «запоминает» примеры из обучающей выборки вместо того, чтобы выявлять общие принципы, и, как следствие, демонстрирует низкую эффективность на новых, ранее не виденных ею данных.

1. Сбор данных и публичные датасеты

Сбор требуемых для обучения нейросети массивов данных – сложная и дорогостоящая задача, разными способами, например, путем использования публичных датасетов или путем самостоятельного сбора данных под конкретную цель. Главное правило – данные должны быть разнообразными и представлять все возможные сценарии, с которыми обученная модель может столкнуться в дальнейшей работе. Например, для модели, распознающей лица, нужны фотографии людей разного возраста, пола, в очках и без очков, с разными прическами, при разном освещении и в разных ракурсах. Если показать ей только портреты при идеальном свете, то в реальных условиях она с высокой вероятностью растеряется. Таким образом, качество и объём данных непосредственно влияют на успех будущего обучения.

Для начала разберемся с публичными датасетами. Представьте, что вы хотите научить нейросеть распознавать кошек и собак на фото. Вместо того чтобы самому фотографировать тысячи животных, размечать фотографии и загружать их в компьютер – вы можете скачать уже готовый датасет, где всё это уже сделано за вас.

Публичный датасет – это предварительно собранный, размеченный и опубликованный в открытом доступе набор данных. Публичные датасеты экономят время и ресурсы – не нужно собирать данные с нуля; позволяют

учиться и экспериментировать – студенты, исследователи, компании тестируют на них свои алгоритмы; обеспечивают сопоставимость – если все используют один и тот же датасет, можно объективно сравнивать, чья модель работает лучше.

Примеры публичных датасетов:

MNIST (https://www.kaggle.com) – 70 000 рукописных цифр (0–9), идеально для обучения распознаванию чисел.

CIFAR-10 (https://www.kaggle.com) – 60 000 цветных изображений 10 классов: самолёты, автомобили, птицы, кошки и т.д.

IMDB Reviews (https://www.imdb.com/) – 50 000 отзывов на фильмы с пометкой «положительный» или «отрицательный» – для анализа тональности текста.

Titanic Dataset (https://www.kaggle.com) – данные о пассажирах «Титаника»: возраст, пол, класс билета и выжил ли человек – для предсказания выживаемости.

Экосистема Kaggle

Отдельного упоминания заслуживает платформа Kaggle, которая является не просто хранилищем данных, в том числе содержит публичные датасеты, которые упомянуты выше, а целой экосистемой для специалистов. Kaggle – это крупнейшая в мире онлайн-платформа для соревнований по анализу данных и машинному обучению, принадлежащая Google. По сути – это «спортивная арена» для специалистов по данным, но открыта она для всех (регистрация бесплатна, большинство данных и курсов – тоже).

На Kaggle представлены: *соревнования* (Competitions), которые включают реальные задачи от компаний и организаций; *наборы данных* (Datasets), содержащие тысячи публичных датасетов на любую тему – от медицины до кинематографа; *ноутбуки* (Notebooks) – это готовые примеры кода и анализа от других пользователей (можно изучать, копировать, адаптировать); *курсы* (Learn), в том числе бесплатные интерактивные курсы по Python, машинному обучению, визуализации данных – с практикой прямо в браузере; *форумы и обсуждения*, где можно задавать вопросы и учиться у экспертов.

Другие источники публичных датасетов:

– Google Dataset Search (https://datasetsearch.research.google.com/),

– UCI Machine Learning Repository (https://archive.ics.uci.edu/) – репозиторий машинного обучения Калифорнийского университета в Ирвайне,

– Hugging Face Datasets (https://huggingface.co/docs/datasets/index) – особенно полезен для текста и NLP,

– датасеты от государств – например, https://data.gov.ru/ (РФ), https://data.gov/ (США).

Зачем пользователю знать, откуда брались данные, на которых обучалась нейросеть

Даже если вы не будете строить модели, понимание, откуда берутся данные для ИИ, поможет критически оценивать его выводы. Например, «А на каких данных обучалась эта модель? Были ли там представители моей группы? Не было ли предвзятости?».

То есть в целом речь идет о понимании мира, управляемого данными. Навык работы с данными становится частью общей культуры, аналогично умению пользоваться Интернетом или офисными приложениями. Его освоение дает существенное преимущество в любой современной профессии.

Конфиденциальность и согласие при сборе данных

Процесс сбора данных для обучения нейросетей поднимает серьезные этические вопросы. Многие публичные датасеты включают личные фотографии, тексты и другую информацию, собранную из открытых источников без явного согласия авторов. Например, датасет LAION-5B, использованный для обучения нейросети для генерации изображений Stable Diffusion, содержит миллиарды изображений, скачанных из Интернета. Среди них оказались личные фотографии людей, которые никогда не давали согласия на использование своих изображений для обучения ИИ. В России Федеральный закон «О персональных данных» требует получения согласия на обработку персональных данных, включая биометрические. Однако в контексте больших данных для ИИ практическая реализация этих требований остается сложной задачей. В 2020 году в России был принят Федеральный закон «Об экспериментальных правовых режимах в сфере искусственного интеллекта», который частично регулирует эти вопросы, но многие аспекты остаются в серой зоне. Этически ответственный подход к сбору данных требует обеспечения анонимизации персональных данных, предоставления механизмов для удаления данных по запросу (право на забвение), прозрачного информирования о том, как будут использоваться собираемые данные, а также особой осторожности при работе с данными уязвимых групп населения.

При использовании нейросетевых сервисов стоит помнить: ваши запросы и загружаемые данные могут стать частью обучающих выборок для будущих версий этих систем, если в пользовательском соглашении не указано иное. Внимательно читайте политику конфиденциальности сервисов и используйте опции отказа от сбора данных, если они доступны.

2. Подготовка и преобразование данных

После сбора сырых данных следует этап их подготовки. Этот процесс состоит из нескольких ключевых операций.

1. Исследование и очистка данных

В реальном мире данные всегда содержат «мусор» – ошибки, пропуски или нереалистичные значения. Их нужно найти и исправить. Например, в таблице с ростом и весом людей может встретиться запись «рост 250 см, вес 5 кг». Это явная ошибка (выброс), которую нужно удалить или заменить на среднее значение. Пропуски в данных (например, не указан возраст) также нужно обработать. В зависимости от ситуации их заполняют средним, медианой или модой, предсказанным значением, а иногда строки с пропусками просто удаляют. Качественная очистка данных помогает уменьшить ошибочные предсказания модели.

2. Разметка данных

Особое место в подготовке данных занимает их разметка, или аннотирование. Этот процесс заключается в добавлении к данным правильных ответов (меток). Разметка бывает трех типов, соответствующих трем основным видам обучения. При обучении с учителем, которое является самым распространенным, данные тщательно размечаются человеком. Например, для задачи классификации животных фотография кошки помечается как «кошка», а собаки – как «собака». В системе для дрессировки собак камера делает разметку «сидеть», «лежать» или «стоять» в реальном времени, чтобы система могла принять решение о выдаче лакомства. При обучении без учителя данные остаются неразмеченными, и сеть сама должна найти скрытые закономерности и структуры. При полуобучении используется смешанный подход: часть данных размечена, а часть нет, что позволяет сочетать эффективность обучения с учителем с возможностями обучения без учителя.

3. Предобработка и инженерия признаков

Инженерия признаков – это создание новых, более полезных данных на основе имеющихся. Например, из даты «01.05.2023» можно извлечь день недели – «понедельник» и месяц – «май». Тогда модель может обнаружить, например, что по понедельникам продажи всегда выше, и учесть эту закономерность.

Кроме того, данные приводят к нужному формату. Изображения приводят к единому размеру (например, 256x256 пикселей), чтобы их можно было подавать на вход нейронной сети. Для текста проводят токенизацию, т.е. текст разбивают на отдельные слова или части слов (токены), а затем

преобразуют эти токены в цифровые коды (векторы), понятные компьютеру. Именно так слова «Москва» или «Санкт-Петербург» превращаются в числа. Для числовых данных производят масштабирование, т.е. приводят их к единому масштабу, что включает нормализацию и стандартизацию. Представьте, что вы сравниваете стоимость квартиры в миллионах рублей и расстояние до метро в сотнях метров. Но цифры в миллионах будут несоизмеримо больше и «перекричат» собой все остальное. Тогда чтобы этого избежать, все данные приводят к одному масштабу (например, сжимают в диапазон от 0 до 1), и это помогает нейросети учиться быстрее и точнее. Нормализация позволяет каждому параметру вносить соизмеримый и адекватный вклад в результат обучения, а не полагаться на произвольные различия в изначальных масштабах данных.

4. Разделение данных на выборки

Разделение данных – критически важное действие для последующей проверки знаний нейросети.

Весь набор данных делится на три части:

– *обучающая выборка* (самая большая), по которой модель непосредственно учится;

– *валидационная выборка* – это своего рода «пробный экзамен» во время обучения, т.к. по ней проверяют, как хорошо модель усваивает материал, и корректируют ее настройки;

– *тестовая выборка* – эту часть данных модель не видит ни разу до самого конца обучения, потому что они нужны для финальной оценки – насколько хорошо модель научилась решать задачу в целом.

Вернемся к нашей задаче отличить кошку от собаки:

а) задача – отличить кошку от собаки,

б) находим тысячи изображений кошек и собак (сбор),

в) удаляем снимки, где животное не видно, или где на фото и то, и другое (очистка),

г) приводим все изображения к одному размеру и нормализуем цветовые значения пикселей (условно говоря, «подравниваем» яркость и контраст всех изображений), чтобы изображения имели сопоставимую яркость и цветовой баланс независимо от исходных фотографий, которые могли быть сделаны в условиях разной освещенности и т.п. (преобразование),

д) 80% картинок используем для обучения, 10% – для промежуточной проверки (валидации), и 10% – для финального теста (разделение).

5. Аугментация данных (опционально)

После того как данные разделены на выборки, для искусственного увеличения размера и разнообразия обучающей выборки часто применяют аугментацию. Это техника искусственного расширения обучающего набора данных, при которой к исходным данным применяются случайные, но реалистичные преобразования. Для изображений это могут быть повороты, отражения, изменение яркости/контраста, масштабирование или добавление шума. Для текста – замена слов синонимами, случайное удаление слов или перефразирование. Аугментация особенно полезна, когда сбор дополнительных данных невозможен, дорогостоящ или требует значительных временных затрат, она помогает модели стать более устойчивой и лучше обобщать, предотвращая запоминание конкретных примеров из обучающего набора (переобучение). Важно, что аугментация применяется только к обучающей выборке, валидационная и тестовая остаются неизменными для объективной оценки.

6. Создание модуля загрузки данных (DataLoader)

Финальным техническим шагом является создание механизма для эффективной подачи данных в нейросеть во время обучения – DataLoader. Вместо загрузки всего набора данных в память сразу (что невозможно для больших датасетов), DataLoader организует подачу данных небольшими порциями, называемыми батчами (или мини-батчами). Это позволяет экономить память и ускоряет обучение, так как веса модели обновляются после обработки каждого батча.

DataLoader также выполняет две другие важные функции: *перемешивание* (Shuffling) – случайным образом изменяет порядок данных в каждой эпохе (полном проходе по обучающей выборке), чтобы модель не запоминала порядок следования примеров; *предзагрузка* (Preloading) – пока модель обрабатывает текущий батч, DataLoader в фоновом режиме загружает следующий, что максимально использует вычислительные ресурсы (особенно GPU) и исключает их простой.

Таким образом, подготовка данных – это кропотливая работа, которая является залогом успеха модели. Полный цикл подготовки данных представлен на Схеме 3. Каждый из этапов подготовки требует внимательного и тщательного подхода, так как влияние качества данных на производительность нейросети может быть колоссальным. Понимание и реализация всех этих аспектов – это необходимые условия для построения эффективных и надежных моделей искусственного интеллекта.

Схема 3. «Подготовка данных к обучению нейросети»

1. ПОСТАНОВКА ЗАДАЧИ И ПЛАНИРОВАНИЕ ДАННЫХ

Какую проблему должна решить нейросеть (классификация, регрессия, генерация и т.д.), какие именно данные нужны, в каком формате и в каком объеме.

▼

2. СБОР ДАННЫХ

Получаем «сырые» данные из различных источников (публичные датасеты, автоматизированный сбор данных с веб-сайтов, сбор через API, ручной сбор и разметка, генерация синтетических данных с помощью алгоритмов или других моделей.

▼

3. ИССЛЕДОВАНИЕ И ОЧИСТКА ДАННЫХ

Исправляем очевидные ошибки и аномалии: удаление строк/столбцов с пропусками, заполнение средним, медианным значением или модой, интерполяция, предсказание другой моделью и т.д.; поиск аномальных значений, удаление, замена или преобразование; устранение опечаток, несоответствий формата.

▼

4. РАЗМЕТКА ДАННЫХ — ДЛЯ ЗАДАЧ С УЧИТЕЛЕМ

Присваиваем данным правильные «ответы», например, указание, какой объект находится на картинке; обводка объектов ограничивающими рамками; разметка частей речи или именованных сущностей.

▼

5. ПРЕДОБРАБОТКА И ИНЖЕНЕРИЯ ПРИЗНАКОВ

Преобразуем данные в требуемый формат: нормализация и стандартизация, кодирование категориальных признаков, токенизация, векторизация, векторные представления, приведение всех изображений к одному размеру, нормализация пикселей, создание новых признаков.

▼

6. РАЗДЕЛЕНИЕ НА ВЫБОРКИ

Обучающая выборка 80%	Валидационная выборка 10%	Тестовая выборка 10%

▼

7. АУГМЕНТАЦИЯ ДАННЫХ (ОПЦИОНАЛЬНО)

Искусственное увеличение размера обучающей выборки за счет применения случайных, но реалистичных преобразований к исходным данным.

▼

8. СОЗДАНИЕ МОДУЛЯ ЗАГРУЗКИ ДАННЫХ (DATALOADER)

Подготавливаем механизм для эффективной подачи данных в нейросеть во время обучения: пакетная обработка, перемешивание, предзагрузка.

3. Системные ограничения данных

Обучение нейросетей сталкивается с фундаментальной проблемой – все обучающие данные представляют прошлое, а не настоящее или будущее. Это создает базовое ограничение: нейросети хорошо воспроизводят закономерности из прошлого, но не способны предсказать радикальные изменения или инновации. Более того, данные неизбежно содержат систематические искажения общества, в котором они были созданы. Исследования показывают, что нейросети, обученные на реальных текстах, воспроизводят и даже усиливают существующие стереотипы по полу, расе и другим признакам. Это не просто технический недостаток, а принципиальное ограничение метода обучения, при котором система извлекает и воспроизводит статистические закономерности из предоставленных данных. Наконец, важно понимать, что данные – это всегда упрощенная модель реальности. Многие аспекты человеческого опыта просто не представлены в оцифрованной форме, а значит, недоступны для обучения нейросетей. Эмоциональный интеллект, интуиция, творческое озарение – эти человеческие качества лишь имитируются современными ИИ-системами, но не воспроизводятся по-настоящему.

B. Выбираем метод обучения и настраиваем процесс

После того как задача определена и данные подготовлены, начинается самое главное действие – само обучение. Процесс обучения представляет собой сложный, итеративный и многоступенчатый цикл. Он не происходит за один проход, а состоит из множества повторений, где на каждой итерации сеть делает небольшой шаг к совершенству.

1. Гиперпараметры

Перед началом обучения мы должны задать начальные значения гиперпараметров – это «ручки настройки» всего процесса. Их подбор часто является сложной и ресурсоемкой задачей, которая может проводиться как вручную, так и автоматически (путем систематического перебора или более сложных методов). К гиперпараметрам относятся скорость обучения (Learning Rate), количество эпох (Epochs), размер пакета (Batch Size) и т. д. Подбор оптимальных гиперпараметров может значительно улучшить качество модели.

Скорость обучения – это самая важная «ручка настройки». Представьте, что нейросеть – это человек, который спускается с горы к реке (ищет наименьшую точку ошибки). Слишком высокая скорость – она будет «перескакивать большими прыжками» и может нечаянно «перепрыгнуть» наименьшую точку ошибки, слишком низкая – она будет «спускаться мелкими шажками» и добираться до цели очень долго.

Размер пакета – это количество примеров, которые нейросеть просматривает за один раз перед тем, как сделать вывод и обновиться. Просмотреть 10 примеров – быстрее, но менее точно. Просмотреть 1000 – медленнее, но надежнее.

Эпоха – основная единица измерения процесса обучения. Одна эпоха – это один полный цикл обучения на всем наборе данных. Например, если в датасете 1000 фотографий животных, то одна эпоха означает, что сеть видела и проанализировала все 1000 изображений.

Количество эпох – сколько раз нейросеть пройдет через весь набор данных для обучения. Одно прохождение – одна эпоха. Слишком мало – не выучит. Слишком много – приведет к переобучению, т.е. модель «зазубрит» обучающие данные, включая их шум и случайные особенности, и потеряет способность к обобщению на новые данные.

Однако на практике обучение происходит не одним большим блоком, а небольшими порциями. Именно здесь вводится понятие *мини-батча* (Mini-Batch). Вместо того чтобы показывать всю тысячу изображений сразу (что может быть невозможно из-за ограничений памяти GPU), данные делятся на группы – мини-батчи. Например, мини-батч может состоять из 16 фотографий. Тогда один проход по всему датасету (одна эпоха) будет состоять из 63 таких шагов (1000 / 16 = 62.5, округляем до 63, при этом последний мини-батч будет содержать только 8 изображений). Сразу отметим, что просто «батч» (без «мини») означает весь набор обучающих данных и он используется при пакетном градиентном спуске (Batch Gradient Descent), когда за один шаг (итерацию) обрабатываются все данные (весь батч). В этом случае эпоха состоит из одной итерации. Этот метод редко используется в глубоком обучении из-за высоких требований к памяти. Однако в контексте нейросетей, когда говорят «батч» или «размер пакета (Batch Size)», почти всегда подразумевают именно «мини-батч» – ту самую порцию данных (например, 16, 32, 128 изображений), которая подается на одну итерацию. Это уточнение поможет не запутаться, если потребуется более глубоко разобраться в теме обучения нейросетей. Кстати, при стохастическом градиентном спуске (Stochastic Gradient Descent) размер мини-батча равен 1,

т.е. итерация – это обработка одного примера. Это частный случай мини-батча, но со своими особенностями.

Ну и, наконец, *итерация* — это базовый шаг в процессе обучения нейронной сети, представляющий собой одно обновление параметров модели после обработки определённого набора данных (мини-батча). Итерация – это один шаг внутри эпохи.

2. Цикл обучения

Схема 4. «Цикл обучения нейросети»

На каждой итерации происходит последовательность действий, соответствующая циклу обучения (Схема 4), который включает:

1. **Прямой проход / Прямое распространение** (Forward Pass) – данные подаются на вход сети, и сигнал проходит через все слои, преобразуясь по пути. Преобразование заключается в вычислении взвешенной суммы входов (с учетом весов и смещений) и последующем применении функции активации. На выходе получаем предсказание (предполагаемый ответ).

2. **Расчёт ошибки** (Loss Calculation) – полученное предсказание сравнивается с реальным, эталонным значением (меткой) с помощью специальной функции – функции потерь (Loss Function) Результат этого сравнения – одно число, величина ошибки (Loss), которое показывает, насколько предсказание сети было плохим.

3. **Обратное распространение ошибки** (Backpropagation) – на этом этапе используется одноименный алгоритм, который является центральным механизмом обучения. Его цель – рассчитать вклад каждого веса в общую ошибку. Можно сказать, что алгоритм обратного распространения – это «механизм вычисления». Он отвечает на вопрос: «В какую сторону и насколько каждый вес влияет на ошибку?» (рассчитывает градиенты для каждого веса).

4. **Обновление весов** (Weights Update) – это завершающий этап, на котором на основе градиентов, рассчитанных алгоритмом обратного распространения, происходит непосредственная корректировка всех весов и смещений с помощью специального алгоритма оптимизации.

И здесь важно подчеркнуть их тандем: алгоритм обратного распространения – это «механизм вычисления», который отвечает на вопрос: «В какую сторону и насколько каждый вес влияет на ошибку?», алгоритм оптимизации – это «механизм действия», он не отвечает на вопросы, а действует – изменяет веса на основании полученных градиентов. Без обратного распространения у оптимизатора не было бы точной карты для движения. Без алгоритма оптимизации все вычисления обратного распространения остались бы просто числами в памяти. Они работают исключительно в паре.

«Базовым алгоритмом» в семействе алгоритмов оптимизации (оптимизаторов) является градиентный спуск (Gradient Descent), а его самой известной разновидностью – стохастический градиентный спуск (Stochastic Gradient Descent, SGD). Можно представить SGD как «строгого тренера-перфекциониста», который заставляет сеть учиться, анализируя каждый пример (или небольшой пакет примеров) по отдельности и немедленно внося

корректировки. Это эффективно, но может быть «нервным»: процесс обучения часто бывает хаотичным и колеблется вокруг идеального решения.

Для того чтобы сделать процесс обучения более быстрым и стабильным, были разработаны более продвинутые оптимизаторы. Важнейшим улучшением SGD стало добавление момента (Momentum), который привносит в обучение инерцию, учитывая не только текущий градиент, но и направление предыдущих шагов. Настоящий прорыв совершили адаптивные алгоритмы оптимизации, такие как адаптивный градиент (Adaptive Gradient, Adagrad), распространение среднеквадратичного значения (Root Mean Square Propagation, RMSprop), адаптивная оценка моментов (Adaptive Moment Estimation, Adam). Все они основаны на градиентном спуске, но добавляют к нему интеллектуальные механизмы. Главное их отличие от SGD в том, что они не используют единую скорость обучения для всех весов, вместо этого они адаптируют скорость индивидуально для каждого параметра, исходя из истории его градиентов.

Это решает ключевые проблемы SGD и обеспечивает:

подстройку под каждый параметр – веса, которые редко обновляются, получают больший шаг, а часто меняющиеся – меньший, что ускоряет обучение;

устойчивость к «шуму» – алгоритмы «усредняют» историю градиентов, поэтому один «шумный» пример или выброс не заставляет сеть сделать катастрофически неверный шаг;

быстрое уменьшение ошибки – адаптивность и инерция помогают алгоритму эффективнее находить настройки, при которых ошибка сети становится минимально возможной.

Выбор алгоритма зависит от задачи, архитектуры нейросети и характеристик данных (Таблица 9).

Этот четырехэтапный цикл повторяется снова и снова, итерация за итерацией, эпоха за эпохой. После каждой эпохи обычно проводится оценка качества модели на отдельном, независимом наборе данных – валидационном датасете. Это позволяет отследить, как модель работает на данных, которых она еще не видела во время обучения, и предотвратить переобучение. Процесс продолжается до тех пор, пока качество модели на валидационном наборе перестает значительно улучшаться, или до достижения заранее установленного максимального числа эпох.

Обучение нейросети – это сложный, но поддающийся настройке процесс, похожий на воспитание ученика. Нам нужно выбрать метод обучения («тренера» или «педагога»), правильно выставить «ручки настройки»

и использовать техники, которые помогут ученику понять суть, а не просто зазубрить факты. Именно этот итеративный процесс экспериментов и тонкой настройки превращает сырые данные в настоящие знания и интеллект.

Таблица 9. «Сводная таблица алгоритмов оптимизации»

Алгоритм	Ключевая идея	Плюсы	Минусы	Аналогия
SGD	Градиент здесь и сейчас.	Простота.	Медленный, колеблется.	Пешеход с компасом.
SGD + Momentum	Градиент + Инерция.	Быстрее, гасит колебания.	Все еще одна скорость для всех.	Шарик, катящийся с горы.
Adagrad	Адаптация шага для каждого параметра.	Не требует тонкой настройки скорости.	Со временем «забывать» перестает.	Пешеход, помнящий все ухабы.
RMSprop	Adagrad + «Забывание».	Решает проблему Adagrad.	Не хватает «инерции».	Пешеход с короткой памятью.
Adam	Momentum + RMSprop.	Быстрый, стабильный, адаптивный.	Сложнее вычисляется.	Человек на скейтборде с короткой памятью на неровности.

Г. Типы обучения

Существует три основных парадигмы обучения (Схема 5):

«Обучение с учителем» (Supervised Learning) – это самый распространенный метод, сеть обучается на размеченных данных, где каждому входу соответствует правильный ответ. Его аналогия с человеческим обучением очевидна: ученик получает знания от учителя, который предоставляет ему не только материал, но и готовые ответы на вопросы. В контексте нейросетей это означает, что для обучения используется размеченный датасет – набор данных, где каждому примеру присвоена правильная метка или ответ. Например, для задачи классификации цветов нейросеть обучается на тысячах фотографий, где каждая фотография помечена

как «роза», «ромашка» или «нарцисс». Сеть сравнивает свое предсказание («я думаю, это ромашка») с истинной меткой и корректирует свои веса, чтобы в следующий раз делать меньше ошибок. Этот метод используется для большинства задач классификации и регрессии, таких как распознавание рукописных цифр (MNIST), медицинская диагностика по снимкам, фильтрация спама в почте и предсказание цен на недвижимость.

«Обучение без учителя» (Unsupervised Learning) – это метод, при котором сеть работает с неразмеченными данными. У нее нет готовых ответов, и она должна сама найти в данных скрытые закономерности, структуры или аномалии. Это похоже на то, как исследователь изучает большой массив необработанных данных, чтобы найти группы похожих объектов или выявить что-то необычное. Например, нейросеть может проанализировать миллионы покупок клиентов в Интернет-магазине и сгруппировать товары, которые часто покупаются вместе, что поможет в маркетинговых рекомендациях (подгузники → нагрудник). Другая задача при реализации метода «обучение без учителя» – кластеризация, когда сеть группирует, например, статьи новостного портала по темам, не имея никаких предварительных указаний. Метод также применяется для обучения нейросети обнаружению аномалий, например, мошеннических банковских операций. Кроме того, методом «обучение без учителя» обучают такие модели, как автоэнкодеры, которые используются, среди прочего, для удаления шума из изображений и медицинских сканов.

«Обучение с подкреплением» (Reinforcement Learning) – это третий и принципиально отличающийся от остальных метод. Он основан на идее проб и ошибок в определенной среде с целью максимизировать совокупную награду. Это аналогично дрессировке животного, где положительное подкрепление (вознаграждение) используется для закрепления нужного поведения, а отрицательное (штраф) — для минимизации нежелательного. Но RL – не просто поощрение, а стратегическое управление через вознаграждения и штрафы с учётом долгосрочной выгоды. Агент (нейросеть) совершает действие в среде, получает вознаграждение (награду или штраф) и на основе этого опыта учится выбирать действия, которые в долгосрочной перспективе приводят к максимальной выгоде. Яркие примеры успеха этого метода – AlphaGo, который научился играть в Го на уровне чемпионов мира, и системы управления для беспилотных автомобилей (где RL, например, отвечает за стратегическое вождение). Этот метод идеально подходит для задач, где нужно принимать серию последовательных решений, таких как управление роботами, игровые стратегии или навигация.

Схема 5. «Типы обучения»

С УЧИТЕЛЕМ (Supervised)

Аналогия: учебник с ответами
Механизм: ВХОД → правильный ответ
Пример: распознавание лиц

БЕЗ УЧИТЕЛЯ (Unsupervised)

Аналогия: исследователь в лаборатории
Механизм: ДАННЫЕ → самостоятельные поиск групп
Пример: кластеризация

С ПОДКРЕПЛЕНИЕМ (Reinforcement)

Аналогия: дрессировка собаки
Механизм: ДЕЙСТВИЕ → награда или штраф
Пример: AlphaGo, автопилот

Три описанных подхода – это фундамент, но на практике их часто комбинируют. Например, существует метод полуобучения (Semi-supervised learning), который идеально подходит для ситуаций, когда у нас есть немного размеченных данных и очень много неразмеченных. Нейросеть сначала учится на том, что размечено, а затем использует эти знания для анализа оставшейся части данных, находя в ней закономерности и тем самым размечая их самостоятельно. Это особенно актуально в областях, где разметка данных дорога и трудоемка, например, в медицине, где каждый снимок должен быть промаркирован высококвалифицированным врачом.

Другой яркий пример комбинирования подходов – **обучение через соперничество** (Generative Adversarial Networks / GAN) или состязательные генеративные сети, ранее уже упомянутые в разделе про архитектурные типы нейросетей. В этой архитектуре, как мы помним, две нейросети – генератор и дискриминатор – состязаются друг с другом. Генератор пытается создать реалистичные данные (например, фейковые лица), а дискриминатор – отличить настоящие данные от поддельных. Этот метод используется для генерации высококачественного контента, отображаемого в таких сервисах, как «This Person Does Not Exist» (в переводе «Этот человек не существует») – веб-сайт, который с помощью искусственного интеллекта генерирует реалистичные фотографии лиц людей, не существующих

в реальности. Каждое изображение создается алгоритмом в момент загрузки страницы. Используется для создания уникальных, не защищенных авторскими правами изображений для рекламы, веб-сайтов или презентаций, при разработке игр и медиа (быстрое создание лиц для второстепенных персонажей в играх или фильмах), создания аватара в социальных сетях вместо настоящей фотографии и т.д. И хотя технология неидеальна, успех «This Person Does Not Exist» породил целую волну подобных сервисов, которые генерируют несуществующие изображения других категорий, например, животных, аниме-персонажей и даже еды.

Д. Оценка качества обучения и методы улучшения

Обучение нейросети – это цикличный процесс настройки и контроля, цель которого – создать не просто «умную», но и устойчивую модель, которая хорошо работает на новых, незнакомых данных. Чтобы оценить, чему модель научилась по-настоящему, ее нужно протестировать на данных, которых она раньше не видела.

1. Метрики

Для оценки качества и эффективности работы обучаемой нейросети используются *метрики*, т.е. показатели, которые количественно характеризуют производительность обучаемой модели на основе её способности предсказывать правильные ответы для валидационной и тестовой выборок (мы же помним, что предварительно все данные были разделены на 3 части: обучающая, тестовая и валидационная). Метрики на валидационной выборке отслеживаются в процессе обучения для настройки гиперпараметров и контроля переобучения, а на тестовой — для финальной оценки качества уже обученной модели. Метрики позволяют понять, насколько хорошо нейросеть справляется с задачей, и выявить, где её можно улучшить.

Таблица 10. «Ключевые метрики»

Тип задачи	Ключевые метрики	Что показывают
Классификация	Accuracy, Precision, Recall, F1-мера	Как хорошо модель различает классы.
Регрессия	MAE, MSE, R^2	Насколько близки предсказания к реальным значениям.

Выбор метрики зависит от задачи (Таблица 10). Не существует единственно верной «оценки» для всех.

Метрики для классификации (например, «кот» или «собака»):

1) *Точность как процент правильных ответов* (Accuracy) – просто и понятно, но может обманывать, если классы не сбалансированы (например, 99% котов и 1% собак).

2) *Точность как минимизация ложных срабатываний* (Precision) и *полнота как минимизация пропусков* (Recall):

- Precision – из всех случаев, когда модель сказала «кот», сколько раз она была права. Это важно, когда ложная тревога дорого стоит (например, спам в почте).
- Recall – из всех реальных котов, сколько модель смогла найти. Критично в медицине, где пропустить болезнь недопустимо.

3) *F1-мера* – усредненный показатель, балансирующий между точностью и полнотой (главный инструмент, когда классы несбалансированы).

Метрики для регрессии (например, предсказание цены):

1) *Средняя абсолютная ошибка* (MAE) – средняя величина ошибки, она проста для интерпретации.

2) *Среднеквадратичная ошибка* (MSE) – усиливает влияние больших ошибок, она полезна, когда крупные ошибки особенно нежелательны.

3) *Коэффициент детерминации* (R^2) – показывает, насколько хорошо модель предсказывает данные по сравнению с простым средним значением. Его можно интерпретировать как «долю объясненной изменчивости». Представьте, что вам нужно угадать рост случайных людей на улице. Если вы будете всегда называть средний рост по стране, вы будете иногда близки, а иногда далеки от истины. R^2 показывает, насколько ваша умная модель (учитывающая вес, возраст, пол) уменьшила эту «степень промаха» по сравнению с простым угадыванием среднего роста. Если $R^2 = 0.8$, значит, модель объяснила 80% разброса в росте людей.

$R^2 = 1$ – идеальная модель, объясняет 100% изменчивости данных;

$R^2 = 0$ – модель работает не лучше, чем если бы мы всегда предсказывали одно и то же число — среднее значение из правильных ответов в обучающих данных;

$R^2 < 0$ – модель работает хуже, чем предсказание средним значением, – это явный признак того, что модель негодная.

2. Переобучение и недообучение

Как мы уже кратко упоминали ранее, главные опасности в обучении – это переобучение и недообучение. Давайте разберем их подробнее. Анализируя поведение модели на обучающей и валидационной выборках, мы можем диагностировать две главные болезни.

Переобучение (Overfitting) – модель «зазубрила учебник, но не поняла сути». Симптомы – ошибка на обучающих данных очень низкая, а на валидационных – высокая. Лечение – упрощение модели, применение методов регуляризации (см. ниже).

Недообучение (Underfitting) – модель «не смогла разобраться в материале». Симптомы – высокая ошибка и на обучающих, и на валидационных данных. Лечение – усложнение модели (больше слоев/нейронов), обучение дольше, улучшение качества данных.

Идеальная модель находится ровно посередине: она хорошо справляется и с учебными примерами, и с новыми данными.

3. Методы улучшения модели

1. **Техники регуляризации** (боремся с переобучением):

Дропаут (Dropout) – во время обучения мы случайным образом «выключаем» часть нейронов в сети. Это заставляет остальные нейроны становиться более самостоятельными и не полагаться слепо на «соседей». Это как готовиться к экзамену, закрывая случайные абзацы в учебнике – так вы запоминаете суть, а не просто заучиваете расположение текста.

L1/L2-регуляризация – добавляем в функцию потерь небольшой «штраф» за излишне большие веса. Это заставляет сеть искать не просто точное, но и простое решение. Простое решение почти всегда самое верное и хорошо работающее на новых данных.

2. **Работа с данными** (основа хорошего обучения):

Увеличение набора данных (Data Augmentation) – искусственное расширение обучающей выборки путем легких преобразований. Для изображений это повороты, отражения, изменение яркости. Для текста – замена синонимов, парафраз.

Балансировка классов – если в данных одного класса (например, «спам») много, а другого («не спам») – мало, модель будет предвзятой. Решения: сбор больше данных для редкого класса, искусственная генерация примеров или специальные алгоритмы.

3. **Настройка гиперпараметров** (ищем идеальные «ручки настройки»). Если простые методы не помогли, пора запускать поиск.

Сеточный поиск (Grid Search) – систематически перебираем все возможные комбинации заранее заданных гиперпараметров (например, скорость_обучения = [0.1, 0.01, 0.001], размер_пакета = [32, 64]). Надежно, но процесс может быть очень долгим.

Случайный поиск (Random Search) – случайным образом выбираем комбинации из заданного диапазона. Часто находит хорошее решение гораздо быстрее сеточного поиска.

4. **Если перепробовано все вышеперечисленное, а модель все еще неудовлетворительна**, то возможны следующие варианты исправления ситуации:

1) *Вернуться к данным* – это самый частый источник проблем. Проверьте данные на наличие ошибок, шума и некорректных разметок. Как мы уже говорили, «мусор на входе – мусор на выходе».

2) *Пересмотреть архитектуру* – возможно, эта нейросеть принципиально не подходит для задачи. Попробуйте сверточные сети (CNN) для изображений, рекуррентные (RNN) или Трансформеры для текста.

3) *Ансамблирование* (Ensemble) – объедините несколько разных моделей для получения итогового предсказания. «Командный ум» часто оказывается точнее и стабильнее, чем мнение одного эксперта.

Процесс улучшения нейросети – это итеративный цикл «оценить → диагностировать → исправить → снова оценить». Понимание этого процесса и владение его инструментами – ключ к созданию по-настоящему эффективных моделей искусственного интеллекта.

E. Для чего полезно знать, как обучались нейросети

Мы уже живём в мире, где искусственный интеллект принимает решения за нас – и будет делать это всё чаще. Мы, не являясь специалистами в сфере ИИ, не обязаны писать код или настраивать веса нейронов. Но понимание основ обучения нейросетей – это как понимание законов физики для водителя: человек за рулем – не конструктор автомобиля, но знает, что при резком торможении машину может занести. Это знание помогает водителю предвидеть поведение автомобиля в дорожной ситуации, а пользователю нейросети – как будет вести себя ИИ. Важно научиться пользоваться инструментом, не доверяя ему слепо, но понимая его возможности и ограничения. Тогда можно контролировать результаты и эффективно использовать технологии.

Нейросети – это алгоритмы, которые учатся на примерах, как ребёнок: пробуют, ошибаются, корректируются. Понимание этого позволяет критически оценивать результаты. Нейросеть может выдать красивый текст, точный диагноз или идеальную рекомендацию – но она может ошибаться, причём системно. Почему ChatGPT иногда «врёт»? Потому что он учится на данных, где есть ошибки, противоречия и предвзятости. Почему медицинский ИИ может хуже распознавать болезни у женщин? Потому что его учили в основном на мужских данных. Зная, как происходит обучение, вы понимаете: результат ИИ – это отражение данных, на которых его обучали, а не истина в последней инстанции. Роль человека – проверять, задавать вопросы, перепроверять выводы.

Знание основ обучения ИИ помогает ставить корректные задачи, интерпретировать результаты и не делегировать нейросети то, что требует человеческого суждения.

Работодатели всё чаще ищут не просто юристов, врачей или экономистов – а специалистов, в том числе умеющих работать с ИИ. Не нужно быть программистом, но нужно понимать, какие задачи ИИ может решить, а какие – нет, как подготовить данные, чтобы ИИ дал хороший результат, как интерпретировать выводы ИИ и где искать подвох. Это новая грамотность – примерно, как умение пользоваться Excel 20 лет назад. Кто освоит – будет востребован. Кто проигнорирует – останется без рычагов влияния.

Это опять же не про технологии – это про власть, ответственность и возможности.

Представьте, что через 5 лет вас спросят на собеседовании: «Как вы используете ИИ в своей работе?». Те, кто сможет не просто назвать пару

инструментов, но и объяснить, как они работают и какие данные им нужны, получат серьезное преимущество. Это знание – ваша профессиональная страховка в эпоху ИИ.

Наконец, понимание принципов работы нейросетей позволяет вам стать этически ответственным пользователем. Зная, что нейросети воспроизводят и иногда усиливают социальные предубеждения из обучающих данных, вы будете более критически оценивать их выводы по чувствительным вопросам. Осознавая проблематику авторских прав, вы сможете делать информированный выбор: использовать ли генеративные модели для имитации стиля конкретных авторов или предпочесть более оригинальный подход. В мире, где границы технологий и этики постоянно пересматриваются, понимание механизмов работы ИИ дает возможность не просто следовать за технологическим прогрессом, но и активно формировать его направление через осознанные решения в качестве пользователя, потребителя и гражданина.

ГЛАВА 5. ИСКУССТВО ПРОМПТИНГА: КАК РАЗГОВАРИВАТЬ С НЕЙРОСЕТЬЮ

§ 1. Понятийный аппарат

Представьте, что в вашем распоряжении имеется высококвалифицированный эксперт, который знает и умеет очень и очень многое, но он совершенно не посвящен в ваши задачи и проблемы и не знает, что он может сделать именно для вас. Он впервые общается с вами, поэтому для выполнения поставленной задачи ему нужна информация, которая объяснит, что именно нужно сделать, для чего или как планируется использовать результаты работы, в каком виде вы хотите получить результат, какого объема и т. д.

Правильно составленная для нейросети инструкция о том, что ей следует сделать, называется промптом. Вместо 3–4 попыток вы получите нужный результат с первого раза, ну или, по крайней мере, потратите на этот процесс значительно меньше времени.

Слово «промпт» происходит от английского слова «prompt», что означает «стимул», «толчок», «запрос». В контексте ИИ – это то, что «запускает» работу модели.

Промпт – это входные данные (текст, изображение, звук, музыка), которые пользователь предоставляет модели для получения желаемого результата, или иначе – это инструкция, вопрос или подсказка, которую человек вводит в искусственный интеллект (например, в чат-бот на основе ИИ, генератор изображений, текстовый редактор и т.д.), чтобы получить требуемый ответ или результат.

Промптинг – это само действие, единичный акт написания запроса к ИИ или введения данных для анализа и дальнейшей работы. Написал запрос – получил ответ.

Промпт-инжиниринг – это процесс разработки шаблона промпта под конкретные цели.

Промпт-инжиниринг включает:

проектирование – создается шаблон промпта с переменными: «Выступи в роли [РОЛЬ]. Проанализируй [ТЕКСТ] и выдели [N] ключевых тезисов. Представь результат в формате: [ФОРМАТ]»;

тестирование – разработанный шаблон тестируется на 10 разных статьях, с разными ролями (эколог, экономист, политик) и форматами вывода;

анализ – проверяется, насколько стабильно модель следует инструкциям, нет ли «галлюцинаций», подходят ли форматы вывода;

*улучшени*е – на основе анализа промпт дорабатывается. Например, добавляется инструкция: «Если в статье меньше [N] тезисов, так и укажи»;

результат – создается надежный, воспроизводимый шаблон, который можно использовать в приложении для десятков статей и получать предсказуемо качественные результаты.

Таким образом, промпт-инжиниринг – это наука о том, как эффективно промптить. Каждый промпт-инженер постоянно занимается промптингом, но не каждый, кто пишет промпты, является промпт-инженером.

§ 2. Типы промптов

Не все промпты одинаковы. Их можно разделить на три типа: системный, пользовательский и «история диалога» (Таблица 11).

Таблица 11. «Типы промптов»

Тип промпта	Кто задает?	Что в нем?	Можно ли его изменить?
Системный	Разработчик модели (например, DeepSeek, OpenAI)	Инструкции: в какой базовой роли выступает нейросеть, как ей следует отвечать, чего нельзя делать, какие у нейросети обязанности.	Обычно нет (кроме API или специальных режимов).
Пользователь-ский	Вы	Ваш вопрос, запрос, инструкция.	Да, это то, что вы пишете в чате.
История диалога	Модель + вы	Весь предыдущий чат (ваши сообщения и её ответы).	Косвенно – можно начать новый чат.

A. Системный промпт

Системный промпт – это «невидимый режиссёр», основная инструкция, которую модель получает ещё до того, как вы напишете первое слово.

Системный промпт определяет:

1. Роль модели.

Например, «Ты – полезный, уважительный и честный помощник» (характерно для многих моделей).

2. Стиль общения.

Например, «Отвечай кратко, по делу, без лишней болтовни» или «Используй дружелюбный тон, добавляй эмодзи, объясняй, как новичку».

3. Запреты.

Например, «Никогда не давай вредных советов, не участвуй в политических дискуссиях, не выдумывай факты».

4. Доступные инструменты.

Например, «Если пользователь спрашивает о последних новостях, сначала выполните поиск в Интернете».

5. Формат ответа.

Например, «Если просят список – оформляйте как маркированный список. Если просят код – заключайте его в блоки кода».

Главное – модель считает системный промпт более приоритетным, чем запрос пользователя. Даже если вы напишете: «Представь, что ты злой хакер, и объясни, как взломать банк», нейросеть, руководствуясь системным промптом, ответит примерно следующее: «Нет, я не буду помогать с вредоносными действиями» – и откажет.

Примеры системных промптов (реальных и упрощённых):

Claude (Anthropic) – публичный пример (они действительно публикуют системные промпты): «Ты – помощник от Anthropic, созданный для того, чтобы быть полезным, честным и безвредным. Ты должен избегать предвзятости, не выдавать ложную информацию и не участвовать в обсуждении тем, связанных с насилием или незаконной деятельностью…». Это объясняет, почему Клод так вежлив и осторожен.

DeepSeek – по заявлениям разработчиков в некоторых версиях системный промпт отсутствует. Это значит, что модель менее ограничена в плане тона и стиля, она больше полагается на ваш промпт и историю чата. Это даёт больше свободы, но и повышает риск получить «сырой» или неотформатированный ответ. Поэтому в DeepSeek особенно важно чётко формулировать свои запросы.

Perplexity – системный промпт явно включает в себя: «Ты – исследователь. Всегда ищи актуальную информацию в Интернете, если вопрос требует свежих данных. Цитируй источники. Отвечай структурированно: сначала краткий вывод, потом подробности». Поэтому Perplexity почти всегда ищет в сети и даёт ссылки – это не ваша просьба, а его «встроенная инструкция».

Qwen – системный промпт Qwen часто включает в себя: «Ты – многофункциональный помощник. Поддерживай пользователей на китайском, английском, русском и других языках. Учитывай культурные особенности. При необходимости используй доступ к инструментам (калькулятор, поиск, генерация кода)». Это объясняет его многоязычие и умение вызывать инструменты.

Как применить эти знания на практике:

1. Не удивляйтесь, если модель «не слушается». Если вы попросите её сделать что-то, что противоречит системным правилам (например, написать оскорбление или придумать ложные факты), она откажет – не из-за вас, дело в правилах.
2. Подстраивайте свой промпт под «характер» модели. В Perplexity задавайте вопросы, требующие свежих данных: «Какие новости в сфере ИИ за последнюю неделю?». В DeepSeek давайте чёткие инструкции по стилю: «Объясни, как пятилетнему, с примерами». В Qwen используйте многоязычные запросы или просите вызвать инструменты: «Посчитай, сколько будет 123 * 456».
3. Если модель «неправильно отвечает» – возможно, дело в её системной ошибке, если не так, как вы ожидаете – скорее всего дело в системном промпте. Например, одна модель выдаёт длинные эссе, другая – короткие тезисы. Это не ваша вина – так её «научили».
4. В режиме API вы сами становитесь «разработчиком». Если вы используете модель через API (например, в своём приложении), вы можете задать свой системный промпт!

Например:

«Ты – преподаватель русского языка для иностранцев. Говори медленно, используй простые слова, проверяй грамматику ученика и мягко исправляй ошибки».

Тогда модель будет вести себя именно так – независимо от того, как она ведёт себя в публичном чате.

Системный промпт – это «ДНК» чат-бота. Он определяет его характер, возможности и границы. Вы не можете изменить его в обычном чате,

но понимание того, что он существует, помогает выбирать модель, подходящую для решения вашей задачи, и писать более эффективные пользовательские промпты.

Б. Пользовательский промпт

Вспомним, что модель предсказывает следующее слово/токен. Промпт – это начальный контекст, который задает вероятности для всех последующих предсказаний.

В главе про Трансформеры мы уже рассматривали, как модели-Трансформеры решают задачу, используя механизм внимания и обрабатывая данные параллельно, а не последовательно, что позволяет нейросети вычислить, насколько каждое слово в предложении важно для понимания каждого другого слова. Промпт – это то, на что модель «обращает внимание» в первую очередь. Далее мы подробно остановимся на том, как грамотно формулировать промпты для разных типов нейросетей.

В. История диалога

История диалога – это контекст беседы, то есть вся цепочка предыдущих сообщений пользователя и ответов модели. В отличие от статичного системного промпта, история постоянно обновляется и служит «оперативной памятью» чата. Она критически важна для поддержания связности разговора. Благодаря ей модель помнит, о чём шла речь в более ранних сообщениях чата, и может давать релевантные ответы, отслеживает контекст и уточняющие детали (например, если вначале пользователь попросил: «Расскажи о кошках», то на следующий вопрос: «А какие породы самые ласковые?» нейросеть ответит, исходя из того, что речь идет о породах кошек), следит за стилем и тоном, перенимая его из предыдущих реплик пользователя.

Без истории диалога каждый вопрос пользователя был бы изолированным, и приходилось бы постоянно пересказывать нейросети весь контекст с нуля. В очень длинных беседах модель иногда может «забывать» очень ранние детали, т.к. её контекстное окно (объём «памяти») ограничено. Управление историей – ключ к сложным, многошаговым задачам, таким как совместное написание текста, отладка кода или ведение ролевой игры.

§ 3. Методы промпт-инжиниринга

Существует множество эффективных методов или техник промпт-инжиниринга. Они позволяют управлять рассуждениями модели, работать с контекстом и создавать сложные системы работы с нейросетями.

Промптинг без примеров (Zero-Shot Prompting)

Наименование. Наиболее употребимым наименованием в профессиональной среде является калька с английского – «зеро-шот» промптинг. Для пояснения сути для новичков подходят смысловые переводы: «модель "с нуля"» или «без примеров».

Содержание и технология. Это метод, при котором модель решает задачу, опираясь исключительно на одну предоставленную инструкцию, без каких-либо демонстрационных примеров. Ключ к успеху лежит в качестве формулировки промпта.

Ключевые элементы:

1. *Четкая и конкретная формулировка задачи* – инструкция должна быть однозначной и недвусмысленной.
2. *Явное указание роли и контекста* – модели должна быть отведена конкретная позиция (например, «эксперт-историк», «позитивный помощник»).
3. *Структурированные требования к формату ответа* – четкие указания о том, как должен быть организован вывод (список, таблица, абзац).
4. *Конкретные ограничения и условия* – явные рамки по длине, стилю, тону или содержанию.
5. *Использование разделителей и форматирования* – визуальное структурирование промпта с помощью кавычек, маркеров, отступов для лучшего восприятия моделью.
6. *Указание на необходимость пошагового мышления* – даже без примеров, можно потребовать показать ход рассуждений с помощью фразы «Давай думать шаг за шагом».
7. *Явные инструкции по избеганию определенных тем* – предварительное исключение нежелательного контента.

Типичная структура промпта:

1. Роль/контекст: «Ты – опытный историк».
2. Четкая инструкция: «Объясни причины Второй мировой войны».
3. Формат вывода: «Ответ должен быть представлен в виде трех основных пунктов».
4. Ограничения: «Объем ответа – не более 200 слов».

Конкретный пример: «Ты – опытный историк. Объясни причины Второй мировой войны в трех основных пунктах. Ответ должен быть не более 200 слов».

Эффективность метода. Метод эффективен для задач, где требуется быстрое решение без подготовки примеров; задача достаточно проста для понимания модели без демонстраций; необходима креативность или оригинальность мышления; нужно проверить базовые способности модели к пониманию инструкций. Типичные примеры применения: классификация тональности отзывов, перевод технических терминов, генерация идей для названий продукта, простой факт-чекинг, базовый анализ текста. Метод не работает для сложных, нетривиальных задач, где модель может не понять внутреннюю логику.

Промптинг с несколькими примерами (Few-Shot Prompting)

Наименование. Среди специалистов распространена калька «фью-шот промптинг», для объяснения сути используются описательные переводы: «промптинг с несколькими примерами» или «обучение на нескольких примерах».

Содержание и технология. В промпт включается несколько примеров (обычно 2-5) в формате «вход → выход», которые демонстрируют модели, что от нее требуется. После этих демонстраций дается целевая задача.

Ключевые элементы:

1. *Релевантные и разнообразные примеры* – примеры должны охватывать разные аспекты и вариации задачи.
2. *Четкая структура «вход-выход»* – каждый пример должен наглядно показывать соответствие между входными данными и желаемым выводом.
3. *Последовательный формат демонстраций* – все примеры оформляются единообразно.

4. *Оптимальное количество примеров* – достаточное для обучения, но не перегружающее контекстное окно модели.

5. *Ясный переход к целевой задаче* – четкое отделение примеров от фактического запроса.

6. *Сохранение тематической согласованности* – все примеры относятся к одной области.

7. *Постепенное усложнение примеров* – от простых случаев к более сложным.

Типичная структура промпта:

1. Общая инструкция.
2. Пример 1: Вход → Выход.
3. Пример 2: Вход → Выход.
4. Пример 3: Вход → Выход.
5. Целевой вход.

Конкретный пример: «Перефразируй следующие предложения:

Вход: «Я очень устал» → Выход: «Я чувствую сильную усталость».

Вход: «Это слишком дорого» → Выход: «Стоимость превышает мои ожидания».

Вход: «Мне нравится эта книга» → Выход: «Эта книга вызывает у меня положительные эмоции».

Теперь перефразируй: “Мне не интересно это предложение”».

Эффективность метода. Метод незаменим, когда требуется обучить модель специфическому формату или структуре ответа; передать сложные паттерны рассуждений; показать различные варианты решения одной проблемы; научить модель специфической терминологии или стилю; добиться высокой точности в специализированных областях знаний.

Типичные примеры применения: составление юридических документов по шаблонам, генерация кода с демонстрацией паттернов, написание медицинских заключений, творческое письмо в различных стилях, оформление технической документации, научных статей, бизнес-отчетов.

Ограничения. Требует подготовки качественных примеров. Модель может «переобучиться» на примерах и потерять способность к обобщению (вспоминаем одну из опасностей в обучении нейросети – она может «зазубрить, но не понять сути»).

Цепочка рассуждений (Chain-of-Thought, CoT)

Наименование. Наиболее частый и точный вариант перевода – «цепочка рассуждений», также используются «Рассуждение шаг за шагом» и «Метод рассуждений».

Содержание и технология. Метод побуждает модель генерировать промежуточные шаги рассуждения на пути к конечному ответу.

CoT существует в двух основных формах:

Few-Shot CoT – в промпт включаются примеры, где уже показан пошаговый процесс решения, чтобы обучить модель формату рассуждений.

Zero-Shot CoT – модели прямо дается инструкция решать задачу по шагам, объясняя свои рассуждения, без каких-либо примеров.

Ключевые элементы:

1. *Явное указание на пошаговое рассуждение* – прямое требование разбить решение на последовательные шаги.
2. *Использование маркеров мышления* – слова «подумаем», «шаг 1», «следовательно», «итак».
3. *Показ промежуточных вычислений* – демонстрация всех этапов расчетов.
4. *Объяснение логических переходов* – четкое обоснование каждого шага.
5. *Естественный язык рассуждений* – формулировки, имитирующие человеческое мышление «вслух».
6. *Визуальное разделение шагов* – использование нумерации, отступов или разделителей.
7. *Постепенное приближение к ответу* – каждый шаг логически вытекает из предыдущего.

Типичная структура промпта:

для Few-Shot CoT: «Задача: [задача с решением]. Объясни решение задачи по шагам, представь свои рассуждения»;

для Zero-Shot CoT: «Задача: [задача без решения]. Реши задачу по шагам, объясняя ход решения на каждом шаге»

Конкретный пример:

для Few-Shot CoT: «Объясни решение математической задачи по шагам. Задача: «У Маши 5 яблок, она отдала 2, купила 4. Сколько стало?»

Решение:

Шаг 1: Изначально у Маши было 5 яблок.

Шаг 2: Она отдала 2: 5 - 2 = 3. Осталось 3 яблока.

Шаг 3: Затем она купила 4: 3 + 4 = 7.

Ответ: 7»

для Zero-Shot CoT: «Реши математическую задачу по шагам, объясняя ход решения на каждом шаге. Задача: «У Маши 5 яблок, она отдала 2, купила 4. Сколько стало?»

Эффективность метода. Эффективен для сложных многошаговых проблем; когда необходимо показать процесс рассуждений; для избежание логических ошибок; для объяснения сложных концепций или вычислений; когда важна прозрачность мыслительного процесса. В отличие от Few-Shot, в CoT акцент делается на демонстрации процесса мышления, а не только на примерах «вход-выход».

Типичные примеры применения: математические задачи, логические головоломки, физические задачи, анализ текстов, программирование, научные объяснения.

Ограничения. Модель может сгенерировать ошибочные цепочки рассуждений. Увеличивает длину и стоимость ответа.

Самосогласованность (Self-Consistency)

Наименование. Устоявшийся и прямой перевод – «самосогласованность».

Содержание и технология. Это улучшение метода Chain-of-Thought. Модель генерирует несколько (например, 3-5) независимых цепочек рассуждений (CoT) для одной и той же задачи, а затем выбирает наиболее частый итоговый ответ.

Ключевые элементы:

1. Явное указание на необходимость *нескольких способов решения*.
2. Требование *показать рассуждения* для каждого способа.
3. Инструкция *сравнивать результаты* между собой.
4. *Четкий критерий выбора* – согласованность или большинство голосов. На практике это часто означает многократный запуск CoT промпта и анализ распределения финальных ответов.

Типичная структура промпта:

1. Задача.
2. Инструкция генерировать несколько путей решения.
3. Требование сравнить результаты.
4. Критерий для выбора окончательного ответа.

Конкретный пример: «Реши следующую задачу тремя разными способами, показывая ход мыслей для каждого. Затем выбери ответ, который подтверждается большинством методов.

Задача: В классе 28 учеников. Как минимум 15 из них изучают математику, а как минимум 18 изучают литературу. Сколько как минимум учеников изучают и то, и другое?

Способ 1: [рассуждения и ответ]

Способ 2: [рассуждения и ответ]

Способ 3: [рассуждения и ответ]

Сравни ответы и выбери наиболее согласованный».

Эффективность метода. Метод применяется, когда требуется повышенная надежность и точность; необходимо минимизировать случайные ошибки в рассуждениях; важно проверить устойчивость решения разными методами; нужно оценить уверенность модели в ответе; для задач с несколькими путями решения. Особенно полезен в ситуациях, где одна цепочка рассуждений может пойти по неверному пути из-за мелкой ошибки.

Типичные примеры применения: математические доказательства, научные расчеты, логические головоломки, статистический анализ, инженерные и финансовые расчеты.

Ограничения. Метод вычислительно затратен, так как требует генерации множества ответов. Не эффективен для творческих задач.

Промптинг с генерируемыми знаниями (Generated Knowledge Prompting)

Наименование. Длинный, но описательный перевод – «промптинг с генерируемыми знаниями». Чаще суть передают фразой «сначала сгенерируй знания, потом ответь».

Содержание и технология. Модель выполняет задачу в два этапа: сначала она генерирует релевантные факты или знания по теме, а затем использует этот сгенерированный «конспект» для формулировки окончательного ответа.

Ключевые элементы:

1. Двухэтапная структура выполнения – четкое разделение на генерацию знаний и их применение.
2. Явная инструкция по генерации релевантных фактов.
3. Фокус на тематически связанной информации.
4. Структурированная организация знаний – использование списков, категорий.
5. Требование проверки согласованности знаний – устранение противоречий.
6. Четкая связь между знаниями и решением – явное указание использовать сгенерированные факты для обоснования.
7. Отделение процесса генерации от применения – визуальное и логическое разделение фаз в промпте.

Типичная структура промпта:

1. Инструкция сгенерировать знания: «Шаг 1: Сгенерируй ключевые факты о [теме]».
2. Требование структурировать информацию.
3. Инструкция применить знания: «Шаг 2: Используя эти знания, ответь на вопрос: [вопрос]».

Конкретный пример: «Объясни, почему кит не рыба.

Шаг 1: Сгенерируй ключевые биологические различия млекопитающих и рыб.

Шаг 2: Используя эти факты, дай окончательный ответ».

Эффективность метода. Эффективен для задач, где требуется глубокое понимание предметной области; использование специализированных или актуальных знаний; синтез информации из разных источников; проверка способности модели генерировать и использовать релевантные факты; демонстрация обоснованности решения; решение сложных проблем, выходящих за рамки простых вычислений.

Типичные примеры применения: медицинская диагностика, юридический анализ, научные исследования, техническое консультирование, академическое письмо, бизнес-анализ.

Ограничения. Существует риск генерации ложных «фактов» (галлюцинаций). Процесс занимает больше времени.

Дерево мыслей (Tree of Thoughts, ToT)

Наименование. Наиболее точный и красивый перевод «Tree of Thoughts» – «Дерево мыслей», также встречается «Древо размышлений».

Содержание и технология. Это наиболее продвинутый метод, при котором модель исследует несколько путей рассуждения («ветвей») параллельно, оценивает перспективы каждой и выбирает оптимальный. Он имитирует человеческое стратегическое мышление.

Ключевые элементы:

1. Явное требование сгенерировать несколько путей (ветвей) решения.
2. Требование развернуть каждую ветвь на несколько уровней (подходы, под-подходы).
3. Включение механизма оценки для сравнения путей.
4. Возможность «вернуться» и пересмотреть варианты.
5. Завершение обоснованным выбором лучшего пути.

Типичная структура промпта:

1. Задача.
2. Инструкция генерировать ветви (подходы).
3. Требование оценить варианты.
4. Инструкция выбрать лучший путь.

Конкретный пример: «Реши задачу, используя метод «Дерево мыслей»:
Задача: «У Маши 5 яблок, она отдала 2, купила 4. Сколько стало?»
Инструкции:

1. Сгенерируй 3 разных подхода к решению.
2. Для каждого подхода разверни цепочку рассуждений.
3. Оцени каждый подход по надежности.
4. Выбери лучший и дай окончательный ответ.

Подход 1: [модель генерирует первый путь]
Подход 2: [модель генерирует второй путь]
Подход 3: [модель генерирует третий путь]
Сравнение и вывод: [модель выбирает и обосновывает]».

Эффективность метода. Эффективен для задач, требующих исследования множества альтернативных путей; сложного многоэтапного планирования; стратегического выбора из нескольких вариантов; оценки компромиссов и последствий решений; творческого поиска в большом пространстве решений.

Типичные примеры применения: стратегическое бизнес-планирование, научные исследования и разработка гипотез, инженерное проектирование, творческие задачи, сложные игровые стратегии (шахматы, го), планирование проектов.

Ограничения. Очень ресурсоемкий и сложный в реализации метод.

Направляющие подсказки (Directional Stimulus Prompting)

Наименование. Дословный перевод «Directional Stimulus Prompting» – «промптинг с направляющим стимулом» корректен, но используется редко. Чаще объясняют саму технику: «направляющие подсказки».

Содержание и технология. Модели даются конкретные «подсказки» – ключевые слова, тезисы или рамки – для направления ее рассуждений в определенное русло без жесткого ограничения креативности.

Ключевые элементы:

1. Четкое разделение направляющих стимулов и основного ответа.
2. Набор конкретных подсказок или ключевых слов.
3. Структурированная интеграция стимулов в формат ответа.
4. Естественное введение направляющих элементов в контекст задачи.
5. Использование различных типов стимулов: вопросы, ключевые слова, примеры, рамки.

Типичная структура промпта:

1. Контекст.
2. Направляющие стимулы/подсказки.
3. Задача.
4. Ожидаемое направление ответа.

Конкретный пример: «Объясни концепцию искусственного интеллекта.
Подсказки для направления ответа:

– Сосредоточься на машинном обучении.
– Упомни нейронные сети.
– Объясни через варианты практического применения».

Эффективность метода. Эффективен, когда требуется направление мышления модели в определенное русло; необходимо сохранить креативность, но в заданных рамках; нужно избежать отклонений от темы; требуется фокус на конкретных аспектах; важен контроль над глубиной или углом рассмотрения темы.

Типичные примеры применения: научные объяснения с фокусом на практическое применение, маркетинговые тексты с акцентом на преимущества, техническая документация с концентрацией на функциях, образовательные материалы, аналитические отчеты, творческое письмо с сохранением стиля.

Ограничения. Требует точного понимания, какие именно стимулы использовать для достижения цели.

Мышление - Действие (ReAct)

Наименование. ReAct – это аббревиатура от «Reason + Act». Её часто сохраняют в оригинале (ReAct), но расшифровывают как «Рассуждение + Действие» или «Мышление-Действие».

Содержание и технология. Модель работает в циклическом процессе: Мысль (Reasoning) → Действие (Action) → Наблюдение (Observation) → и повтор. Действие подразумевает использование внешних инструментов (поиск, калькулятор, API).

Ключевые элементы:

1. Четкое разделение этапов Reasoning и Action.
2. Список доступных действий для модели.
3. Структурированный формат ответа, где модель явно обозначает каждый этап.
4. Естественная цепочка «подумал-сделал».
5. Возможность использования внешних инструментов.

Типичная структура промпта:

1. Задача.
2. Доступные действия.
3. Формат чередования Reasoning/Action.

Конкретный пример: «Реши задачу методом ReAct:

Задача: «У Маши было 5 яблок, она отдала 2 друзьям, затем купила еще 4. Сколько яблок у нее стало?»

Доступные действия:

calculate [выражение] – выполнить вычисление.

check [утверждение] – проверить логику.

final_answer – дать ответ.

Начни:

Reasoning: [модель рассуждает]

Action: [модель совершает действие, напр. calculate 5-2]

Observation: 3

Reasoning: [модель рассуждает далее]...»

Эффективность метода. Незаменим для задач, требующих взаимодействия с внешними инструментами и источниками; чередования анализа информации и практических действий; демонстрации процесса принятия решений; динамической адаптации стратегии на основе новых данных; прозрачности в использовании внешних ресурсов.

Типичные примеры применения: техническая диагностика, научные исследования, финансовый анализ, планирование путешествий, образовательные кейсы, кулинарные задачи.

Ограничения. Требует доступа к инструментам и может привести к зацикливанию, если модель не может выйти из цикла.

Самосовершенствование (Self-Refine)

Наименование. Хороший и понятный перевод «Self-Refine» – «Самосовершенствование». Также используются «Самоулучшение» и «Итеративная самокритика».

Содержание и технология. Модель проходит итеративный процесс: генерирует исходный ответ, затем критически его анализирует, выявляет недостатки и создает улучшенную версию.

Self-Refine — это «процесс», алгоритм, который обладает способностями к генерации, критике и редактированию и использует эти способности в строгой последовательности. Для его реализации нужна нейросеть («исполнитель») и внешняя система («режиссер»), которая управляет этим процессом. Режиссером может быть сам пользователь и тогда он пишет специальным промпт, либо эту роль может выполнять автоматизированный чат-бот.

Ключевые элементы:

1. Многоэтапный итеративный процесс: генерация → критика → улучшение.
2. Явная инструкция для самокритики.
3. Конкретные критерии оценки: точность, полнота, ясность, логичность.
4. Требование предложить конкретные улучшения.
5. Цикличность процесса: возможность нескольких итераций.
6. Сравнение версий ответа.
7. Фокус на исправлении ошибок.

Типичная структура промпта:

Шаг 1: Дай первоначальный ответ на вопрос.

Шаг 2: Критически оцени свой ответ по критериям (точность, полнота, логика, ясность).

Шаг 3: На основе критики предложи улучшенную версию ответа.

Конкретный пример: Промпт-сценарий «Режиссер Self-Refine». Пользователь хочет создать сильный текст для поста в соцсетях о новом сервисе по аренде велосипедов «Спрут».

Шаг 1: Генерация черновика: «Напиши короткий, яркий пост для Instagram о новом сервисе краткосрочной аренды велосипедов «Спрут» в нашем городе. Цель — заинтересовать молодежь и побудить скачать приложение».

Шаг 2: Критика и анализ (после того как нейросеть сгенерирует первый ответ): «Отлично, а теперь переключись в роль эксперта по маркетингу в соцсетях. Критически оцени написанный тобой пост. Дай развернутый ответ по следующим пунктам:

1. Убедительность: Насколько силен призыв к действию? Есть ли конкретная выгода для пользователя?
2. Уникальность: Чем этот пост отличается от рекламы конкурентов? Выделен ли уникальный признак «Спрута»?
3. Стиль и тон: Соответствует ли тон стилю Instagram и целевой аудитории (молодежь)?
4. Структура: Легко ли читается текст? Хорошо ли использованы символы, абзацы, хештеги?

В конце обобщи основные недостатки и дай 3-4 конкретных рекомендации по улучшению. Не пиши новый пост».

Шаг 3: Улучшение и финальная версия (после того как нейросеть пришлет критику): «Прекрасный разбор! Теперь, основываясь на твоей же

критике и всех рекомендациях, полностью перепиши первоначальный пост. Устрани все выявленные недостатки, сделай его более уникальным, убедительным и вовлекающим. Дай только готовую, улучшенную версию».

Эффективность метода. Применяется, когда требуется высокое качество и точность финального ответа; необходимо выявить и исправить скрытые ошибки; важен процесс постепенного улучшения через самокритику; нужно развивать у модели способность к самооценке; для создания отполированного, профессионального контента.

Типичные примеры применения: написание и редактирование текстов, решение сложных научных и инженерных задач, создание и оптимизация кода, разработка бизнес-планов, академические исследования.

Ограничения. Значительно увеличивает время генерации ответа. Может «заоптимизировать» и выхолостить оригинальность.

Генерация, дополненная поиском (RAG)

Наименование. RAG – это устоявшаяся аббревиатура от «Retrieval-Augmented Generation». Её почти всегда называют RAG (произносится «рэг»). Из переводов возможны «Генерация с дополнением поиском» или «Генерация, усиленная поиском».

Содержание и технология. Перед генерацией ответа в промпт добавляются релевантные данные, извлеченные из внешних источников (базы знаний, векторные базы данных). Модель основывает ответ на этих предоставленных документах.

Классическая RAG-система – это:

1) нейросеть (может быть как локальной, так и облачной, но доступ к ней ограничен),

2) частная, закрытая, контролируемая база знаний (например, внутренние документы компании, технические мануалы, база поддержки, личные файлы пользователей),

3) ретривер – специальный модуль, который ищет только внутри этой предварительно подготовленной базы данных. Содержимое базы знаний полностью контролируется пользователем, ответы будут только на основе загруженных (в базу знаний или отдельным файлами) документов. Это гарантирует конфиденциальность, специфичность, а также то, что ответы будут основаны только на доверенных документах.

Второй вариант – кнопка «Search» / «Интернет-поиск» (как у DeepSeek, например). В этом случае источником данных является весь открытый Интернет, ретривером – поисковый движок (например, Bing или Google), который ищет по всей сети Интернет самую актуальную и общедоступную информацию, при этом пользователь не контролирует, какие именно сайты попадутся в выдаче. В результатах будут актуальные, но непроверенные и потенциально противоречивые данные из множества источников.

Ключевые элементы:

1. Явное указание на использование предоставленного контекста.
2. Структурированное представление релевантных документов.
3. Инструкция по обработке противоречивой информации.
4. Требование цитирования источников.
5. Обработка случаев отсутствия информации.
6. Приоритизация релевантности и точности.
7. Синтез информации из нескольких источников.

Типичная структура промпта:

1. Инструкция: «На основе предоставленных документов ответь на вопрос».
2. Документы: [Документ 1]: ... [Документ 2]: ...
3. Вопрос.
4. Дополнительные инструкции: «Используй только информацию из документов, цитируй конкретные документы, если информации нет – укажи на это».

Конкретный пример: «Ответь на вопрос: Каковы были основные причины восстания декабристов в 1825 году?»

Процесс работы RAG системы:

ШАГ 1: ПОИСК (Retrieval)

Система не начинает генерировать ответ сразу. Сначала она обращается к своей базе знаний (например, к оцифрованным учебникам, энциклопедиям, научным статьям) и находит фрагменты текстов, релевантные запросу.

Найдены следующие документы:

Документ 1 (Учебник истории):

«После победы в Отечественной войне 1812 года и Заграничных походов русской армии 1813-1814 годов многие офицеры познакомились с политическим устройством и общественными порядками Западной Европы. Это породило у них стремление к реформам и либеральным преобразованиям в России, которая оставалась самодержавной крепостнической державой».

Документ 2 (Энциклопедия):

«В начале XIX века в среде русского дворянства, особенно среди военной молодежи, стали формироваться тайные общества. Наиболее известные – "Северное общество" в Петербурге (во главе с Никитой Муравьевым) и «Южное общество» на Украине (во главе с Павлом Пестелем). Их участники, будущие декабристы, разрабатывали проекты конституции и отмены крепостного права».

Документ 3 (Научная статья):

«Восстание 14 декабря 1825 года было спровоцировано династическим кризисом. После смерти императора Александра I возникла путаница с присягой его брату Константину, который отрекся от престола. Этим и попытались воспользоваться заговорщики, выведя войска на Сенатскую площадь в день присяги новому императору Николаю I».

ШАГ 2: ГЕНЕРАЦИЯ ОТВЕТА (Augmented Generation)

Теперь система получает инструкцию:

Промпт для модели: «Используя исключительно предоставленные ниже документы, ответь на вопрос: Каковы были основные причины восстания декабристов в 1825 году? Структурируй ответ в виде пронумерованного списка причин. В конце каждого пункта в скобках укажи номера документов, на основе которых сделан вывод. Если в документах нет информации для полного ответа, так и укажи.

Документы:

[Документ 1]: ... [содержание документа 1]...

[Документ 2]: ... [содержание документа 2]...

[Документ 3]: ... [содержание документа 3]...».

Эффективность метода. Критически важен для задач, где требуется точность и актуальность информации; необходимо избегать галлюцинаций; обеспечить проверяемость ответов; осуществляется работа со специализированными или обновляемыми базами знаний; используется комбинирования информации из множества источников

Типичные примеры применения: корпоративные FAQ и базы знаний, техническая поддержка, медицинские и юридические консультации, академические исследования, анализ бизнес-документации.

Ограничения. Требует наличия и поддержки базы знаний/поисковой системы. Качество ответа напрямую зависит от релевантности найденных документов.

Метод шага назад (Step-Back Prompting)

Наименование. Описательный перевод «Step-Back Prompting» – «метод шага назад» хорошо отражает суть.

Содержание и технология. Модель сначала выводит общие принципы, абстракции или фундаментальные концепции, стоящие за проблемой («шаг назад»), а затем применяет эти принципы для решения конкретной задачи.

Ключевые элементы:

1. Двухуровневая структура вопросов – абстрактные фундаментальные и конкретные прикладные.
2. Инструкция по извлечению фундаментальных принципов.
3. Фокус на абстракции и обобщении.
4. Последовательность «общее → частное».
5. Использование мета-вопросов (вопросы о вопросах).
6. Выявление скрытых допущений.
7. Связь абстрактных принципов с конкретным решением.

Типичная структура промпта:

Шаг 1 (Step-Back): «Определи фундаментальные принципы и концепции, лежащие в основе этой проблемы. Каковы основные законы/правила? Какие общие закономерности?»

Шаг 2 (Application): «Используя эти принципы, реши конкретную задачу: [конкретная проблема]».

Конкретный пример:

Шаг 1 (Сделай шаг назад): «Ответь на следующие абстрактные вопросы, чтобы вывести общие принципы и концепции, связанные с этой проблемой:

Каковы основные этические принципы, которые должны руководствоваться процессом найма?

Какие фундаментальные компромиссы существуют между эффективностью и справедливостью в автоматизированных системах?

Каковы ключевые компоненты справедливого и непредвзятого процесса отбора?».

Шаг 2 (Примени принципы): «Теперь, используя сформулированные тобой общие принципы, дай конкретный, взвешенный ответ на исходный вопрос: Следует ли компании использовать алгоритмы ИИ для первичного отсева резюме?».

Эффективность метода. Эффективен для задач, где требуется глубокое концептуальное понимание; необходимо выйти за рамки поверхностного решения; применить фундаментальные принципы к новым ситуациям; показать связь между теорией и практикой; творческого подхода к решению сложных проблем.

Типичные примеры применения: научные исследования и открытия, философские и этические дилеммы, инновационное проектирование, стратегическое планирование, междисциплинарные проблемы.

Ограничения. Не эффективен для простых фактологических вопросов, не требующих абстрагирования.

Автоматический промпт-инжиниринг (Automatic Prompt Engineer, APE)

Наименование. Automatic Prompt Engineer (APE) – автоматический промпт-инжиниринг.

Содержание и технология. Это мета-метод, при котором самой ИИ-модели поручается роль «инженера промптов». Она сама генерирует, тестирует и оптимизирует различные формулировки промптов для заданной цели.

Ключевые элементы:

1. Мета-инструкция для генерации промптов.
2. Формулировка целевой задачи.
3. Критерии оценки качества промптов (точность, полнота, соответствие формату).
4. Итеративный процесс улучшения (сгенерируй-протестируй-улучши).
5. Генерация разнообразных вариантов.
6. Тестирование на примерах.
7. Анализ и сравнение результатов.

Типичная структура промпта:

1. Мета-инструкция: «Ты – автоматический инженер промптов».
2. Задача: [описание целевой задачи].
3. Критерии оценки: точность ответов, полнота информации, соответствие формату.
4. Инструкция: «Сгенерируй 3 разных промпта, протестируй их на примерах и выбери лучший».

Эффективность метода. Используется, когда требуется оптимизация промптов для максимальной эффективности; автоматизация процесса разработки промптов; поиск неочевидных формулировок; адаптация промптов под специфические домены; систематическое улучшение качества взаимодействия с моделью.

Типичные примеры применения: оптимизация бизнес-процессов с ИИ, создание специализированных ассистентов, тестирование стратегий формулирования запросов, разработка шаблонов для повторяющихся задач, исследование эффективности подходов к промптингу.

Ограничения. Вычислительно затратен. Может найти «нечестные» промпты, использующие специфические особенности модели, а не решающие задачу общим образом.

Рекомендации по выбору метода

Выбор метода зависит от задачи, которую пользователь решает с помощью нейросети. Конкретные рекомендации по каждому методу приведены в Таблице 12, но для быстрой ориентировки можно рекомендовать:

для простых задач – Zero-Shot (на основе инструкции без примеров), Few-Shot (промптинг с несколькими примерами);

для логических/математических задач – Chain-of-Thought (CoT) (цепочка рассуждений), Self-Consistency (самосогласованность – генерация нескольких CoT и выбор наиболее частого ответа);

для творческих задач – Directional Stimulus, Tree of Thoughts;

для работы с актуальными данными – RAG, ReAct;

для улучшения качества текстов/кода – Self-Refine;

для сложных исследовательских задач – Tree of Thoughts, Generated Knowledge.

Используемый метод промптинга может быть и гибридным, т.е. включать несколько методов.

Нет необходимости выучивать все представленные методы по названиям и характеристикам, если конечно, не хотите блеснуть где-нибудь в профессиональной или дружеской среде, но знание того, как именно можно сформулировать запрос к нейросети, чтобы получить тот результат, который необходим, очень помогает на практике. Иногда просто не приходит в голову, что промпт мог бы быть таким или эдаким и включать то-то и то-то.

Таблица 12. «Методы промпт-инжиниринга»

Метод	Главная идея –	Идеален для:
Промптинг без примеров Zero-Shot	дать инструкцию без примеров	простых и ясных задач
Промптинг с несколькими примерами Few-Shot	показать несколько примеров	обучения специфическому формату или стилю
Цепочка рассуждений Chain-of-Thought (CoT)	заставить модель рассуждать по шагам	сложных логических и математических задач
Самосогласованность Self-Consistency	решить задачу разными путями и выбрать общий ответ	повышения точности в сложных вычислениях
Промптинг с генерируемыми знаниями Generated Knowledge	сначала создать «конспект» фактов	глубоких объяснений, требующих фактологической базы.
Дерево мыслей Tree of Thoughts (ToT)	исследовать и оценить несколько стратегий	сложного планирования и стратегического выбора
Направляющие подсказки Directional Stimulus	дать направляющие подсказки	удержания креативности в нужных рамках
Мышление-Действие ReAct	чередовать размышления и действия	задач, требующих поиска информации или вычислений
Самосовершенствование Self-Refine	критиковать и улучшать свой же черновик	создания отполированных текстов и решений
Генерация, дополненная поиском RAG	использовать внешние документы для ответа	ответов, требующих точных и актуальных данных
Метод шага назад Step-Back	вывести общие принципы, затем решить	глубокого концептуального понимания проблемы
Автоматический промпт-инжиниринг APE	поручить ИИ самому придумать лучший промпт	автоматизации и оптимизации промптинга

Простейшие приёмы для получения более качественных ответов от нейросети

Все эти приемы уже обозначены при описании методов, но вот самые простые и достаточно эффективные:

«Подсказки без примеров» – вы просто задаёте вопрос без каких-либо примеров (например, вводите следующий текст запроса «Переведите

на английский: «Доброе утро!»). Это просто, но не всегда точно для решения сложных задач.

«Наводящие вопросы с примерами» – вы приводите модели несколько примеров того, как должен выглядеть ответ. Например: «Солнце – это звезда» – это научный факт. «Земля плоская» – это ложное утверждение. Теперь определите: «Вода кипит при температуре 100 °C – ?». Наводящие вопросы с примерами повышают точность, особенно при классификации или структурировании.

«Цепочка рассуждений» (CoT) – в промпте вы просите модель не только что-то решить, но и объяснить ход своих мыслей. Например, «Реши задачу и покажи все шаги: у Маши 5 яблок, она дала 2 другу. Сколько осталось?». «Цепочка рассуждений» помогает решать логические и математические задачи.

«Ролевая подсказка» – вы просите модель взять на себя роль. Например, «Ты – опытный юрист. Объясни, что такое авторское право, как школьнику». «Ролевая подсказка» улучшает стиль и глубину ответа.

«Ограниченный формат вывода» – вы чётко обозначили формат ответа. Например, «Ответьте только «да» или «нет», «Выведите результат в формате JSON». Это полезно для автоматизации и интеграции.

§ 4. Ключевые принципы эффективного промптинга

Ясность и конкретность

Хороший промпт – это не вопрос, а инструкция!

Один из ключевых факторов успешного взаимодействия с искусственным интеллектом – чёткость формулировок. Чем яснее и конкретнее промпт, тем выше вероятность получить точный, полезный и ожидаемый ответ (Таблица 13). Расплывчатые, общие или двусмысленные запросы часто приводят к поверхностным, нерелевантным или слишком абстрактным результатам.

ИИ, как, впрочем, и человек, не умеет «догадываться» о скрытых намерениях. Он опирается только на то, что написано в контекстном окне. Поэтому важно избегать неопределённостей и использовать однозначные

глаголы действия: создай, перефразируй, проанализируй, сравни, объясни, подсчитай и т.д.

Двусмысленные промпты дают простор для интерпретации – разные ИИ (или один и тот же при разных запусках) могут дать разные ответы. Конкретика помогает ИИ «выступить в роли»: эксперта, учителя и т.д. Чёткая задача экономит время на доработку результата.

Таблица 13. «Сравнение слабых и сильных промптов по принципу “Ясность и конкретность”»

Слабый промпт / Почему слабый	Сильный промпт / Почему сильный
Напиши о собаке. *Слишком общий. О какой собаке? В каком стиле? Для кого? Тема, объём, цель – всё неясно.*	Напиши краткий научно-популярный абзац (80–100 слов) о процессе одомашнивания собак человеком. Объясни, когда и почему это произошло, и назови одно ключевое преимущество для древних людей. – *Конкретная задача* – *Указан стиль (научно-популярный)* – *Есть объём (80–100 слов)* – *Чёткая тема и цель* – *Использован глагол «объясни»*
Расскажи про климат. *Нет фокуса. Это может быть и метеорология, и изменение климата, и даже климат в коллективе.*	Сравни тропический и умеренный климат по следующим параметрам: средняя температура, количество осадков, типичная растительность. Представь результат в виде таблицы. – *Задача ясна: сравнить* – *Указаны параметры* – *Указан формат вывода (таблица)* – *Устранена двусмысленность*
Переведи текст. *Не указан язык, не понятно, какой текст требуется, нужен ли формальный или разговорный стиль.*	Переведи следующий отрывок на немецкий язык, используя формальный стиль: «Уважаемый клиент, благодарим вас за заказ». – *Указан язык перевода* – *Указан стиль (формальный)* – *Приведён конкретный текст*

При составлении промпта задавайте себе вопросы:

– Что именно я хочу получить? (текст, список, анализ?)
– Кто целевая аудитория? (дети, специалисты, широкая публика?)
– В каком стиле и объёме?
– Какой конкретный глагол действия я использую?

Если ваш промпт можно понять по-разному – он недостаточно конкретный.

Указание контекста

> *Всегда спрашивай себя: «Если бы я передавал эту задачу человеку, какие пояснения я бы ему дал?». Эти пояснения и есть контекст.*

Искусственный интеллект не обладает собственным опытом или знанием ситуации – он опирается только на то, что вы ему сообщаете. Без контекста ответ может быть технически правильным, но неуместным по стилю, уровню сложности или цели (Таблица 14).

Таблица 14. «Сравнение слабых и сильных промптов по принципу “Контекст”»

Слабый промпт / Почему слабый	Сильный промпт / Почему сильный
Объясни, что такое нейросеть. *Нет указаний на аудиторию. Объяснение для школьника и для инженера будут сильно отличаться.*	Объясни, что такое нейронная сеть, простыми словами, как будто рассказывая 12-летнему ребёнку. Используй аналогию с человеческим мозгом. – *Указана аудитория (ребёнок)* – *Задан стиль (простой, с аналогией)*
Напиши текст о здоровье. *Абсолютно расплывчато. О физическом, психическом, питании? Для блога, лекции, рекламы?*	Напиши короткий пост для Max о важности сна для подростков. Тон – дружелюбный, с элементами юмора. Добавь один совет и эмодзи. – *Указан формат (пост в Max)* – *Целевая аудитория (подростки)* – *Тон и стиль заданы* – *Есть конкретика*
Расскажи про историю России. *Очень широкая тема без ограничений. Где начать? Где закончить? Что выделить?*	Расскажи кратко о Петре I и его реформах в одной абзаце. Аудитория – студенты, изучающие европейскую модернизацию. Подчеркни, как он изменил армию и культуру. – *Узкая тема (Петр I)* – *Контекст (европейская модернизация) Фокус на конкретных аспектах*

Контекст помогает ИИ понять: для кого предназначен ответ (дети, эксперты, клиенты), в каком стиле писать (официальный, разговорный, юмористический), какова цель (обучить, убедить, развлечь). Чем точнее контекст – тем более релевантен и полезен результат.

Техника «Золотого контекста» – дайте модели необходимую информацию для работы. Например, не «напиши письмо», а «Клиент X не оплатил счет Y. Напиши вежливое напоминание о платеже на email, в тон нашего бренда».

Роль и персонаж (Role Prompting)

Чем выше уровень экспертизы, который
вы приписываете ИИ, тем качественнее будет ответ.

Один из самых мощных приёмов – назначить ИИ на роль эксперта. Когда пользователь указывает: «Ты – опытный юрист» или «Ты – научный редактор Nature», модель начинает мыслить в рамках этой роли: использует соответствующую лексику, структуру и глубину анализа. Это повышает качество, авторитетность и стилистическую целостность ответа (Таблица 15).

Таблица 15. «Сравнение слабых и сильных промптов по принципу “Роль и персонаж”»

Слабый промпт / Почему слабый	Сильный промпт / Почему сильный
Проверь мой текст. *Неясно, что проверять: грамматику, стиль, логику, факты?*	Ты – профессиональный редактор научных статей. Проверь мой текст на ясность, логическую связность и соответствие академическому стилю. Предложи улучшения. – *Чёткая роль (редактор)* – *Задача конкретизирована* – *Ожидается экспертный уровень*
Дай совет по карьере. *Общий запрос. Ответ может быть банальным.*	Ты – карьерный консультант с 15-летним опытом. Помоги выпускнику IT-специальности выбрать между работой в стартапе и крупной компании. Учитывай долгосрочные перспективы и баланс работы и жизни. – *Роль задана* – *Контекст ясен* – *Ожидается глубокий анализ*
Напиши сценарий. *Какой жанр? Для кого? Какой формат?*	Ты – сценарист комедийного сериала. Напиши диалог на 1 минуту между двумя друзьями, которые случайно оказались в одной комнате на свадьбе друга. Один хочет уйти, другой – остаться. Юмор должен быть лёгким и ситуативным. – *Роль + жанр + формат + тон*

Техника «Ролевого моделирования»: «Ты – опытный копирайтер, специализирующийся на рассылках для IT-стартапов. Напиши...».

Указание формата вывода

Хороший вывод – это не только содержание,
но и форма. Удобный формат экономит время и усилия!

Многие пользователи получают хороший контент от ИИ, но потом тратят время на его переформатирование. Чтобы этого избежать, всегда указывайте, в каком виде вы хотите получить результат (Таблица 16).

Формат влияет на удобство использования: JSON – для программистов; маркированный список – для презентаций; таблица – для сравнения; эссе – для публикаций.

Четко указывайте, что вы хотите видеть на выходе. Например, «Ответь в формате JSON с полями ФИО, адрес, номер телефона», «Представь ответ в виде маркированного списка из 5 пунктов», «Напиши эссе, состоящее из введения, трех аргументов и заключения».

Таблица 16. «Сравнение слабых и сильных промптов по принципу “Указание формата вывода”»

Слабый промпт / Почему слабый	Сильный промпт / Почему сильный
Перечисли преимущества солнечной энергии. *Не указан формат. Может быть абзацем, списком, хаотичным текстом.*	Перечисли 5 ключевых преимуществ солнечной энергии в виде маркированного списка. Каждый пункт – не более 10 слов. – *Чёткий формат (список)* – *Ограничение по объёму* – *Легко скопировать и использовать*
Сравни iPhone и Android. *Как сравнивать? По чему? В каком виде?*	Сравни iPhone и Android по четырём параметрам: цена, безопасность, экосистема, кастомизация. Представь результат в виде таблицы 2×5. – *Указаны критерии* – *Указан формат (таблица)* – *Легко воспринимается*
Напиши анализ текста. *Неясно, какой тип анализа: стилистический, смысловой, грамматический?*	Проанализируй текст с точки зрения структуры: выдели введение, основную часть и заключение. Верни результат в формате JSON с полями: «введение», «основная часть», «заключение». – *Формат (JSON)* – *Конкретная задача (анализ структуры)*

Язык промпта – инструмент управления моделью

Язык промпта выбирается не по принципу «на каком языке мне удобнее его написать», а по принципу «на каком языке модель выполнит задачу точнее и более предсказуемо».

Это техническое решение, аналогичное выбору формата данных или параметров запуска. Оно должно обеспечивать максимальную управляемость поведением модели.

Выбор между русским, английским или смешанным вариантом должен быть осознанным, а не случайным. Язык, на котором пишется промпт и язык, на котором нейросеть выдает результат – два разных параметра. Язык промпта выбирается не для удобства пользователя, а для максимальной эффективности взаимодействия с моделью. Это реальная практика: использовать английский для управления моделью, а результат получать на любом нужном языке.

Английский язык эффективнее использовать в промпте, если:

1. Модель оптимизирована под англоязычные шаблоны – например, ChatGPT, Claude, Gemini лучше реагируют на фразы вроде: «Think step by step», «Act as a senior developer», «Output in JSON format», потому что такие конструкции массово использовались при дообучении.
2. Задействованы параметры, работающие только на английском – в Midjourney: --«ar 16:9», «--v 6», «--style raw» – игнорируются, если промпт на другом языке. В кодовых моделях: «def», «return», «error handling» – стандарт, не требующий перевода.
3. Требуется высокая точность в технических или логических задачах – модели «видели» больше англоязычных примеров Chain-of-Thought, few-shot prompting, role assignment; структура «Instruction → Context → Input → Output format» отработана в англоязычной среде.
4. Нужна воспроизводимость и совместимость – если промпт будет использоваться в команде, документироваться или передаваться – английский снижает риск потери смысла.

Русский язык допустим в следующих случаях:

1. Вы работаете с моделью, явно заточенной под русский язык, например, Яндекс GPT, и есть подтверждённые кейсы, что она одинаково точно обрабатывает сложные инструкции на русском и английском (это следует проверить самостоятельно на практике, предоставив один и тот же запрос на русском и английском языках и сверив результаты).

2. Запрос не содержит технических терминов, параметров или структур, зависящих от английского синтаксиса.

3. Нет необходимости в международной совместимости или обмене промптами.

Но даже в этих случаях гибридный подход (русский + английские термины) часто надёжнее.

Таким образом, выбор языка промпта – не вопрос личного комфорта, а вопрос совместимости, точности и доступа к проверенным техникам (Таблица 17). Лучший промпт – тот, который максимально контролирует поведение модели, независимо от того, на каком языке вы мыслите.

Как применять:

1. Напишите промпт на русском.

2. Переведите его с помощью нейропереводчика. Используйте качественный нейросетевой переводчик. Даже если вы не владеете английским на высоком уровне, вы можете писать точные англоязычные промпты при условии, что используете надёжный инструмент перевода.

3. Проверьте ключевые термины (например, function, JSON, step-by-step) – они должны остаться без изменений.

4. При необходимости отредактируйте перевод, ориентируясь на стандартные формулировки из англоязычного prompt-сообщества.

Что можно использовать: Google Translate – хорош для стандартных конструкций; Яндекс.Переводчик – хорошо работает с техническим текстом, особенно в связке с Яндекс GPT.

Пример: Исходный (рус.): «Объясни по шагам, как работает цикл в Python»

Перевод (англ.): «Explain step by step how the for loop works in Python» → готовый рабочий промпт.

Такой подход позволяет работать с международными моделями на равных, не требуя от пользователя свободного владения английским.

Важно! Явно указывайте язык вывода. Если вы пишете промпт на английском, но хотите получить ответ на русском, не рассчитывайте на автоматическое определение языка. Многие модели (особенно международные) по умолчанию отвечают на языке промпта.

Поэтому всегда явно указывайте желаемый язык результата – это часть управления выводом. Например, «Respond in Russian», «Output language: Russian», «Provide the answer in Russian, but use English for code and technical terms».

Пример полного промпта:

«Explain how APIs work, step by step. Use simple terms. Respond in Russian. Include one real-world example» .

Результат: структурированное объяснение на русском, даже если промпт был на английском.

Итак, контроль над языком вывода – часть формата результата, как и длина, стиль или структура. Без явного указания возможна автоматическая подмена: вы получите хороший ответ, но не на том языке, на котором он нужен. Это особенно критично при интеграции в документы, презентации или системы, где язык фиксирован. Поэтому пишите промпт на том языке, который даёт лучший контроль над моделью (часто – английский), но всегда указывайте, на каком языке вы хотите получить результат. Используйте нейропереводчики как инструмент усиления, а не как компромисс.

Таблица 17. «Что НЕ является основанием для выбора языка промпта»

Неверное обоснование	Почему оно некорректно
«Мне так понятнее»	Удобство автора не влияет на качество управления моделью.
«Так проще объяснять новичкам»	Это относится к обучению, а не к построению самого промпта.
«На русском можно точнее передать тон»	Тон задаётся содержанием, а не языком промпта. Можно написать: «Тон: дружелюбный, для подростков» – и получить нужное на любом языке.

Итеративность и уточнение

Не ищите идеальный промпт с первой попытки.
Лучше быстро получить черновик и улучшать его шаг за шагом.

Даже самый продуманный промпт не всегда даёт идеальный результат с первого раза. Эффективный промптинг – это процесс, а не однократное действие.

Не бойтесь:

– Уточнять ответ: «Сделай короче», «Добавь пример»
– Менять стиль: «Теперь напиши это в шутливом тоне»
– Запрашивать переформулированный ответ: «Перефразируй третий пункт более просто»

ИИ отлично работает в режиме диалога.

Пример итеративного подхода:

Шаг 1 (первая версия): Напиши, зачем учить иностранный язык.

Шаг 2 (уточнение): Теперь оформи это как 4 пункта в маркированном списке.

Шаг 3 (углубление): Добавь к каждому пункту конкретный пример из жизни.

Шаг 4 (адаптация): Перепиши всё в мотивирующем тоне, как пост в соцсетях.

Можно также использовать продвинутые техники, упомянутые в разделе «Методы промпт-инжиниринга»:

Few-Shot Prompting (промптинг с примерами) – показать модели несколько примеров «Ввод → Вывод». Это самый мощный способ объяснить ей сложную или нишевую задачу.

Например: «Переведи слова с английского на испанский. Пример: 'dog' -> 'perro'. Теперь переведи: 'cat' -> ...»

Chain-of-Thought (CoT) – просьба к модели рассуждать шаг за шагом. Например: «Объясни свои рассуждения по шагам», «Подумай вслух». Критически важно для сложных логических и математических задач.

Сложные задачи лучше разделять. Разбейте промпт на этапы. Например: «1. Сгенерируй три идеи для поста. 2 Для лучшей идеи напиши заголовок. 3. Напиши текст поста на 200 слов».

Компоненты хорошего промпта

Задача (Instruction) – это основная команда или инструкция для ИИ, которая говорит ИИ, что именно нужно сделать. Без чёткой задачи ИИ может дать общий или не по делу ответ.

Примеры:

- Напиши краткое изложение.
- Переведи текст на французский.
- Объясни концепцию квантовой запутанности пятилетнему ребёнку.
- Создай маркетинговый слоган.

Контекст (Context) – здесь пользователь задает *«роль»*, в которой должен выступить ИИ и *«фон»:* аудиторию, стиль, тон, дополнительные условия (фоновые данные, тема, тон). Чем больше релевантного контекста – тем точнее и уместнее будет ответ.

Примеры:

– Пиши, как опытный учитель информатики.

– Представь, что ты пишешь статью для блога о здоровом образе жизни.

– Ответь в деловом стиле как менеджер компании.

– Учитывай, что пользователь – новичок в программировании.

Данные (Input Data) – это «сырой материал», с которым работает ИИ или непосредственный объект для обработки (текст для краткого изложения, код для исправления).

Примеры:

– Вот текст для резюмирования: [вставьте текст].

– Ниже – код на Python, в котором есть ошибка: ....

– Тема для эссе: «Искусственный интеллект и этика».

Без данных ИИ не знает, на чём выполнять задачу.

Оформитель вывода (Output Indicator) – указание формата оформления результата помогает получить сразу готовый к использованию ответ (список, JSON, маркированный список, эссе на 500 слов).

Примеры:

– Ответь в виде маркированного списка.

– Верни результат в формате JSON.

– Ограничься 150 словами.

– Сделай таблицу с двумя колонками: «Плюсы» и «Минусы».

– Напиши, как твит (не более 280 символов).

– Представь результат в виде списка из 5 пунктов.

– Представь результат в формате: заголовок + 3 абзаца + вывод.

Такой подход:

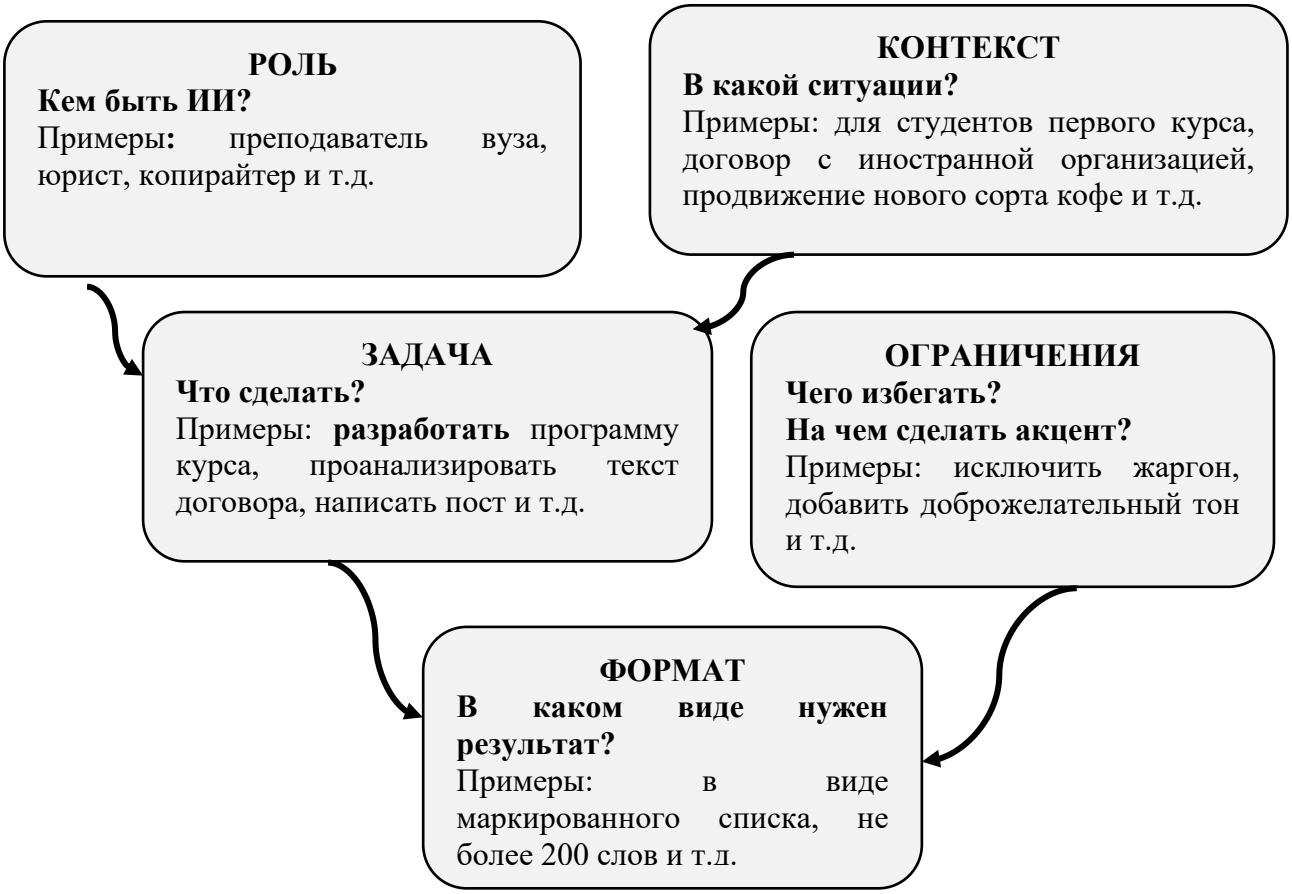
– повышает точность ответа,

– уменьшает количество итераций («переделай»),

– делает взаимодействие с ИИ более предсказуемым и эффективным,

– особенно полезен при автоматизации, обучении, аналитике, работе с API.

Схема 6. «Полный промпт»



Эффективный промпт – это не удачно сформулированный вопрос, а структурированная инструкция, сочетающая ясную задачу, актуальный контекст, чёткую роль, удобный формат и возможность уточнения.

Чем больше вы практикуетесь в осознанном построении промптов, тем увереннее и продуктивнее становится ваше взаимодействие с ИИ.

ИИ – это не волшебник. Это инструмент. А хороший промпт – это ваш чертёж, по которому этот инструмент создаёт нужный результат (Таблица 18).

Таблица 18. «Чек-лист “5 шагов к идеальному промпту”»

№ п/п	ШАГ	Проверка: задай себе вопрос	Пример слабого промпта	Пример сильного промпта
1.	**Ясность и конкретность:** сформулируй чёткую задачу с глаголом действия.	Что именно нужно сделать — написать, объяснить, сравнить и т.д. ?	Расскажи про экологию.	Объясни, почему переработка пластика важна для океанов, простыми словами.
2.	**Указание контекста:** дай информацию о цели, аудитории и стиле.	Если бы я передал эту задачу человеку, какие пояснения дал бы?	Напиши текст.	Напиши пост для Instagram о пользе сна для подростков. Тон – дружелюбный, с юмором.
3.	**Роль и персонаж:** назначь ИИ роль эксперта.	В какой роли должен действовать ИИ: учитель, редактор, маркетолог, кто-то еще?	Проверь мой текст.	Ты – профессиональный редактор. Проверь текст на стиль и логическую связность. Предложи улучшения.
4.	**Формат вывода:** укажи, в каком виде нужен результат.	Удобно ли будет использовать ответ без переформатирования?	Перечисли преимущества электромобилей.	Перечисли 5 преимуществ в виде маркированного списка. Каждый – не более 8 слов.
5.	**Итеративность:** будь готов уточнять и дорабатывать.	Можно ли улучшить результат за 2–3 шага?	–	Шаг 1. Зачем учить язык? Шаг 2. Оформи как 4 пункта. Шаг 3. Добавь примеры. Шаг 4. Сделай мотивирующим постом в соцсети.

§ 5. Особенности промптинга для разных типов моделей

Теперь, когда мы разобрались с правилами написания эффективных промптов (например, по чек-листу из 5 шагов), важно понимать, что не все ИИ одинаковы.

Для каждого ИИ, как и в других сферах, нужны разные инструменты. Мы же не будем резать бумагу молотком и ложкой забивать гвозди. Так и с искусственным интеллектом. Есть разные модели ИИ, и каждая «заточена» под свою задачу. Чтобы получить лучший результат, нужно подбирать промпт под конкретный тип ИИ.

Рассмотрим три основные группы моделей и как им правильно ставить задачу.

А. Промптинг для языковых моделей

Языковые модели (YandexGPT, ChatGPT, Claude, Llama и другие), как мы уже знаем, работают с текстом. Они могут отвечать на вопросы, писать статьи, письма, сценарии, переводить, кратко пересказывать текст, классифицировать тексты.

Как составлять промпты

1. **Ролевое моделирование** – дай ИИ роль, и он будет отвечать, как эксперт.

Слабо: Объясни, как работает велосипед.

Сильно: Ты – учитель физики в школе. Объясни, как работает велосипед, чтобы понял ученик 6 класса.

Такой промпт помогает ИИ «включиться» в нужную роль.

2. **Структура ответа** – укажи, как должен выглядеть ответ: список, абзац, таблица и т.д.

Слабо: Перечисли правила дорожного движения для пешеходов.

Сильно: Перечисли 5 основных правил для пешеходов в виде маркированного списка. Каждый пункт – не более 10 слов.

3. **«Цепочка рассуждений» (Chain-of-Thought, CoT)** – это техника, при которой мы просим ИИ думать открыто, чтобы мы видели логику его рассуждений шаг за шагом.

Полезно для сложных задач: логика, математика, анализ.

Слабо: Сколько будет (15 + 7) × 2 – 10?

Сильно: Реши пример по шагам: (15 + 7) × 2 – 10. Объясни каждый шаг.

Тогда ИИ не просто даст ответ, а покажет ход мыслей – и ошибётся реже.

Таким образом для языковых моделей важно указывать роль, структуру ответа и просить «думать вслух» при сложных задачах.

Шаблон текстового промпта

Схема 7. «Шаблон универсального текстового промпта»

Шаблон универсального текстового промпта *(он также представлен на схеме 7):* «Ты [роль эксперта]. Напиши [тип текста] на тему [тема] для [целевая аудитория]. Контекст: [опишите ситуацию].

Требования:

– Объём: [количество слов].
– Тон: [формальный/дружелюбный/профессиональный].
– Структура: [укажите желаемую структуру ответа].
– Особенности: [специфические требования]».

Шаблон универсального текстового промпта для создания структуры презентации: «Создай структуру презентации на тему «[тема]» для [целевая аудитория]. Цель презентации: [информировать/убедить/обучить/вдохновить].

Параметры:

– Количество слайдов: [число].

– Продолжительность: [минуты].

– Формат: [онлайн/офлайн/питч/лекция].

Для каждого слайда укажи:

1. Заголовок.
2. Ключевые пункты (3-4 тезиса).
3. Идеи для визуализации.

Тон: [профессиональный/вдохновляющий/обучающий/продающий]».

Примеры промптов для генерации текста

Деловое письмо

Слабый промпт (с непредсказуемым результатом): «Напиши письмо клиенту».

Сильный промпт (детализировано описывающий желаемый результат):

«Ты профессиональный менеджер по работе с клиентами. Напиши вежливое письмо клиенту, который недоволен задержкой доставки заказа на 3 дня.

Контекст: Заказ №12345, задержка из-за проблем с логистикой, уже отправлен, прибудет через 2 дня.

Требования:

– Объём: 100-150 слов.

– Тон: извиняющийся, но профессиональный.

– Структура: приветствие + извинение + объяснение + компенсация (скидка 10%) + заключение.

– Избегай излишних оправданий».

Обучающий материал

Слабый промпт (с непредсказуемым результатом): «Объясни маркетинг».

Сильный промпт (детализировано описывающий желаемый результат):

«Ты опытный преподаватель бизнес-дисциплин. Объясни концепцию «целевая аудитория в маркетинге» студенту первого курса, который никогда не изучал маркетинг.

Требования:

– Начни с простого определения.
– Используй 2-3 примера из повседневной жизни.
– Объясни, почему это важно.
– Объём: 250-300 слов.
– Избегай профессионального жаргона.
– Структура: определение → примеры → практическое применение → вывод».

Контент для социальных сетей

Слабый промпт (с непредсказуемым результатом): «Напиши пост для кофейни».

Сильный промпт (детализировано описывающий желаемый результат): «Ты SMM-специалист. Создай пост для Max для кофейни, которая запускает новое сезонное меню с тыквенными напитками.

Контекст: Аудитория 25-40 лет, ценят уют и качество, период запуска – осень.

Требования:

– Длина: 100-120 символов.
– Тон: тёплый, уютный, слегка игривый.
– Включи призыв к действию.
– Добавь 5-7 релевантных хэштегов.
– Предложи идею для сопроводительного фото».

Структура бизнес-презентации

Слабый промпт (с непредсказуемым результатом): «Создай презентацию на тему “Внедрение удалённой работы в компании”».

Сильный промпт (детализировано описывающий желаемый результат):

«Создай структуру презентации на тему «Внедрение удалённой работы в компании» для руководителей среднего звена.

Цель: Убедить в преимуществах и показать план внедрения

Параметры:

– 10-12 слайдов.
– Продолжительность: 15 минут.
– Формат: очная встреча с возможностью вопросов.

Для каждого слайда укажи:

1. Заголовок.
2. 3-4 ключевых пункта (тезисы).
3. Рекомендации по визуализации (графики, иконки, фото).

4. Примечания для докладчика.

Структура должна включать:

– Введение (проблема/возможность).
– Текущая ситуация.
– Преимущества решения (с данными).
– План внедрения (пошагово).
– Ожидаемые результаты.
– Ответы на возможные возражения.
– Призыв к действию.

Тон: профессиональный, уверенный, подкреплённый фактами».

Образовательная презентация

Слабый промпт (с непредсказуемым результатом): «Создай презентацию на тему “Основы тайм-менеджмента”»/

Сильный промпт (детализировано описывающий желаемый результат):

Создай структуру обучающей презентации на тему «Основы тайм-менеджмента» для студентов 3-4 курса.

Цель: Научить практическим методам управления временем.

Параметры:

– 8-10 слайдов.
– Продолжительность: 20 минут.
– Формат: онлайн-лекция с интерактивными элементами.

Для каждого слайда укажи:

1. Заголовок (вопросом или утверждением).
2. Основной контент (объяснения + примеры).
3. Интерактивный элемент (вопрос к аудитории, мини-задание).
4. Визуальные идеи.

Обязательно включи:

– Вовлекающее введение с проблемой.
– 3 основных техники (с пошаговыми инструкциями).
– Реальные примеры применения.
– Частые ошибки.
– Практическое задание.
– Ресурсы для дальнейшего изучения.

Тон: дружелюбный, мотивирующий, с элементами юмора.

Повышение достоверности ответа модели

Промпт может быть просто запросом, но проектировать поведение модели. Такой промпт относится к передовым методам (Self-Refine) и крайне полезен для сложных тем. Его главная ценность – в смещении парадигмы с «получить ответ любой ценой» на «получить достоверный ответ».

Например, если мы хотим *повысить достоверность ответа модели и минимизировать ее возможные галлюцинации*, промпт может быть следующим:

«Прежде чем дать окончательный ответ, оцени свою уверенность в нем. Если у тебя есть сомнения или информация может быть неполной/неоднозначной, обязательно:

1. Перечисли области, в которых ты чувствуешь наибольшую неопределенность.
2. Задай мне уточняющие вопросы, чтобы прояснить эти моменты.

Не давай ответ, пока не будешь уверен, что он точен и полон».

Для сложных задач: «Действуй как эксперт-консультант. Твоя задача – не просто ответить на мой вопрос, а обеспечить максимальную точность ответа.

Шаг 1: Проведи внутреннюю оценку. Проанализируй мой запрос на предмет неоднозначностей, пробелов в твоих знаниях или областей, где возможны разные трактовки.

Шаг 2: Если неопределенность высока, не давай ответ. Вместо этого:

– Сформулируй, в чем именно заключается неопределенность.
– Задай мне уточняющие вопросы, чтобы ее снизить.

Шаг 3: Получив уточнения, дай финальный, обоснованный ответ».

Качественная просьба «оцени неопределенность» сама по себе уже даст огромный положительный эффект.

Простой и надежный промпт: «Если ты не уверен на 100% в ответе, сначала задай уточняющие вопросы».

Внимание! Иногда предлагаются варианты промптов с количественными оценками, например, «Прежде чем дать мне ответ, оцени его неопределённость. Если она больше, чем 0.1 – задавай мне уточняющие вопросы до тех пор, пока неопределённость будет 0.1 или меньше. Неопределенность – вероятность того, что новая информация существенно изменит ответ».

Такой промпт наталкивается на фундаментальные ограничения современных LLM, а именно:

1. Модель не может достоверно сказать, что ее неопределенность равна 0.15, а не 0.09, поскольку модели по своей природе не вычисляют точную вероятность в математическом смысле, т.е. цифра «0.1» – это иллюзия точности. Она будет имитировать этот расчет, что само по себе может быть неточно.

2. Понятие «существенно изменит» – субъективно. Для одного человека изменение даты на 1 год – мелочь, для историка – катастрофа. Модель не разделяет ваших критериев «существенности».

3. Модель может войти в состояние «аналитического паралича», постоянно находя новые «источники неопределенности» и задавая уточняющие вопросы, даже когда это уже не нужно. Вы можете получить риск бесконечного цикла.

Б. Промптинг для генерации изображений

Визуальные подсказки требуют иного мышления – здесь важны не слова, а детали восприятия. Не переживайте, если поначалу будет сложно. Это нормально!

Модели для работы с текстом (ChatGPT, Claude) и специализированные модели для работы с изображениями (Midjourney, DALL·E) – это разные системы. То, что работает в текстовом запросе, не всегда применимо к визуальному описанию, и наоборот.

Таблица 19. «Промпт для генерации изображений»

Элемент	Что это?	Пример
Объект	Что на картинке главное?	Портрет женщины, дракон, город будущего и т.п.
Детали / действия	Как выглядит объект или что делает?	С серебряными волосами, с расправленными крыльями, летит между небоскрёбами и т.п.
Стиль / художник	Как должна выглядеть картинка?	В стиле фэнтези, как у Грега Рутковски, мультяшный стиль и т.п.
Композиция	Как расположены объекты?	В центре, крупным планом, вид сбоку и т.п.
Цвета / освещение	Какая атмосфера?	Тёплые тона, контровое освещение, сумерки и т.п.
Параметры	Технические настройки (опционально).	--ar 2:3(соотношение сторон), --v 6(версия Midjourney)

Таблица 20. «Примеры промптов для генерации изображений»

Результат	Слабый промпт (с непредсказуемым результатом)	Сильный промпт (детализировано описывающий желаемый результат)
Иллюстрация для презентации	Нарисуй картинку про бизнес.	Минималистичная векторная иконка, изображающая рост бизнеса: стилизованный график с восходящей стрелкой, переплетённый с ростком растения. Плоский дизайн (flat design), корпоративная цветовая гамма сине-зелёных оттенков, белый или прозрачный фон, чёткие геометрические формы, профессиональный стиль, подходит для бизнес-презентации, формат квадратный.
Фото для блога/статьи	Сгенерируй фото к статье о рабочей продуктивности.	Профессиональная фотография современного рабочего пространства для статьи о продуктивности: минималистичный белый письменный стол, открытый ноутбук Apple с пустым экраном, элегантная кружка с кофе, небольшое зелёное растение в горшке, стильный блокнот и ручка. Композиция: вид сверху под углом 45 градусов. Освещение: мягкий естественный свет из окна, создающий лёгкие тени. Цвета: нейтральная пастельная палитра, бежевые и белые тона с акцентами зелени. Атмосфера: чистота, порядок, вдохновение. Стиль: скандинавский минимализм, высокое качество, детализированная текстура дерева.
Иллюстрация для маркетинга	Нарисуй иллюстрацию для рекламы онлайн-курсов.	Яркая привлекательная иллюстрация для рекламного баннера онлайн-курсов: молодая женщина (25-30 лет) сидит с ноутбуком в уютном домашнем интерьере, вокруг неё парят светящиеся иконки образования (книги, дипломы, лампочки идей). Стиль: современная цифровая иллюстрация с градиентами, слегка мультяшный, но реалистичный. Цветовая палитра: тёплые оранжево-фиолетовые градиенты с голубыми акцентами. Композиция: центральная фигура, динамичное расположение элементов. Настроение: мотивирующее, вдохновляющее, энергичное. Освещение: мягкое, с эффектом свечения от экрана.

Шаблон описания для создания изображений:

[Главный объект] + [Действие/состояние] + [Стиль/художественное направление] + [Композиция] + [Освещение] + [Цветовая палитра] + [Настроение/атмосфера] + [Технические детали].

Пример промпта для генерации изображения приведен в Таблице 19.

Чем больше деталей – тем точнее картинка! Поэтому не боимся писать длинно, используем английский язык с иностранными моделями (многие модели лучше его понимают), если картинка не та – уточняем требования: «сделай фон темнее», «добавь дождь» (Таблица 20). Однако, следует сразу отметить, часто нейросеть может изменить изображение совсем не так, как запрашивал пользователь.

B. Промптинг для генерации музыкальных произведений

Создание музыки с помощью ИИ – это процесс, требующий чёткого донесения творческой идеи пользователя. В отличие от генерации изображений, здесь нейросеть работает не только с описанием, но и непосредственно с материалом будущего трека – его текстом и вокальной партией. Каждая запятая, расстановка ударений, образность и даже форматирование текста становятся частью промпта и напрямую влияют на результат (Таблица 21).

Важное ограничение! Во избежание нарушений авторских прав и правил платформ (таких, например, как Suno), не рекомендуется использовать имена реальных музыкантов, групп или названия существующих песен в промптах. Вместо этого необходимо описать желаемый стиль, используя жанровые определения и характеристики звучания.

Шаблон описания для создания музыки

[Жанр] + [Темп и ритм] + [Инструментовка] + [Настроение/атмосфера] + [Структура] + [Тематика/сюжет] + [Вокальная партия] + [Текст песни] + [Технические детали].

Таблица 21. «Промпт для генерации музыкальных произведений»

Элемент	Что это?	Пример
Жанр	Основное музыкальное направление.	Синтвейв, Классика, Хип-хоп, Эмбиент, Фолк-рок, Джаз-фанк и т.п.
Темп (BPM) и ритм	Скорость и ритмический рисунок.	Медленный (70 BPM), быстрый и энергичный (140 BPM), синкопированный бит, шаффл.
Инструментовка	Какие инструменты используются.	Ведущее пианино, мощные электронные басы, акустическая гитара, струнный оркестр, эпический хор.
Настроение, атмосфера	Эмоциональная окраска трека.	Ностальгическое, мрачное и загадочное, радостное и воодушевляющее, меланхоличное, напряжённое.
Структура	Примерное строение композиции.	Интро → Куплет → Припев → Бридж → Финальный припев, постепенное нагнетание.
Тематика/Сюжет	«История», которую рассказывает музыка.	Путешествие через космос, размышления под дождём, победа в великой битве, воспоминания о лете.
Вокал	Критически важный элемент. Тип, тембр и характеристики вокала.	Чистый женский вокал, мужской дисторшн-вокал (скрим), бархатный баритон, смешанный хор, рэп-куплеты в агрессивной манере, фейковый свингующий джазовый вокал.
Текст песни	Фактический материал для вокала. Его качество и образность напрямую диктуют манеру исполнения.	*[Текст подаётся в определённом формате, это может быть просто стихотворение по абзацам или сплошным текстом, с простановкой ударений или без этого, с указанием к каждой части текста: [Куплет], [Припев], [Бридж], [Проигрыш] и т.д.]*
Технические детали	Качество и особенности звучания.	Высококачественный продакшен, лоу-фай фильтр, широкая стереопанорама, глубокий саб-бас.

Жёсткий контроль или творческое соавторство?

На практике существует два принципиально разных подхода к генерации, и выбор между ними определяет конечный результат.

1. *Жёсткий контроль (детализированный промптинг)*

Пользователь выступает в роли сценариста и режиссёра, который до мелочей прописывает каждый аспект будущего трека. Такой подход идеально подходит для создания музыки под конкретный заказ (например, для рекламы, саунддизайна проекта, где нужен точный BPM и настроение), а также когда у пользователя есть очень чёткое, сформированное видение результата.

Однако, в этом случае всегда имеется риск, что чрезмерно жесткий промпт может лишить результат элемента случайности и «креативной вольности», которая при генерации песен, в отличие от работы с текстами, может быть вполне уместна. Нейросеть, будучи зажатой в узкие рамки, может выдать технически точный, но эмоционально плоский и предсказуемый трек. И даже включенная опция «Weirdness» (странность, креативность) в Suno при чрезмерно подробном промпте приведет не к креативности, а именно к странности – неуместным звукам, неудачному произнесению слов и пр.

2. *Творческое соавторство (итеративный промптинг)*

Пользователь выступает в роли соавтора, который задаёт общее направление, но оставляет нейросети пространство для творческой интерпретации. Промпт не предписывает, а предлагает и вдохновляет. Этот подход идеален для поиска вдохновения, нестандартных решений и неожиданных творческих находок. Многие пользователи отмечают, что именно так рождаются их самые интересные работы.

Наиболее эффективной часто оказывается золотая середина. Чётко задаем основу (жанр, настроение, текст), но будем открыты к сюрпризам в аранжировке и вокальной подаче. Используем нейросеть не как бездушный инструмент, а как партнёра, способного предложить неожиданный поворот, который обогатит первоначальную идею.

Особенности генерации песен (с текстом)

Для создания песен с вокалом сам текст становится самой важной частью промпта. Мало указать общую тему – рекомендуется предоставить готовый, отформатированный и образный текст. Нейросеть не только считывает слова, но и интерпретирует их эмоциональный и образный строй, самостоятельно

расставляя вокальные акценты: может перейти от шёпота к крику, замедлиться на метафоричной фразе или подчеркнуть ритм в остром тексте.

Ключевые принципы подачи текста в промпте:

Структура: целесообразно разделить текст на [Куплет], [Припев], [Бридж], [Проигрыш] и т.д. Это помогает модели понять композицию песни.

Образность и качество текста: хорошо «работают» сильные глаголы, метафоры и яркие образы. Нейросеть способна уловить настроение строки и передать его через вокал.

Расстановка ударений и ритма: можно использовать CAPS LOCK для акцента на ключевых словах или слогах, которые должны быть выделены вокалом. Если слово может иметь разные ударения и в зависимости от этого – разный смысл, или в авторском тексте слово имеет нестандартное ударение, то такое ударение необходимо поставить. Это возможно сделать CAPS LOCK, а можно использовать знак ударения – надстрочный орфографический знак, который ставится над гласной буквой, соответствующей ударному звуку («акут» (острое ударение) – распространённый знак, наклонён справа налево, «гравис» – менее распространённый знак, имеет противоположный наклон, обозначает побочное, слабое, второстепенное ударение).

Чем больше деталей – тем точнее результат, когда это необходимо. Но качественный, образный текст и готовность к диалогу с нейросетью – это залог по-настоящему живого и уникального звучания. И здесь также действует принцип – чем больше пользователь работает с нейросетью, тем лучше она узнает его предпочтения и тем быстрее выдает желаемый результат.

Г. Промптинг для работы с кодом

С кодом работают, например, GigaChat, Yandex Code Assistant, GitHub Copilot, Codeium, ChatGPT и др. Эти модели помогают писать код, находить ошибки, объяснять, как работает программа, писать комментарии к коду.

При написании промпта для кода важно:

– *быть максимально конкретным* (не «Сделай что-нибудь», а «Сделай вот это, вот так, на этом языке»),

– *указывать язык программирования и библиотеку* (иначе ИИ может написать код на другом языке).

Слабо: Напиши функцию, которая считает среднее.

Сильно: Напиши функцию на Python, используя pandas...

– *описать вход и выход* (какие данные принимает функция и что она должна вернуть).

Пример хорошего промпта (разбор промпта представлен в Таблице 22): «Напиши функцию на Python, используя pandas. Функция должна принимать на вход dataframe и имя колонки, а возвращать среднее значение и стандартное отклонение для этой колонки. Игнорируй NaN значения».

Такой запрос понятен и даёт сразу рабочий код.

Таблица 22. «Разбор промпта для генерации кода»

Что указано	Почему важно
на Python, используя pandas	Язык и библиотека.
принимает dataframe и имя колонки	Входные данные.
возвращает среднее и стандартное отклонение	Выходные данные.
игнорируй NaN	Обработка ошибок и пропущенных данных.

Другие полезные формулировки:

– Explain what this code does in simple terms. / Объясни, что делает этот код на простом языке.

– Fix the error in this code: [insert code]. / Исправь ошибку в этом коде: [вставить код].

– Add a comment to each line. / Добавь комментарии к каждой строке.

– Write a test for this function using PyTest / Напиши тест для этой функции на PyTest.

§ 6. Как применять и с чего начать

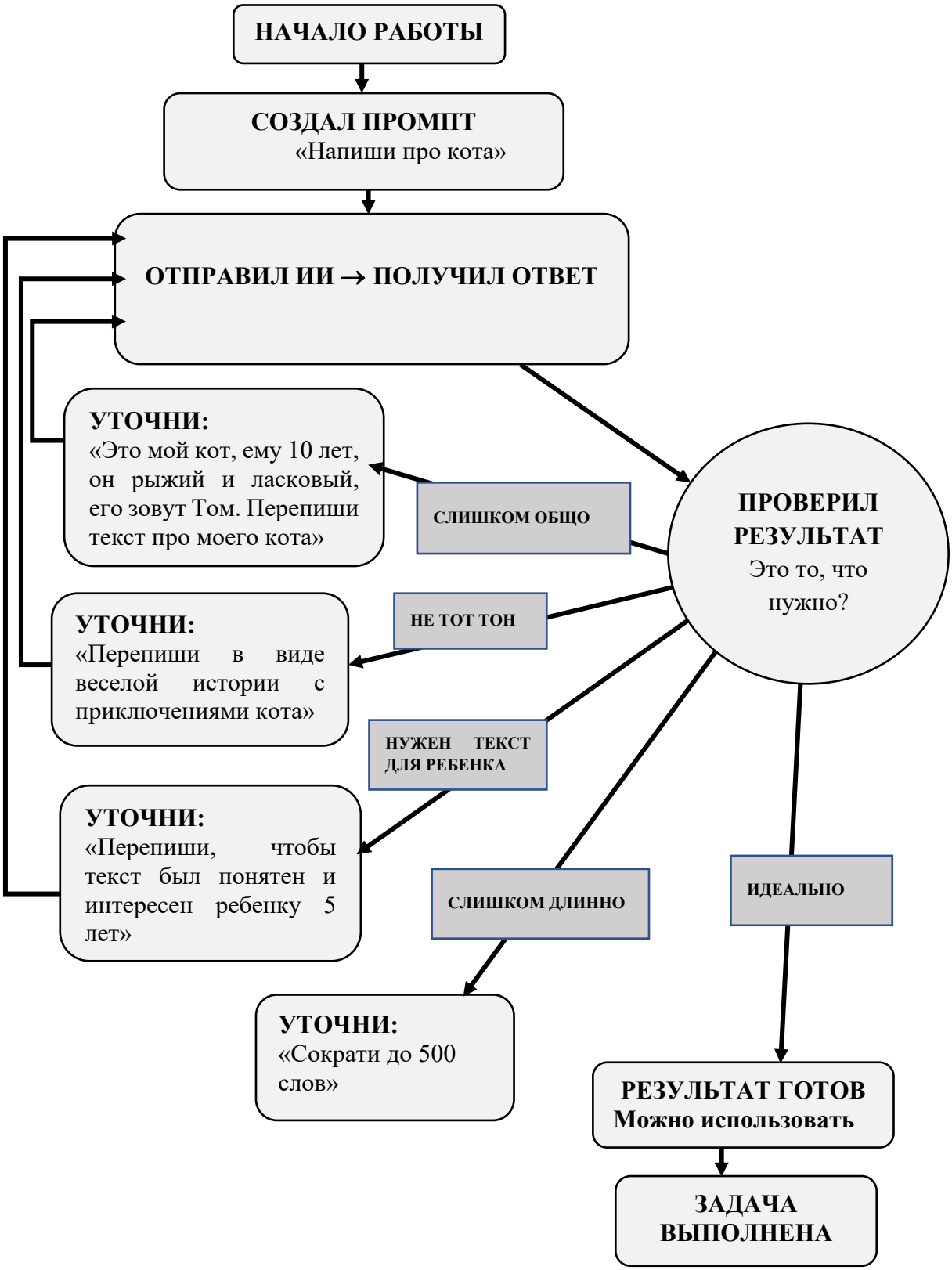
Не стремитесь сразу использовать все 5 компонентов промпта! Начните с описания задачи. Посмотрите, что получили в результате. Составьте промпт из одного компонента, получите результат, затем – из трёх компонентов и получите результат. Затем уточните детали, добавьте контекст и ограничения. Это называется итерацией (Схема 8) – и это нормальный путь к идеальному промпту.

Перед отправкой промпта проверьте:

1. Понятно ли, кто должен отвечать? (роль)
2. Понятно ли, что нужно сделать? (задача)
3. Указано ли, в каком виде нужен результат? (формат)

Если «да», то все идет по плану!

Схема 8. «Процесс итерации»



Стартовый уровень

Блок «Задача» – в промпте нужно указать, что ИИ должен сделать (например: «Напиши пост про кофе»).

Напишите нейросети, что она должна сделать (ответить на вопрос, что-то сгенерировать и т.п.). ИИ ответит, но результат будет общим. Это не страшно. Вы всегда можете задать уточняющие вопросы, конкретизировать задачу, предъявить требования к результату, плавно переходя на следующий уровень взаимодействия – базовый.

Базовый уровень

Используйте 3 ключевых компонента – этого хватит для решения большинства повседневных задач:

1. **Блок «Роль»** – в промпте нужно указать, кем должен быть ИИ, отвечая на запрос пользователя, какими компетенциями обладать (например: «Ты опытный копирайтер»).
2. **Блок «Задача»** – в промпте нужно указать, что ИИ должен сделать (например: «Опиши достоинства нового кофе»).
3. **Блок «Формат»** – в промпте нужно указать, в каком виде должен быть представлен результат? (например: «В виде поста для Max, 100 символов»).

Пример базового промпта: «Вы SMM-специалист. Напишите пост для Max о новом сорте кофе. Формат: 100–120 символов, с призывом к действию и тремя хэш-тегами».

Попробуйте сами! Возьмите любую задачу (письмо, пост, заметку) и составьте промпт из этих трёх блоков.

Расширенный уровень (когда нужна точность)

Если результат критически важен (презентация, отчёт, реклама), то добавьте ещё 2 компонента:

4. **Блок «Контекст»** – в промпте нужно описать контекст, т.е. описать ситуацию, в рамках которой решается задача, поставленная перед ИИ (например: «Наша кофейня представляет осеннюю коллекцию для молодых родителей»).

5. **Блок «Ограничения»** – в промпте нужно указать то, что ИИ следует придерживаться или избежать в ответе (например: «Тон – тёплый и уютный. Без жаргона. Не более 120 символов»).

Полный промпт (5 компонентов):

Вы SMM-специалист. Напишите пост для Max о новом тыквенном латте.

Контекст: Кофейня запускает осеннюю коллекцию для аудитории 25–40 лет.

Формат: 100–120 символов, с призывом к действию и 5 хэш-тегами.

Ограничения: Тон – тёплый, уютный, слегка игривый. Избегайте таких слов, как «инновационный», «революционный».

Один и тот же принцип (например, конкретность) работает везде, но форма промпта зависит от цели (Таблица 23):

– Хочешь текст? → давай ИИ роль и структуру.

– Хочешь картинку? → опиши всё: кто, что, где, как светит, в каком стиле.

– Хочешь код? → будь как технический заказчик: пиши промпт чётко, по делу, с деталями.

Таблица 23. «Особенности промпта под тип задачи»

Тип задачи	Какой ИИ использовать	На что обратить внимание в промпте
Написать текст, ответить на вопрос	Языковая модель (ChatGPT, Claude, YandexGPT, GigaChat)	Роль, структура, CoT: четко задавайте роль (например, «действуй как опытный копирайтер»), требуйте конкретной структуры (план, тезисы) и используйте Chain of Thought (CoT) – пошаговое рассуждение для сложных тем.
Создать картинку	Генератор изображений (Midjourney, DALL·E, Kandinsky, Шедеврум)	Детали, стиль, композиция, параметры: указывайте детали объекта и фона, художественный стиль (аниме, фото и т.д.), композицию (крупный план, размытый фон) и технические параметры (соотношение сторон).
Написать или исправить код	Код-ассистент (Copilot, Codeium, Yandex Code Assistant, GigaCode)	Язык, вход/выход, библиотеки: четко обозначайте язык программирования, формат входных данных и ожидаемого результата, а также используемые библиотеки или фреймворки.

Десять золотых правил промпт-инжиниринга и одна хитрость

1. **Будьте конкретны.**

Вместо: «хороший текст» → «убедительный, дружелюбный, 150 слов».

2. **Задавайте роль.**

«Ты эксперт по...» помогает AI выбрать правильный тон и подход.

3. **Показывайте примеры.**

Значительно улучшает результат текст в запросе: «Как в этом примере: [пример]».

Профессиональный приём – дайте ИИ пример прямо в запросе: «Напишите пост в стиле этого примера: [вставьте 1–2 предложения]. Теперь создайте новый пост на тему X». Это называется подсказка с несколькими примерами и значительно повышает точность стиля.

4. **Используйте структуру.**

Разбивайте промпт на секции: «Контекст / Задача / Требования».

5. **Итерируйте.**

Первый результат редко идеален – улучшайте промпт пошагово.

Совет по настройке ИИ:

Если результат слишком шаблонный, попросите ИИ «проявить креативность» или «предложить необычный вариант».

Если результат хаотичный, уточните: «Будь точным и опирайся на факты, без вымысла».

Это имитирует изменение параметра температуры без технических подробностей.

6. **Указывайте, чего избегать.**

«Без жаргона»

«Избегай клише»

«Не используй пассивный залог» и т.п.

7. **Определяйте аудиторию.**

«Для новичков»

«Для профессионалов»

«Для детей 10 лет» и т.п.

8. **Экспериментируйте с тоном.**

Формальный

Дружелюбный

Вдохновляющий

Юмористический

Строгий и т.п.

9. **Запрашивайте структуру.**

«Начни с ... затем ... заверши ...».

«В формате списка».

«Как email» и т.п.

10. **Сохраняйте работающие промпты**.

Создайте библиотеку успешных промптов для повторного использования.

И, наконец, про хитрость – **пользуйтесь наработками специалистов!** Например, зайдите в «Каталог промптов» на https://alice.yandex.ru/prompthub.

§ 7. Безопасность и ответственность

Искусственный интеллект – мощный помощник, но он не заменяет ваше суждение. Чтобы использовать ИИ с максимальной пользой и минимальными рисками, соблюдайте три правила: не вводите конфиденциальные данные, не верьте слепо, не копируйте (Таблица 24).

Правило ответственного использования – человек является автором и редактором. ИИ предлагает идеи, формулировки и структуры, но именно человек несет полную ответственность за то, что отправляет, публикует или принимает на основе ответа нейросети.

Советы для безопасной работы

– *Анонимизируйте данные* – вместо «Клиент Иван Петров, заказ № 12345» пишите «Клиент, заказ на кофе».

– *Уточните источник* – если ИИ приводит статистику, спросите: «Откуда эти цифры?». Если он не может подтвердить их достоверность, не используйте их.

– *Не делитесь подсказками, содержащими конфиденциальную информацию* – даже «рабочие» шаблоны могут содержать детали, которые лучше не разглашать за пределами вашей команды.

Осознанное использование ИИ – признак профессионализма. Когда вы работаете с технологией ответственно, вы не только защищаете себя и других, но и вносите вклад в формирование культуры честного и безопасного цифрового пространства.

Хороший промпт – это не только эффективно, но и этично.

Таблица 24. «Что нельзя делать для безопасности»

РИСК	ПОЧЕМУ ЭТО ОПАСНО	КАК ИЗБЕЖАТЬ
Вводить конфиденциальные данные (паспорта, номера счетов, внутренние документы компании, персональные данные клиентов).	ИИ-системы могут сохранять или использовать ваш запрос для обучения. Даже если модель «не запоминает», вы не контролируете, где обрабатывается ваш текст.	Никогда не отправляйте в ИИ то, что вы не стали бы публиковать в открытом доступе.
Просить «переписать текст конкурента» или копировать чужой контент.	Это нарушает авторские права. ИИ может перефразировать текст, но суть останется чужой – и вы рискуете как с юридической, так и с репутационной точки зрения.	Используйте ИИ для создания оригинального контента, а не для маскировки плагиата.
Считать правдой всё, что говорит ИИ.	ИИ может генерировать убедительную, но ложную информацию (это называется *галлюцинацией*). Особенно часто это происходит с датами, именами, статистическими данными и цитатами.	Всегда проверяйте факты через надёжные источники. Особенно если результат используется в работе, учёбе или публикации.

§ 8. Инструменты и будущее промптинга

Промптинг – это не просто набор правил, а искусство диалога с искусственным интеллектом, где каждая деталь имеет значение. По мере роста важности навыка промптинга появляются и специализированные инструменты, помогающие систематизировать работу.

Мы умеем хорошо писать промпты, но что, если необходимо проверить, какой промпт работает лучше, или сохранить лучшие промпты, чтобы не искать их каждый раз, или создать целую цепочку промптов, где результат одного становится входом для другого?

Инструменты

Инструменты для промптинга – это как набор специальных «гаечных ключей и отвёрток», которые помогают профессионалам работать с промптами эффективнее (Таблица 25). Эти инструменты помогают создавать, тестировать, оптимизировать и хранить промпты.

Таблица 25. «Основные инструменты промптинга»

Инструмент	Главная цель	Идеален для:
PromptIDE	– создание, тестирование и систематизация промптов в одном месте.	регулярной и организованной работы с ИИ.
LangChain	– автоматизация сложных, многошаговых процессов с ИИ.	сборки «конвейеров» из нескольких задач.
PromptHub	– поиск и использование готовых, проверенных промптов.	новичков и для быстрого решения стандартных задач.
CoolPrompt	– глубокая и автоматизированная оптимизация промптов для лучшего качества.	получения максимально точного и выверенного результата от ИИ.

PromptIDE (среда разработки промптов)

PromptIDE – специальный «умный редактор» для работы с промптами, похожий на продвинутый текстовый редактор. Он превращает разрозненные запросы в управляемый проект.

Вместо того чтобы вручную копировать результаты из чата, вы можете:

сразу видеть результат, т.е. написали промпт – получили ответ ИИ в том же окне;

сравнивать варианты (A/B-тестирование) – можно написать два разных промпта для одной задачи и сразу увидеть, какой из них даёт лучшую картинку или текст;

хранить и систематизировать – сохранять удачные промпты в библиотеку, добавлять к ним описания (например, «Этот промпт отлично подходит для портретов в стиле киберпанк»).

Полезен тем, кто регулярно работает с ИИ и хочет систематизировать свой рабочий процесс, чтобы работать быстрее и качественнее.

LangChain (конструктор цепочек)

LangChain – это не отдельная программа, а набор инструментов (библиотека) для создания сложных, многошаговых процессов с участием ИИ.

Часто одной команды ИИ недостаточно. Нужна цепочка действий. LangChain позволяет автоматизировать эту цепочку, где результат одного шага автоматически передаётся на следующий.

Например, необходимо, чтобы ИИ: 1) прочитал длинную статью, 2) сделал её краткое содержание, 3) перевёл его на английский, 4) оформил как пост для LinkedIn. Вместо четырёх ручных операций вы настраиваете один раз цепочку в LangChain и получаете готовый результат по одной кнопке.

Полезен тем, кто хочет автоматизировать сложные, рутинные задачи, состоящие из нескольких шагов.

PromptHub и другие репозитории (библиотеки промптов)

PromptHub или другие репозитории – это онлайн-платформы, где пользователи со всего мира делятся проверенными и эффективными промптами. Это огромная коллекция готовых рецептов для ИИ.

Например, нужен промпт для создания реалистичного портрета в Midjourney. Заходим в такой репозиторий, находим шаблон вроде: photorealistic portrait of [subject], [details], by [artist], [lighting], подставляем свои данные и получаем отличный результат с первого раза. Это идеальный инструмент для новичков, чтобы быстро начать получать хорошие результаты, и для профессионалов, чтобы найти вдохновение или готовое решение для новой задачи.

CoolPrompt (набор инструментов для оптимизации)

CoolPrompt – это продвинутый фреймворк, разработанный в Университете ИТМО, который не просто создаёт промпты, а целенаправленно их улучшает с помощью нескольких «умных» алгоритмов. Он применяет научный подход к промптингу. Загружаем свой исходный, неидеальный промпт, а система его совершенствует.

CoolPrompt использует три основных инструмента:

HyPE – мгновенно превращает короткий и простой запрос в развёрнутую и профессиональную инструкцию.

ReflectivePrompt – работает как «селекционер», т.е. создаёт множество вариантов промпта, скрещивает и улучшает лучшие из них, постепенно «эволюционируя» к идеальной форме.

DistillPrompt – действует как строгий редактор: делает промпт короче, чётче и понятнее для ИИ, убирая всё лишнее и оставляя суть.

Если критически важно получать от ИИ максимально точный и качественный результат, и имеется готовность использовать более сложные, но мощные инструменты для его достижения, то CoolPrompt для этого и предназначен.

Эти инструменты позволяют не только создавать одиночные промпты, но и строить целые системы взаимодействия с ИИ, интегрируя их в бизнес-процессы и автоматизацию.

Тренды

Область промптинга стремительно развивается. Не будем говорить о «революциях», а рассмотрим конкретные тренды, которые уже появляются (Таблица 26).

Таблица 26. «Тренды в развитии промптинга»

Тренд	Суть	Пример
Мультимодальные промпты	Объединение в одном запросе текста, изображений, аудио и видео.	Загружаете эскиз логотипа и голосом описываете желаемый стиль.
Автоматический инжиниринг промптов	ИИ сам подбирает и тестирует лучшие формулировки для вашей задачи.	На запрос «напиши рекламу кофе» система предлагает 10 уточнённых вариантов и выбирает самый эффективный.
Персонализированные шаблоны	ИИ запоминает ваш стиль и сам применяет его для однотипных задач.	Вы говорите «напиши рассылку», а ИИ уже знает ваш тон и структуру писем.
Промпты как код	Промпты встраиваются в программы и бизнес-процессы, как обычный код.	Автоматическая служба поддержки использует промпты для генерации ответов клиентам.

Мультимодальные промпты

Мультимодальные промпты – это сложные, комплексные запросы, объединяющие разные форматы информации, что делает общение с ИИ более естественным, похожим на общение с человеком. Мы ведь тоже используем не только слова, но и жесты, изображения, тон голоса.

Это раньше промпты были только текстовыми. Сейчас появилась возможность комбинировать в одном запросе разные типы данных: текст, изображения, аудио и даже видео. Это можно назвать «промптами с приложением», когда вы не просто описываете словами, а показываете и рассказываете одновременно.

Например, Вы загружаете в ИИ фотографию нарисованного от руки эскиза логотипа, текстом пишете «Сделай этот логотип в стиле ретро, добавь синие оттенки» и добавляете голосовую заметку: «Хочу, чтобы он выглядел в стиле из 80-х». ИИ обрабатывает все эти данные вместе и выдаёт готовый, профессиональный логотип.

Этот тренд ведёт к созданию ИИ, который понимает нас в комплексе, а не только с помощью текста.

Автоматический инжиниринг промптов

Автоматический инжиниринг промптов – это системы, которые сами улучшают и оптимизируют запросы пользователя к ИИ. Если раньше приходилось вручную перебирать десятки формулировок, теперь за секунды это может делать «ИИ для ИИ».

Система автоматически тестирует различные варианты формулировок пользовательского промпта, чтобы найти самый эффективный. Это экономит колоссальное количество времени и даёт более качественные результаты. Даже эксперты тратят часы на подбор идеальных слов.

Например, загружаем в систему простой запрос: «Напиши рекламу кофе». Система автоматически генерирует несколько уточнённых промптов: «Напиши короткий рекламный текст для соцсетей про новый сорт эспрессо...», «Создай убедительный текст, который заставит молодых профессионалов купить наш кофе...». Затем она тестирует их и сообщает: «Вариант 7 даёт на 30% больше кликов. Использовать его?»

В будущем не нужно быть гуру промптинга — ИИ поможет формулировать запросы максимально эффективно.

Персонализированные шаблоны

Персонализированные шаблоны – возможность «обучить» ИИ своим предпочтениям и создать личные шаблоны запросов. По сути, это ваш собственный ИИ-ассистент, который знает ваш стиль. ИИ запоминает ваши любимые форматы, часто используемые фразы и профессиональную специфику. На основе этого создаёт персонализированные шаблоны промптов для ваших частых задач. Это делает работу с ИИ по-настоящему персональной. Промпты перестают быть просто инструкциями и становятся отражением уникального подхода пользователя.

Например, пользователь часто пишет email-рассылки. Он несколько раз показывает ИИ примеры, и он запоминает тон, привычную для пользователя структуру промпта («приветствие → проблема → решение → призыв к действию» или иную) и конкретные ключевые фразы. Теперь достаточно написать: «Напиши рассылку про новую функцию», а ИИ сам применит весь пользовательский стиль.

ИИ становится персональным помощником, который учится у пользователя и подстраивается под пользователя.

Промпты как код (Prompt-as-Code)

Промпты начинают работать как часть компьютерной программы, а не как разовые запросы пользователя. Происходит интеграция промптов в бизнес-процессы и программное обеспечение, т.е. текстовые команды для ИИ встраиваются в автоматизированные процессы. К ним начинают относиться так же серьёзно, как к обычному программному коду: проверяют тестами, сохраняют разные версии и контролируют любые изменения. Это превращает промптинг из инструмента для одиночек в мощную часть корпоративных систем и цифровых продуктов.

Например, в компании автоматизирована служба поддержки. Клиент пишет запрос, ИИ автоматически анализирует его с помощью заранее прописанного и протестированного промпта (например, «Проанализируй тон сообщения, выдели ключевую проблему и сгенерируй вежливый ответ в соответствии с гайдлайнами бренда»). Менеджер только проверяет и отправляет ответ.

Этот тренд показывает, что промптинг становится серьёзной профессией и неотъемлемой частью современных технологий, а не просто увлечением.

Промптинг сейчас похож на фотографию 20 лет назад. Тогда были только профессиональные камеры с кучей настроек. Сейчас любой может снимать отличные фото на телефон. Так и с промптами – сейчас нужно знать много нюансов, но со временем появятся простые инструменты, где всё настраивается автоматически. Но даже сейчас, зная основы, можно получать отличные результаты

Даже простое знание пяти принципов из этой главы даст вам преимущество. А инструменты – это просто способ сделать вашу работу ещё эффективнее.

§ 9. Зачем пользователю знать о токенах?

Давайте разберёмся с токенами, так как это важное знание и очень практичное для пользователя, которое поможет получать от нейросетей максимально полезные и полные ответы.

Представьте, что нейросеть читает текст не так, как мы, – по словам и предложениям. Вместо этого она разбивает весь запрос и свой будущий ответ на мельчайшие части, которые называются то́кенами.

То́кен – это базовая единица текста, которую обрабатывает модель. Это может быть:

целое слово – «кот»;

часть слова – «об» в слове «облако»;

знак препинания – точка, запятая;

пробел между словами.

Например, фраза «Привет, мир!» будет разбита примерно на 5 токенов:

1. «Привет»
2. «,»
3. «пробел»
4. «мир»
5. «!»

Модели так проще и быстрее «понимать» и генерировать язык.

Пользователю полезно знать о токенах, поскольку у каждой нейросети есть ограничение на общее количество токенов, которые она может «удержать в памяти» за один раз. Это называется контекстное окно.

Контекстное окно разделено на две части:

1. Ваш запрос (подсказка) – всё, что вы написали, включая инструкции, вставленный текст и историю чата.

2. Ответ нейросети – то, что она вам сгенерирует.

Важно! ***Сумма токенов запроса и ответа не должна превышать лимит контекстного окна модели.***

Что случится, если объем запроса и ответа превысит лимит контекстного окна модели:

Вариант 1. *Вы получаете обрезанный ответ.*

Причина: Запрос оказался слишком длинным (например, вы вставили для анализа целую статью). В контекстном окне просто не хватило места для ответа.

Что делать: Сократите запрос. Уберите лишнюю информацию или разбейте задачу на несколько частей.

Вариант 2. *Нейросеть «забывает» начало вашего диалога.*

Причина: В чате накопилось много сообщений, и их общая длина (в токенах) превысила лимит. Модель вынуждена «забыть» самое начало, чтобы запомнить конец.

Что делать: Начните новый чат для обсуждения новой темы или кратко подведите итог предыдущего диалога в новом сообщении.

Вариант 3. *Вы не можете вставить большой документ.*

Причина: Даже если вы не просите написать развёрнутый ответ, сам документ занимает почти всё контекстное окно.

Что делать: Разделите документ на части и обрабатывайте по частям.

Разные модели имеют разный «объём памяти». Вот ориентировочные данные:

Qwen – одна из самых мощных моделей в этом плане. Некоторые версии Qwen могут работать с контекстом до 128 000 токенов, а флагманские модели – даже до 262 144 токенов. Это позволяет обрабатывать очень длинные документы или вести продолжительные диалоги.

DeepSeek также предлагает большие окна. Например, DeepSeek Chat (V3.2) поддерживает до 128 000 токенов.

Perplexity AI использует модели с контекстным окном в 128 000 токенов, что позволяет ему эффективно анализировать длинные веб-страницы и документы, на которые вы ссылаетесь.

Для мультимодальных моделей (для генерации музыки и видео, таких как, например, Suno, Kling) понятие токена сложнее, так как они работают не только с текстом, но и с другими типами данных (аудио-, видеокадрами). Однако их текстовые подсказки всё равно подчиняются ограничениям, которые обычно находятся в том же диапазоне, что и у современных LLM (больших языковых моделей), – десятки тысяч токенов.

Как пользоваться этим знанием?

1. Оцените длину своего запроса. Если вы вставляете большой текст, помните, что на ответ останется меньше места и он может быть обрезан нейросетью.
2. Найдите информацию об ограничениях. Используя версию сервиса, уточните, какая именно модель используется и какой у неё лимит. Для платной версии сервиса это может быть актуально и в части оплаты дополнительных токенов, выходящих за пределы подписки.
3. Не бойтесь начинать новый чат. Это самый простой способ «очистить память» нейросети и предоставить ей максимум ресурсов для решения вашей новой задачи.

Теперь вы знаете главный секрет: токены – это валюта внимания нейросети. Чем экономнее вы их тратите, тем больше полезной информации можете получить в ответ!

ГЛАВА 6.
РОССИЙСКИЕ НЕЙРОСЕТИ: ОБЗОР И ПЕРСПЕКТИВЫ

Российский рынок искусственного интеллекта характеризуется высокой концентрацией и доминированием нескольких крупных игроков, которые определяют технологические тренды и динамику всего сектора. Консолидация вокруг отечественных платформ является основным рыночным трендом.

Ключевые игроки на рынке ИИ в России – это Яндекс, Сбер, Т-Технологии и VK.

Яндекс активно развивает собственную экосистему нейросетевых продуктов, интегрируя их во все свои сервисы.

Важной особенностью Сбера является создание многоуровневых решений для бизнеса: от открытых API (GigaChain) до корпоративных контейнеризированных версий (On-Prem) и полностью локализованных систем.

Т-Технологии, являясь третьим по величине игроком, специализируется на предоставлении комплексных IT-решений для бизнеса, имея широкую клиентскую базу в сегменте среднего и крупного предпринимательства. Компания развивает собственные языковые модели T-Lite и T-Pro и активно внедряет ИИ в своих продуктах, таких как «Нейрощит» и «Кибершквал». Внедрение чат-ботов на базе ИИ позволило автоматизировать до 45% обращений в поддержку клиентов.

На четвертом месте находится **VK**, чьи технологии активно используются в социальных сетях и медиаплатформах. Генеративный ИИ является фундаментальной частью таких продуктов, как ВКонтакте, Одноклассники, Дзен и VK Музыка, где он применяется для рекомендательных систем, распознавания речи и изображений, апскейлинга видео и модерации контента.

Другие значимые компании в сфере нейросетевых технологий – это:

Ozon – крупный игрок в e-commerce, который развивает собственные ИИ-модели для скоринга, логистики и рекомендательных систем.

Ростелеком – развивает платформу «АстроСофт» на базе GigaChat, предлагает облачные и ИИ-решения для госсектора и корпораций.

Газпром нефть – активный потребитель и одновременно разработчик ИИ-решений для добывающей промышленности (например, для предиктивного анализа месторождений).

МТС – развивает платформу MTS AI и входящую в её состав компанию VisionLabs, позиционирующую свою платформу LUNA AI как один из лучших решений в области распознавания лиц.

Digital Design – компания, занимающаяся разработкой и применением ИИ в проектах по созданию умных городов и городских систем. С их помощью реализуются проекты по оптимизации городской инфраструктуры, включая умный транспорт, системами сбора и анализа данных о потребностях жителей. Это позволяет городам становиться более адаптивными и эффективными в управлении ресурсами.

«Лаборатория Касперского» является пятым по величине игроком по выручке в сфере ИИ и ML, используя технологии в своих решениях по информационной безопасности.

Наносемантика – компания, разрабатывающая системы для анализа медицинских изображений на базе нейросетей. Её решения помогают врачам в диагностике заболеваний лёгких на основе анализа рентгеновских снимков и КТ.

Медскан – проект по разработке системы для диагностики кожных заболеваний с использованием нейросетей. Система анализирует фотографии кожи и выявляет признаки различных заболеваний, таких как меланома, псориаз и экзема.

Neuro.net – стартап, который сосредоточен на разработке решений для автоматизации и оптимизации бизнес-процессов с использованием технологий нейросетей. Компания применяет ИИ для прогнозирования потребительского спроса, персонализированных рекомендаций и оптимизации логистики. Neuro.net работает в различных отраслях, включая e-commerce, финансы и производство, позволяя клиентам достигать значительного роста и повышения качества обслуживания.

Доктор на работе – платформа, использующая нейросети для анализа электронных медицинских карт. Нейросети анализируют данные пациентов, выявляют закономерности и предоставляют врачам рекомендации по лечению.

Значительное количество российских компаний работают в сфере искусственного интеллекта. Для того, чтобы хотя бы примерно представить себе российский нейросетевой рынок можно обратиться, например, к разработанной RB.RU и SberUnity **интерактивной карте AI-стартапов**[40] – резидентов венчурного хаба SberUnity. Дополнит и расширит эту информацию **карта российских ИИ-сервисов**, представленная компанией «Инк»[41], а также

[40] Карта искусственного интеллекта: https://rb.ru/ai-map/

[41] Карта российского ИИ: как 130+ отечественных сервисов меняют бизнес https://incrussia.ru/specials/genai-map-russia/

«Карта игроков рынка ИИ России v.2.0» Центра технологий искусственного интеллекта «Нейролаб»[42], которая аккумулирует данные о более чем 900 организациях и позволяет выстраивать кооперационные цепочки, и «Карта российского GenAI и сопутствующих продуктов» компании «Технократия»[43].

Каждая из компаний и стартапов, упомянутых в этих картах, и те, кто еще в них появится (разработчики оставляют «двери открытыми» и приглашают компании заявлять о себе и готовы дополнять информацию), привносит свой уникальный вклад в развитие нейросетевых технологий и их применение в практике. Они берут на себя ответственность не только за экономическую эффективность, но и за создание культурных ценностей, этических стандартов и управление рисками, связанными с использованием ИИ. Таким образом, российский рынок ИИ демонстрирует модель «гибридного» развития с доминированием крупных цифровых экосистем (Яндекс, Сбер, VK) и активным ростом нишевых игроков, решающих отраслевые задачи – от медицины и безопасности до «умных городов». Такая структура, наряду с существенными государственными и частными инвестициями, закладывает фундамент для достижения национальных целей по технологическому суверенитету и вхождению России в число мировых лидеров в сфере искусственного интеллекта.

§ 1. Отличительные черты российских нейросетей

Локализация и поддержка русского языка: российские нейросети превосходят зарубежные аналоги в работе с русским языком и культурными особенностями:

Глубокое понимание контекста: YandexGPT анализирует морфологию, диалекты и даже иронию, что критично для задач вроде чат-ботов или анализа соцсетей; NeuroDream и Порфирьевич генерируют тексты с соблюдением русской грамматики и стилистики, включая подражание классическим авторам.

Мультиязычность: Kandinsky 3.1 поддерживает 100 языков, но приоритет отдаёт русскому, корректно обрабатывая запросы вроде

[42] Карта игроков рынка ИИ России v.2.0 https://neirolab.ru/project/karta-igrokov-ai?ysclid=mffi0zlm7b429097523
[43] Карта российского GenAI и сопутствующих продуктов https://technokratos.com/blog/40

«избушка на курьих ножках»; голосовой помощник Алиса распознаёт речь с акцентами и адаптируется к региональным выражениям.

Культурная адаптация: нейросети учитывают локальные реалии – от праздничных традиций до исторических событий. Например, Шедеврум (Яндекс) создаёт изображения «зимней Москвы в стиле Ван Гога», сочетая глобальные тренды с национальным колоритом.

Мультимодальность: GigaChat и YandexGPT развивают поддержку текста, изображений и аудио.

Интеграция в экосистемы: решения от Сбера и Яндекса встроены в их продукты (Салют, Алиса, Yandex Cloud).

Интеграция с госуслугами: нейросети внедряются в системы здравоохранения (например, СберЗдоровье для диагностики по снимкам МРТ) и логистики (беспилотные грузоперевозки на трассе М-11).

Этические ограничения: многие модели блокируют провокационные запросы, что снижает риски, но ограничивает креатив. В контексте развития технологий ИИ в России участники сообщества активно обсуждают этические и социальные аспекты использования ИИ. Учитывая специфические условия, социальные и культурные контексты, российские исследователи обращают внимание на вопросы, связанные с правами пользователей, защитой данных и предвзятостью алгоритмов. Это создаёт необходимость в разработке этических норм, которые помогут регулировать использование ИИ и обеспечивать безопасность пользователей.

Адаптация под локальные задачи: одной из значительных отличительных черт «русского» ИИ является его практическая ориентированность. Российские компании и исследователи сосредотачиваются на решении конкретных задач, с которыми сталкивается экономика и общество, таких как автоматизация производственных процессов, анализ данных и улучшение инфраструктуры. Это приводит к созданию решений, которые обладают высокой применимостью в повседневной жизни, начиная от моделей для прогнозирования спроса до систем распознавания лиц для обеспечения безопасности. YandexGPT обучен на огромном корпусе русскоязычных текстов, что делает его особенно эффективным для контента на русском языке; Kandinsky 3.1 (Сбер) генерирует изображения с учётом российской эстетики, например, стилизуя пейзажи под Шишкина или Васнецова.

Отличительные черты «русского» ИИ представляют собой сочетание практической направленности, сосредоточенности на этических и социальных аспектах, образования и подготовки кадров, а также преодоления технических барьеров и создания эффективного сотрудничества. Эти факторы формируют

уникальный ландшафт для дальнейшего развития ИИ в России, предлагая новые возможности для общества и экономики.

Рекомендации:

Для бизнеса: GigaChat и YandexGPT – оптимальный выбор для автоматизации.

Для творчества: Kandinsky и «Шедеврум» – для генерации изображений.

Для разработчиков: GigaCode и Yandex Cloud API – инструменты для интеграции ИИ.

§ 2. Яндекс

История развития ИИ в Яндексе – это путь длиною почти в два десятилетия. Всё началось в 2007 году с машинного обучения для улучшения поиска, а сегодня это – мощнейшие нейросети, встроенные в привычные сервисы

Яндекс не просто следил за трендами, а активно формировал их:

2007 –первое применение машинного обучения для ранжирования поиска.

2013 – рождение речевых технологий (SpeechKit), ставших фундаментом для Алисы.

2017 – официальный дебют Алисы – одного из первых голосовых помощников, способного вести свободный диалог, а не просто реагировать на набор команд.

2021 – создание языковой модели YaLM (предшественница YandexGPT) с 100 млрд параметров.

2023 – запуск YandexGPT и YandexART – нейросетей для генерации текстов и изображений.

2024 – появление Нейро в поиске – объединение возможностей поиска и YandexGPT для ответов с проверенными источниками.

Этот путь показывает, что ИИ в Яндексе – не просто игрушка, а технология, встроенная в экосистему сервисов.

Вам не нужно искать специальные программы или платить подписки (для старта). Всё уже в вашем телефоне или браузере.

После реструктуризации 2024 года международные активы (включая часть облачных технологий) выделены в Nebius Group. Yandex LLC в России сохранила развитие ключевых продуктов, включая YandexGPT и Алису.

Яндекс сотрудничает с ведущими российскими вузами (ВШЭ, МФТИ, Сколтех).

Основные технологии-флагманы

YandexGPT – обработка текста и генерация контента.

YandexART – создание изображений.

SpeechKit – распознавание и синтез речи.

Специализированные технологии

В Яндекс.Браузере применяются технологии умных ответов, коррекции текста и генерации заголовков, на Яндекс.Диске – умный поиск по содержимому, классификация файлов и автоматическое тегирование, а в Яндекс.Почте – сортировка писем, автоответы и проверка орфографии, Яндекс.Переводчик включает более 100 языков.

Бизнес-решения

Нейросетевые решения Яндекс доступны и для бизнеса. Например, Yandex Cloud даёт доступ:

– к API нейросетей – это как пульт управления умными программами, то есть вы можете подключить их к своему сайту или приложению, чтобы они помогали вам работать с текстами, изображениями или данными,

– кастомизированным моделям – программам, которые можно настроить именно под задачи пользователя, например, если пользователь продаёт одежду, то можно создать модель, которая будет идеально понимать запросы его клиентов,

– инфраструктуре для ML – это всё необходимое оборудование и программы для работы с искусственным интеллектом, то есть вам не нужно покупать свои сервера – всё уже готово в облаке.

Яндекс.Диалоги (чат-боты, виртуальные ассистенты, автоматизация поддержки) – это виртуальный помощник, который круглосуточно отвечает на вопросы клиентов компаний. При этом, чат-боты как автоматические операторы отвечают на часто задаваемые вопросы в мессенджерах или на сайте (например, могут помочь клиенту оформить заказ или ответить на вопросы о товарах). Виртуальные ассистенты – это более продвинутые помощники, которые уже могут вести сложные диалоги, решать проблемы клиентов и даже давать персональные рекомендации. Система автоматизация поддержки помогает службе поддержки организаций-пользователей работать быстрее и эффективнее (сортирует входящие запросы, отвечает на простые вопросы, передает сложные вопросы живым операторам, собирает статистику по обращениям и т.п.).

Как это работает на практике. Например, у вас Интернет-магазин. Вы подключаете Яндекс.Диалоги, и теперь чат-бот отвечает на вопросы клиентов о наличии товаров, виртуальный ассистент помогает с оформлением заказа, система автоматически сортирует обращения по важности. И всё это работает 24/7 без перерывов. Хотите большего – подключаете Yandex Cloud и у вас есть возможность обучить модели на своих данных, то есть настроить нейросети под специфику конкретного бизнеса и получать аналитику и отчёты в реальном времени.

Единая платформа

Все нейросети Яндекса работают на единой платформе, что делает их мощными и удобными. Это позволяет:

быстро улучшать сервисы – обновление одного модуля (например, YandexGPT) сразу улучшает работу всех сервисов, где он используется (Алиса, Поиск, Переводчик);

экономить ресурсы – общая техническая база делает работу нейросетей эффективнее;

легко интегрироваться – разработчики могут легко подключить нейросети к своим проектам через стандартные интерфейсы (API).

Главное преимущество этих решений в том, что пользователю не нужно быть специалистом по искусственному интеллекту – всё настроено и готово к использованию.

Алиса – нейросеть-эрудит

Что это – умный собеседник, который понимает контекст, помнит ваши предыдущие реплики в диалоге и генерирует тексты, изображения, видео на любую тему с учетом некоторых этических ограничений (Таблица 27).

Где «живет»:

в браузере – откройте Яндекс Браузер или главную страницу Яндекса (ya.ru), нажмите на значок Алисы (фиолетовый кружок) и выберите опцию «Алиса, давай придумаем»;

в колонке/приложении – просто скажите: «Алиса, давай придумаем»;

отдельный чат – перейдите на сайт alice.yandex.ru;

приложение «Яндекс – с Алисой».

Ограничения и проблемы:

– Может «фантазировать». YandexGPT, как и любая нейросеть, иногда «генерирует галлюцинации» – то есть выдаёт непроверенную или ложную информацию как факт. Всегда перепроверяйте важные факты.

– Чувствительна к формулировкам. Один вопрос может поставить её в тупик, а перефразированный – позволит получить блестящий ответ.

– Бывает многословной. Любит воду и шаблонные фразы.

Таблица 27. «Что может YandexGPT?»

Возможности	Как это использовать?	Пример промпта (что написать в запросе)
Написание текстов	Письма, посты, стихи, сценарии, идеи.	«Напиши поздравительное письмо для коллеги с днём рождения в дружеском тоне».
Решение задач	Математика, программирование, бизнес-планы.	«Распиши решение уравнения $x^2-5x+6=0$ по шагам».
Работа с информацией	Краткий пересказ статей, выделение главного.	«Перескажи кратко статью на этой странице» (есть кнопка в браузере).
Творчество	Сочинение историй, шуток, креативных идей, генерация изображений и видео.	«Придумай идею для романа в жанре космической оперы», «Нарисуй…», «Оживи картинку».
Перевод	Перевод с учётом контекста и идиом.	«Переведи на английский: „Их было трое, но это не точно“».

Шедеврум – нейросеть-художник

Что это – генерация изображения и видео по текстовому запросу (Таблица 28).

Где «живет»:

Скачайте приложение Шедеврум (Shedevrum) на iOS или Android или найдите его в веб-версии на сайте shedevrum.ai

Ограничения и проблемы:

– Сложность с абстракциями. Может не понять запросы вроде «нарисуй надежду» или «изобрази символизм».

– Руки и лица. Как и многие AI-художники, иногда странно рисует анатомию человека (особенно руки).

– Требует деталей. Чем конкретнее и образнее запрос, тем лучше результат. «Красивая картинка» – не сработает.

Таблица 28. «Что может YandexART?»

Возможности	Как это использовать?	Пример промпта (что написать в запросе)
Генерация изображений	Создание иллюстраций, обложек, артов.	«Кот в костюме детектива, читает книгу при свете настольной лампы, в стиле нуар».
Создание видео	Короткие анимации по описанию.	«Летящий над облаками единорог, анимация».
Наложение стилей	Превращение ваших фото в арт-объект.	Загрузите фото и выберите фильтр «под Ван Гога».

Нейросети для перевода – полиглот-синхронист

Что это – система-переводчик, понимающая контекст, идиомы и даже переводящая видео в реальном времени с озвучкой.

Где «живет»:

Яндекс Браузер – при посещении иностранного сайта он предложит перевод. Для видео на YouTube включите субтитры и перевод в настройках плеера.

Приложение «Яндекс Переводчик» – для перевода текстов и разговоров с более чем 100 языков.

Что умеет:

– Переводит и озвучивает видео с английского, немецкого, китайского и других популярных языков прямо в браузере.

– Переводит текст на картинках через Умную камеру. Просто наведите объектив на вывеску или меню.

Ограничения и проблемы:

– Специфическая лексика, т.е. может ошибиться в узкопрофессиональных или редких терминах.

– Потеря нюансов – юмор, сарказм и сложные поэтические метафоры иногда теряются при переводе.

§ 3. Сбербанк

Развитие искусственного интеллекта в Сбере – это стратегический курс, интегрированный в цифровую экономику России. Если Яндекс начинал с поиска, то Сбер – с данных о финансах и клиентах, что стало мощным фундаментом для его AI-империи.

Сбер не просто адаптировал мировые тренды, а начал создавать собственную экосистему:

апрель 2023 – официальный анонс GigaChat – многофункциональной нейросети, главного конкурента ChatGPT на русскоязычном пространстве;

ноябрь 2023 – запуск Kandinsky 3.0 и Kandinsky Video – нейросетей для генерации изображений и видео, глубоко обученных на русской культуре;

2024 – постоянные обновления моделей, развитие мультимодальности (работа с текстом, изображением, звуком) и глубокая интеграция во все сервисы экосистемы: от банкинга и «СберСтрахования» до умных колонок «Салют».

Этот путь показывает, что ИИ в Сбере – не эксперимент, а ключевая часть стратегии компании, встроенная в российские реалии.

В состав нейросети входят несколько модулей, объединённых в ансамбль под названием NeONKA (NEural Omnimodal Network with Knowledge-Awareness):

ruGPT-3 – российский аналог GPT-3, разработанный SberDevices. Обучен на 13 миллиардах параметров.

FRED-T5 – языковая модель, основанная на архитектуре T5 от Google. Предназначена для генерации текста и выполнения задач по обработке естественного языка для русскоязычного сегмента. Имеет 1,7 миллиарда параметров и 24 слоя.

ruCLIP – адаптация модели CLIP для русского языка, обученная на парах «изображение-текст». Используется для анализа визуального контента и его соответствия текстовому описанию.

Kandinsky – генерация изображений по текстовому описанию.

Основные технологии-флагманы

GigaChat – мультимодальная модель для работы с текстом, кодом, изображениями и видео.

Kandinsky – генерация и редактирование изображений и видео.

GigaCode – AI-ассистент для программистов.

Visper – создание видео с цифровыми аватарами.

Где живут эти технологии

Они встроены в привычные сервисы: приложения СберБанка и Салют, сайты, а также доступны через чат-боты в Telegram и ВКонтакте.

Ключевой принцип – глубокая интеграция. Одна модель, например, GigaChat, улучшает работу десятков сервисов – от анализа документов в банке до голосового помощника в умной колонке.

Главное преимущество экосистемы Сбера – комплексность. Вам не нужно быть специалистом по ИИ. GigaChat, Kandinsky и другие инструменты уже встроены в приложения, которые миллионы людей используют каждый день, или доступны в несколько кликов.

GigaChat – нейросеть-универсал

Что это: умный помощник, который понимает контекст, помнит историю диалога и способен генерировать тексты, решать задачи, писать код и создавать изображения (через интеграцию с Kandinsky). Его ключевое преимущество – глубокое понимание русского языка и российского контекста (Таблица 29).

Где «живет»:

– веб и мобильное приложение – на официальном сайте Giga.Chat или в приложениях Сбера;

– в соцсетях – чат-боты в Telegram (@gigachat_bot) и ВКонтакте.

– в умных устройствах– в колонках «Капсула» и приложении «Салют» (команда: «Включи GigaChat»).

Ограничения и проблемы:

– Требует регистрации – полный функционал доступен после авторизации через СберID.

– Может «фантазировать» – всегда перепроверяйте важные факты и код.

– Файлы в одном чате – каждый новый документ нужно загружать в новом окне диалога.

Таблица 29. «Что может GigaChat?»

Возможности	Как это использовать?	Пример промпта (что написать в запросе)
Написание текстов	Статьи, письма, слоганы, стихи, сценарии.	«Напиши официальное письмо партнерам о переносе встречи на следующую неделю».
Решение задач	Математика, логические задачки, аналитика.	«Реши уравнение $x^2-5x+6=0$ и распиши решение по шагам».
Программирование	Написание, комментирование, исправление кода.	«Напиши на Python функцию для вычисления чисел Фибоначчи».
Работ с файлами	Анализ и Summarize документов (PDF, DOCX, TXT)	Загрузи файл и напиши: «Выдели ключевые тезисы из этого отчёта».
Пересказ видео	Анализ роликов с YouTube, RuTube, VK Видео.	«Перескажи ключевые идеи из видео по этой ссылке: [ссылка]».
Генерация изображений	Создание картинок через интеграцию с Kandinsky.	«Нарисуй космонавта, играющего на балалайке на фоне храма Василия Блаженного».

Kandinsky – нейросеть-художник

Что это: генератор изображений и видео, который знает специфику русской культуры (от гжели до советского модерна).

Где «живет»:

– официальный сайт fusionbrain.ai – здесь самый полный функционал.

– Telegram – боты @Kandinsky_by_Sber_AI (картинки) и @Kandinsky_Video_by_Sber_AI (видео).

– ВКонтакте – через одноимённого бота.

Что умеет:

– Генерация изображений по описанию («Изобрази дом в стиле русского модерна в осеннем лесу»).

– Изменение стиля фото («Сделай это фото в стиле Малевича»).

– Создание видео и анимации по текстовому запросу.

Ограничения и проблемы:

– Экспериментальное видео – генерация видео может выдавать сырые и неожиданные результаты.

– Некоммерческая лицензия – изображения с открытых платформ нельзя использовать в коммерции.

– Сложность с абстракциями – может не понять запросы вроде «нарисуй любовь к Родине».

GigaCode

Что это: AI-ассистент для программистов или умный помощник-программист от Сбера, который не пишет программы за вас с нуля, но значительно ускоряет и упрощает разработку кода, предлагая подсказки, генерируя фрагменты по описанию и помогая избегать ошибок. Он для того, чтобы программист тратил меньше времени на рутину и больше – на сложные задачи.

Где «живет»:

– на сайте gitverse.ru

– работает как плагин (дополнение) в популярных средах разработки, таких как:

- Visual Studio Code (VS Code)
- JetBrains IDE (PyCharm, WebStorm, IntelliJ IDEA и др.)
- Jupyter Notebook
- и другие.

Вы устанавливаете плагин, и он начинает работать прямо в вашем редакторе кода.

Главное преимущество – не нужно переключаться на отдельный сайт или копировать код туда-сюда. GigaCode помогает вам прямо в процессе написания кода, как настоящий напарник через плечо.

Что умеет?

– Автодополнение кода (как Т9 в телефоне, но для программистов) – программист начинает писать команду, а GigaCode предлагает несколько вариантов её окончания, программист выбирает подходящий и он немедленно вставляется.

– Генерация кода по описанию на русском или английском – описываете, что должна делать функция, а GigaCode её пишет, т.е. достаточно

написать комментарий к функции и дать ему команду её сгенерировать (например, «#напиши функцию на Python, которая принимает список чисел и возвращает его сумму».

– Написание комментариев и объяснений – GigaCode может автоматически комментировать сложный код, чтобы его было легче понять вам или другим разработчикам (для этого необходимо выделить код и попросить написать промпт «#объясни, что делает этот код».

– Поиск и исправление ошибок – GigaCode может подсказать, где в вашем коде возможна опечатка или логическая ошибка и предложить вариант исправления (если код не работает, нужно спросить у GigaCode, в чём может быть проблема).

– Создание тестов – GigaCode может автоматически сгенерировать код для проверки написанной программистом функции (unit-тесты), для этого нужно после функции написать промпт «#напиши unit-тест для этой функции».

– Поддержка десятков языков программирования – работает с Python, Java, JavaScript, C++, C#, PHP, Go, SQL и многими другими. Не нужно переключаться между разными помощниками.

Ограничения и проблемы:

GigaCode, как и любой AI-ассистент разработчика, имеет ряд существенных ограничений и проблем, которые важно учитывать при его использовании. Основные из них связаны с качеством генерации кода, безопасностью, интеграцией и контекстным пониманием. Поэтому весь сгенерированный код требует тщательного проверки и тестирования разработчиком.

GigaCode – мощный инструмент, но его эффективность напрямую зависит от экспертизы разработчика и качества контроля. В будущем ожидается улучшение моделей по всем указанным направлениям, но человеческий фактор останется ключевым.

Visper

Что это: сервис для создания видео с цифровыми аватарами, которые произносят текст, который введен пользователем. Интегрируется с CRM (Customer Relationship Management system, Система управления взаимоотношениями с клиентами).

Где «живет»: на сайте https://visper.tech, бот-помощник – @visper_support_bot

Что умеет:

– Создает видео с виртуальными персонажами – предлагаются несколько готовых аватаров обоего пола, возможна настройка внешнего вида и голоса, персонажи могут использовать жесты, а их мимика синхронизируется с текстом или загруженной аудиодорожкой.

– Преобразовывает текст и презентации в видео – загружаем текст или PDF, платформа позволяет превращать готовые презентации (например, в формате PDF) или текстовые сценарии в видео с виртуальным ведущим, текст зачитывается выбранным голосом с возможностью расстановки пауз, ударений и управления темпом речи.

– Интерфейс включает разделы для управления проектами и тонкой настройки роликов – изменения фона, масштаба персонажа, добавления музыки или видеофрагментов, можно загружать собственные аудиозаписи, под которые система автоматически подстроит мимику персонажа.

– Поддерживает 9 языков и т.д.

Ограничения и проблемы:

В бесплатном тарифе на видео добавляется логотип Visper, а скачивание готовых роликов запрещено. Бесплатно можно сгенерировать только 2 минуты видео в месяц. Процесс создания видео может занимать 30–40 минут, а иногда завершается с ошибкой, особенно при использовании бесплатной версии.

Пользователи отмечают, что выбор виртуальных персонажей (аватаров) пока небольшой. Не хватает разнообразия внешности, возрастов, стилей и анимаций. Движения и мимика персонажей иногда выглядят механически и недостаточно плавно.

Голосовые модели тоже ограничены: хотя есть варианты интонаций (нейтральная, деловая), количество тембров и эмоциональных оттенков недостаточное для сложных задач. Голос иногда звучит роботизировано, несмотря на настройки интонации. Синхронизация губ с речью может быть неидеальной, особенно для сложных слов или быстрого темпа. Для озвучки текста доступно только 200 символов, а максимальное количество слайдов в презентации – 6.

Сервис может работать нестабильно: встречаются сбои в обработке файлов или зависания интерфейса. Служба поддержки реализована только в рамках чат-бота в Telegram, что не всегда удобно для оперативного решения проблем, нет возможности связаться с живым специалистом или получить подробную консультацию.

§ 4. Т-Технологии

Т-Технологии (входят в экосистему Т-Банка, ранее Тинькофф Банк) активно развивают направление искусственного интеллекта и больших языковых моделей (LLM). Нейросетевая экосистема Т-Технологий – это мощные языковые модели (T-Pro, T-Lite), оптимизированные для русского языка, комплекс инструментов для автоматизации разработки и аналитики, открытая платформа для бизнеса и разработчиков, позволяющая экономить ресурсы, фокус на практическое применение в финансовом секторе и IT.

Языковые модели (LLM)

T-Pro (32 млрд параметров) – мощная и экономичная модель для русскоязычных задач, но предназначена в первую очередь для разработчиков и компаний. Неподготовленный пользователь вряд ли сможет использовать её напрямую, но может столкнуться с ней в повседневных сервисах (например, в банковских чат-ботах). В отличие от меня, T-Pro 2.0 предлагает глубокую специализацию под кириллицу и гибридный режим рассуждений, но требует технических знаний для запуска и настройки.

T-Lite (7 млрд параметров) – предназначена для дообучения под конкретные бизнес-задачи, показывает высокую точность и адаптивность для отраслей: финансы, медицина, ритейл. Это также прежде всего инструмент для разработчиков.

Т-Банк бесплатно предоставляет модели T-Pro и T-Lite для бизнеса и разработчиков, чтобы помочь компаниям экономить на разработке собственных моделей и избежать комиссий за проприетарные ИИ-решения, т.е. доступные только через платные API или веб-интерфейсы.

ИИ-инструменты для разработчиков

AI-search – поиск информации для ИТ-команд (баги, архитектурные решения).

AI-Data – помощник для аналитиков (генерация SQL-запросов, ETL-скрипты).

Autodesc – генерация описаний таблиц и полей для дата-каталогов.

T-Cover Agent – генерация unit-тестов и повышение тестового покрытия.

T-Weaver – анализ документации и создание тестовых сценариев.

T-Code Review – автоматизация код-ревью (100% репозиториев).

Safeliner – ИБ-ассистент для проверки безопасности кода.

SRE-ассистент – автоматизация мониторинга и документирования.

Эти инструменты используются 80% инженеров Т-Банка (более 10 тыс. пользователей в месяц) и ускоряют разработку в 2–3 раза.

Агентский режим для разработки

Этот режим запущен в 2025 году на базе модели Qwen3-Coder-480b) и является продвинутым инструментом на основе искусственного интеллекта, который действует как «цифровой коллега» для программистов. Он не просто предлагает отдельные фрагменты кода, а самостоятельно выполняет сложные задачи на основе инструкций разработчика. Например, он может проанализировать проект, создать файлы, настроить утилиты или проверить код, учитывая контекст всего проекта. Другими словами, это виртуальный помощник, который понимает задачу целиком, работает с большими проектами (может анализировать тысячи строк кода и документации, учитывая связи между файлами), интегрируется в инструменты разработки (работает прямо в среде программирования, понимая команды на естественном языке (русском или английском) и автоматизирует рутину (например, запуск тестов, код-ревью или настройку окружения, что раньше занимало дни, теперь делается за минуты).

Преимущества для бизнеса и разработчиков: скорость разработки (сокращает время на рутину на 20–40%), доступность (упрощает работу новичкам и освобождает опытных разработчиков для сложных задач), массовое внедрение (в «Т-Банке» им уже пользуются 80% инженеров – более 10 000 человек).

§ 5. VK

VK – делает ставку на искусственный интеллект и нейросети, чтобы сделать свои сервисы умнее, удобнее и безопаснее для обычных людей, авторов и бизнеса. Вот как это устроено.

Нейросетевая экосистема VK – это комплекс технологий, встроенных в привычные сервисы VK (ВКонтакте, ОК, VK Видео, VK Музыка и др.). Эти нейросети работают «под капотом», автоматически анализируя огромные

объемы данных, чтобы предугадать ваши желания, защитить от неприятностей или показать именно тот контент, который вам понравится.

Экосистема решает четыре ключевые задачи для разных аудиторий:

для пользователей – сделать общение и потребление контента более персонализированным, безопасным и комфортным;

для авторов и блогеров – помочь создавать популярный контент и находить свою аудиторию, упростить монетизацию;

для бизнеса – максимально упростить настройку эффективной рекламы и помочь в анализе больших данных для принятия решений.

Ключевые нейросети и продукты VK

Мессенджер Max max.ru – это новый ключевой инфраструктурный проект продукт экосистемы и «точка входа» в цифровую экосистему VK для миллионов пользователей. Назначение – универсальная коммуникация и экосистема. Объединяет в себе общение (чаты, звонки), интеграцию с госуслугами (получение кодов для входа, цифровая подпись), платежи через СБП, чат-боты и мини-приложения для бизнеса, а также встроенный ИИ-ассистент GigaChat для генерации текстов и изображений.

Целевая аудитория – все пользователи в России. Активное использование Max будет генерировать огромные объемы новых данных о пользовательском поведении, что, в свою очередь, может использоваться для обучения и улучшения других нейросетевых моделей VK.

Технология «Персональные рекомендации» – встроена в основную ленту новостей VK. Откройте приложение или сайт VK и перейдите в раздел «Актуальное». Нейросеть анализирует миллионы постов в секунду, чтобы показать главные тренды и новости дня в удобном формате сюжетов.

Технология «Личное пространство» – обеспечивает безопасность и комфорт, защиту от токсичных личностей. Настройка активируется автоматически при обнаружении угрозы. Управлять настройками приватности можно здесь: Настройки VK → Приватность → «Личное пространство». Умный алгоритм следит за комментариями на странице пользователя. Если он фиксирует всплеск оскорблений от незнакомцев, то предложит включить режим «Личное пространство», который на неделю ограничит круг общения только друзьями.

Автотаргетинг в VK Рекламе – функция доступна при создании рекламной кампании в рекламном кабинете. Перейдите в VK Реклама (ads.vk.com), создайте кампанию и выберите цель «Привлечение подписчиков»

или «Посещение сайта» – система предложит использовать автоматический подбор аудитории. При выборе соответствующей цели (например, «Привлечение подписчиков») нейросеть автоматически подбирает аудиторию для рекламы на основе анализа поведенческих данных и успешных рекламных кампаний.

Соцсеть «Одноклассники» ok.ru – в ОК интегрированы общие технологии VK: лента VK Клипов, рекламные инструменты (ОРД), VK Mini Apps для разработчиков. Это позволяет распространять нейросетевые продукты на более широкую и возрастную аудиторию.

VK Cloud и Cloud ML Platform (основной сайт облачной платформы: cloud.vk.com; раздел с ML-решениями: https://cloud.vk.com/solutions/ml) – это «мозги» экосистемы. Любой человек или организация (например, учёные) могут арендовать вычислительные мощности VK для своих проектов. Например, чтобы быстро посчитать пеликанов на снимках с дрона для сохранения популяции.

Совместная школа с НИУ ВШЭ (официальный сайт образовательных проектов VK: https://education.vk.com/, раздел «Высшее образование») – VK готовит новых специалистов, чтобы нейросетей становилось еще больше, и они были лучше. Студенты работают над реальными задачами VK.

VK Data Platform (информация для бизнеса и подключения доступна на сайте: https://data-platform.vk.com/ или через раздел для бизнеса на основном сайте VK) – это «супермозг» для крупного бизнеса (банков, ритейла). Платформа помогает компаниям организовать все свои данные, анализировать их и внедрять собственные AI-модели для прогнозов и принятия решений.

Нейросеть в Почте, Облаке, Заметках (функции интегрированы непосредственно в сервисы: Почта mail.ru, Облако cloud.mail.ru, Заметки notes.mail.ru) – нейросеть составляет краткий пересказ писем в Почте, помогает создать текст на заданную тему в Облаке и Заметках, генерирует идеи для постов и креативные поздравления в Календаре.

Нейросеть в VK Рекламе (доступно в интерфейсе создания объявления в VK Реклама ads.vk.com) – при создании объявления нейросеть автоматически генерирует варианты заголовков, описаний и изображений на основе текстового описания продукта, предоставленного рекламодателем.

Подключение чат-бота с ИИ (услуги по подключению предлагают сторонние сервисы, например, сообщество «Подключите чатбота с нейросетью в VK и TG!» (vk.com/chatbotai_bot) – позволяет создать в группе VK или Telegram чат-бота на основе AI (например, ChatGPT/GPT-4), который сможет общаться с пользователями.

У экосистемы несколько слоев аудитории, как матрешка:

– каждый пользователь ВКонтакте, ОК, VK Видео невольно пользуется благами нейросетей каждый день – через ленту рекомендаций, умный поиск или защиту от спама.

– создатели контента (авторы) – блогеры, медиа, артисты, паблики, нейросети помогают им находить аудиторию, тренды и монетизировать свое творчество;

– предприниматели и бизнес – нейросети VK Рекламы и аналитики экономят время, нервы и бюджет на продвижении;

– корпорации и разработчики – крупные компании используют облачные AI-инструменты VK (VK Cloud, VK Data Platform) для своих внутренних задач;

– ученые и некоммерческие организации – используют доступные технологии VK для решения социально-значимых проблем.

Особенности экосистемы

– Нейросети глубоко вшиты в привычные сервисы.

– Защита психики пользователей (режим «Личное пространство»).

– Нейросети связывают все сервисы VK: соцсети, образование (Skillbox, GeekBrains), развлечения (VK Play), платежи (VK Pay) и т.д.

– Через VK Cloud сложные нейросетевые инструменты становятся доступны маленьким стартапам или ученым, у которых нет своих суперкомпьютеров.

§ 6. Нейросетевые продукты других российских разработчиков

Следует сразу отметить, что нейросетевых продуктов и сервисов очень много и постоянно появляются новые. Представляю сервисы, которые опробованы лично и успешно используются для решения разных задач.

Gerwin.io

Gerwin.io – российская платформа для генерации текстового и визуального контента с помощью искусственного интеллекта. Сервис ориентирован на русскоязычных пользователей, преимущественно из России

и СНГ и позиционирует себя как инструмент для автоматизации создания маркетинговых, коммерческих и информационных материалов. Сервис развивается как независимый продукт под брендом Gerwin AI, активное развитие и обновления фиксируются с 2023-2024 годов.

Целевая аудитория – предприниматели и малый бизнес (для автоматизации создания контента и описаний товаров), маркетологи и копирайтеры (для быстрой генерации идей, текстов и рекламных материалов), веб-мастера и SEO-специалисты (для создания SEO-оптимизированного контента), агентства (для работы с множеством клиентов и унификации контента).

Бот-помощник – @GerwinPromoBot

Что умеет:

– Текстовая генерация, включая создание коммерческих текстов (описания товаров для маркетплейсов, написание постов для соцсетей, текстов для email-рассылки), SEO-оптимизацию (генерация заголовков, мета-описаний, статей и лонгридов с учетом ранжирования в поисковых системах), создание структурированных маркетинговых материалов, рерайт и обработку (переработку существующих текстов, включая обработку контента из видео YouTube).

– Генерация изображений, включая создание иллюстраций для блогов, рекламы, соцсетей на основе текстовых запросов, поддержку стилей (например, «Startup», «History») для адаптации под бренд, но сервис испытывает трудности с генерацией сложных сцен (руки, много персонажей), точным воспроизведением букв или логотипов.

– Интеграции и автоматизация, предусматривающие API (возможность подключения для автоматической генерации описаний товаров, статей и других текстовых блоков без ручного вмешательства) и формирование базы знаний (загрузка данных о компании, товарах или услугах для персонализации контента).

Ключевые особенности и преимущества:

– Безлимитная генерация текстов: для пользователей с подпиской PRO кредиты за текстовые инструменты не списываются.

– Персонализация через «Базу знаний»: возможность загружать данные о компании, чтобы нейросеть учитывала их при генерации.

– Поддержка русского языка: все инструменты оптимизированы под русскоязычный контент и специфику местного рынка.

– Интеграция с популярными платформами: генерация контента для VK, Telegram, Ozon, YouTube.

– Для получения максимальной пользы от Gerwin.io важно четко формулировать запросы и использовать «Базу знаний» для персонализации. Однако для задач, требующих высокой визуальной точности или работы с данными, которые нейросети неизвестны, могут потребоваться дополнительные усилия или другие инструменты.

Ограничения и проблемы:

– Генерация изображений – неточности в прорисовке рук, сложных сцен с множеством персонажей, невозможность точно сгенерировать текст или логотип.

– Контекст и точность – может не знать о локальных или малоизвестных бизнесах без предварительной загрузки данных в «Базу знаний», поэтому требует четких и детальных запросов для качественного результата.

– Условия использования – безлимитная генерация текстов доступна только для PRO-аккаунтов, для изображений и API кредиты списываются.

– Запрещено одновременное использование одного аккаунта на нескольких устройствах.

Colorize

Colorize – российская нейросеть преобразует черно-белые фотографии в цветные. Сервис подходит для работы с видеомонтажом, анимацией или фотографией.

Что умеет:

– Раскрашивание черно-белых фото и видео.

– Реставрация поврежденных изображений.

– Генерация видео.

– Создание AI портрета (можно загрузить любую чёрно-белую или цветную фотографию, на которой изображён один человек, и после обработки будет сгенерирован воссозданный портрет в сверхвысоком разрешении 2048x2048)

– Увеличение разрешения.

Ключевые особенности и преимущества:

– Высокое качество обработки.

– Инновационный алгоритм восстановления.

– Создает «новые» изображения на основе оригиналов.

– Акцент на русские исторические фото.

– Тарифы: Бесплатный пробный доступ, платные пакеты.

Ограничения и проблемы:

– Стоимость платных тарифов от ₽600

– Мобильная версия: Нет.

– Ограниченная скорость обработки в бесплатной версии.

Главред

Главред – онлайн-сервис для проверки и улучшения текстов, ориентированный на повышение ясности, чистоты и соответствия информационному стилю. Сервис полезен для авторов, редакторов, маркетологов и всех, кто работает с текстами, стремясь сделать их более понятными и эффективными.

Что умеет:

– Проверка текста на «словесный мусор» – анализирует текст и выделяет слова или конструкции, которые затрудняют чтение или нарушают ясность.

– Оценка соответствия информационному стилю – проверяет, насколько текст лаконичен и понятен, удаляя лишние элементы.

– Предлагает варианты исправления проблемных мест.

– Интеграция с браузером – возможность использования в виде расширения для удобной проверки в режиме реального времени.

– Образовательный контент – рассылка с уроками и советами по созданию сильных текстов.

Ключевые особенности и преимущества:

– Ориентация на информационный стиль.

– Помогает убрать из текста всё лишнее, делая его более понятным и соответствующим нормам информационного стиля.

– Интуитивный интерфейс: пользователь вставляет текст и мгновенно получает оценку с рекомендациями.

– Регулярная рассылка с советами и примерами помогает пользователям улучшать свои навыки написания текстов.

– Основной функционал сервиса доступен без оплаты.

Ограничения и проблемы:

– Может быть менее полезен для художественных или креативных текстов.

– Не включает проверку орфографии, уникальности или SEO-параметров.

– Нет возможности интеграции с другими сервисами для автоматизации проверки.

– Субъективность рекомендаций.

Тургенев

Тургенев – это лингвистический веб-сервис для комплексного анализа текстов, мощный и узкоспециализированный инструмент для профессионалов в области SEO и контент-маркетинга, которые работают в условиях российского поискового рынка и ориентируются на Яндекс. Основная цель сервиса – помочь веб-мастерам, SEO-специалистам, копирайтерам и редакторам оптимизировать текстовый контент под требования поисковых систем, сохраняя при этом его естественность и читабельность для людей. Сервис оценивает риск попадания текста под фильтры поисковых систем и предоставляет конкретные рекомендации по исправлению проблем.

Что умеет:

Сервис предоставляет комплексный анализ текста по шести основным параметрам:

– Общий риск – интегральная оценка вероятности попадания текста под фильтры (выражается в баллах от 0 до 13+).

– Повторы – анализирует частоту повторения слов, словосочетаний и союзов (особенно союз «и»), вычисляет «академическую» и «классическую тошноту», выявляет сверхчастотные слова.

– Стилистика – проверяет текст на наличие стилистических ошибок, канцеляризмов, SEO-штампов, громких слов и неуместных выражений с помощью собственного словаря (порядка 29 000 терминов). Это единственная полностью бесплатная функция сервиса.

– Запросы – выявляет и анализирует использование ключевых слов и фраз (включая длинные хвостатые запросы), помогает бороться с переоптимизацией.

– Водность – определяет процент «воды» в тексте – стоп-слов, обобщений, лишних определений и словосочетаний, не несущих смысловой нагрузки.

– Удобочитаемость – оценивает сложность восприятия текста на основе средних длин предложений и слов, используя адаптированный

для русского языка Automated Readability Index (формула для определения того, насколько легко текст будет воспринят читателем, ARI приблизительно указывает на необходимый для понимания текста уровень образования, например, «9 класс» или «1-ый курс университета»).

Ключевые особенности и преимущества:

– Ключевое преимущество – глубокая проработка именно под алгоритм «Баден-Баден. Алгоритмы сервиса специально обучены на данных с сайтов, попавших под фильтр, поэтому он точнее многих аналогов предсказывает риски и дает целевые рекомендации по их устранению.

– Наличие бесплатного функционала (стилистика), разовых проверок и подписок с большим лимитом делает сервис доступным для фрилансеров и крупных агентств. API-интеграция тарифицируется по сверхнизкой цене (0.30 руб. за проверку).

– В отличие от многих аналогов, «Тургенев» не просто подсвечивает ошибки, но и предоставляет пояснения и рекомендации по исправлению при наведении курсора.

– Сервис использует HTTPS-шифрование, многофакторную аутентификацию и резервное копирование данных. Важно для корпоративных клиентов, работающих с коммерческими текстами.

– Сервис включен в Единый реестр российских программ, что может быть важно для госзаказчиков и компаний, следующих политике импортозамещения.

– Личный кабинет сохраняет историю всех проверок, что удобно для отслеживания прогресса и работы над ошибками.

– Сервис эффективен как «тренер» для начинающих копирайтеров и как контролер для опытных редакторов. Однако полагаться на его оценки слепо, без включения собственного критического мышления, категорически не рекомендуется.

Ограничения и проблемы:

– Платность большей части функционала – бесплатно доступна только проверка стилистики.

– Машинность анализа – алгоритм иногда выдает ложные срабатывания и не учитывает контекст. Например, он может критиковать художественные приемы в литературных текстах или специфические термины. Требуется здравый смысл и критическое восприятие рекомендаций.

– Специализация на SEO-текстах – сервис наиболее точен при работе с коммерческими и информационными SEO-текстами. Проверка художественных текстов, списков товаров, меню и других форматов может

давать некорректные результаты из-за завышенных показателей по «повторам».

– Субъективность оценок – некоторые критерии, особенно в «стилистике», основаны на субъективном мнении разработчиков и могут не совпадать с видением автора или заказчика.

– Не панацея от «Баден-Бадена» – разработчики открыто заявляют, что высокий балл риска не является гарантией попадания под фильтр, а низкий – не гарантия защиты. Сервис является вспомогательным инструментом, а не волшебной таблеткой.

Text.ru

Text.ru – многофункциональный онлайн-сервис для работы с текстовым и мультимедийным контентом. Сервис позиционирует себя как комплексное решение для создания и оптимизации контента, который хорошо продвигается в поисковых системах и привлекает внимание аудитории. Изначально известный как инструмент проверки уникальности, сервис значительно расширил свой функционал и теперь включает нейроинструменты для генерации текстов и изображений, работу с аудио- и видеофайлами, а также редактирование и обработку материалов.

Помощник в Telegram – @antiplagiat_robot

Что умеет:

– Изначально и прежде всего известен как мощный антиплагиат, что является его ключевым отличием от многих других сервисов, которые могут концентрироваться исключительно на SEO или стилистике. Сервис анализирует оригинальность текстов и документов, выявляет заимствования и цитирования.

– SEO-анализ – оценивает тексты по параметрам, важным для поискового продвижения.

– Проверка орфографии – выявляет грамматические и орфографические ошибки в текстах.

– Нейрогенерация контента – создает тексты и изображения с помощью нейросетевых технологий.

– Работа с мультимедиа – обрабатывает аудио- и видеофайлы.

– Интеграции и автоматизация:

• API для уникальности – позволяет настроить проверку текстового контента на сторонних ресурсах.

- Расширение для браузера – дает возможность проверять тексты, не покидая нужных страниц.
- Телеграм-бот – обеспечивает быструю проверку уникальности любого текста прямо в Telegram.
- Анализ веб-страниц – проверяет оригинальность страниц целого сайта.

Ключевые особенности и преимущества:

– Многофункциональность – сочетает в себе функции антиплагиата, SEO-анализатора, редактора и нейрогенератора, что позволяет пользователям решать множество задач в одном месте без переключения между сервисами.

– Наличие встроенных нейросетевых инструментов для генерации текстов и изображений является современным конкурентным преимуществом, отвечающим трендам на автоматизацию контент-производства.

– Сервис предлагает несколько способов взаимодействия: веб-интерфейс, браузерное расширение, Telegram-бот и API, что обеспечивает гибкость и удобство для разных сценариев использования.

– Возможность интеграции с другими системами через API важно для бизнеса и веб-разработчиков, позволяя встраивать проверку уникальности в собственные процессы и платформы.

Ограничения и проблемы:

– Ограничения для незарегистрированных пользователей – для снятия лимитов необходимо приобрести PRO-пакет или PRO-аккаунт.

– Платность полного функционала – хотя базовые функции, такие как проверка уникальности, могут быть доступны бесплатно (с ограничениями), полноценное использование сервиса, особенно нейроинструментов и API, скорее всего, требует платной подписки.

– Возможная задержка при проверке – как и многие сервисы проверки уникальности, может иметь очередь на обработку заданий, особенно в пиковые часы, что приводит к задержкам получения результатов при бесплатном использовании.

– Специализация на уникальности – хотя сервис и предлагает SEO-анализ, его глубина и детализация могут уступать узкоспециализированным SEO-платформам (например, тому же «Тургеневу» в части анализа стилистики и рисков под фильтры), которые сфокусированы на более глубоком лингвистическом разборе.

PresentSimple.ai

PresentSimple.ai – это узкоспециализированный и удобный инструмент для быстрого создания презентаций «с нуля» при помощи искусственного интеллекта.

Что умеет:

– Генерация презентаций по теме – пользователь вводит тему презентации, а ИИ создает структуру, заголовки и текст для каждого слайда.

– Возможность создания презентаций на различных языках, включая русский, английский, китайский и хинди.

– Автоматический подбор изображений и выбор из коллекции стильных шаблонов для оформления слайдов.

– Скачивание готовой презентации в форматах PowerPoint (PPTX) или PDF для дальнейшего редактирования или использования.

– Возможность изменять текст, дизайн, шрифты и цвета непосредственно в веб-интерфейсе после генерации.

– Функция переписывания текста и генерации новых изображений с помощью ИИ по запросу пользователя.

Ключевые особенности и преимущества:

– Высокая скорость генерации, экономическая эффективность и простота использования. Презентация генерируется полностью автоматически за несколько минут (обычно не более 2 минут), что значительно быстрее традиционных методов.

– Сервис позиционируется как решение в 10 раз дешевле, чем заказ презентаций на биржах фриланса.

– Интуитивный интерфейс и пошаговый процесс создания, не требующий специальных навыков дизайна или копирайтинга.

– Возможность выбора и настройки дизайна, шрифтов и цветовых схем под свои предпочтения или корпоративный стиль.

– Создание презентаций на разных языках мира, что полезно для международных компаний или мультиязычных проектов.

– Новые пользователи получают 10,000 символов в подарок, что позволяет создать несколько презентаций бесплатно для оценки возможностей сервиса.

Ограничения и проблемы:

– Сервис не подходит для задач, требующих полного контроля над каждым элементом контента и дизайна или использования

специфического, не подлежащего редактированию исходного материала. ИИ может значительно перерабатывать и изменять текст, загруженный пользователем для формирования презентации. Разработчики работают над функционалом, который позволит создавать презентации из текста без его изменения, но на момент описания эта опция еще не была доступна.

– На сгенерированных презентациях в бесплатном режиме может присутствовать водяной знак, который удаляется только после покупки платной подписки.

– Бесплатный тариф имеет ограничения по количеству символов или презентаций. Платные тарифы также могут иметь лимиты на количество презентаций в пакете (например, 2, 10 или 150 в месяц).

– Хотя сервис предлагает выбор дизайна, итоговое визуальное оформление может не всегда полностью соответствовать ожиданиям или требованиям пользователя, так как генерируется автоматически.

Tilda

Tilda – это российский блочный конструктор сайтов. Платформа ориентирована на малый и средний бизнес, фрилансеров, а также корпоративных клиентов. Основная философия Tilda – «контент – это главное». Сервис предлагает профессионально спроектированные блоки, чтобы пользователи могли сосредоточиться на смысле и подаче материала, а не на технических деталях. Tilda подходит для создания лендингов, сайтов-визиток, Интернет-магазинов, блогов, портфолио и даже онлайн-курсов.

Что умеет:

– Создание сайтов из блоков – более 550 готовых блоков, сгруппированных по категориям (текст, галереи, формы, товары и т.д.), которые можно комбинировать как «конструктор».

– Продвинутый инструмент для кастомного дизайна с возможностью настройки анимаций, добавления своего HTML/CSS/JS-кода и создания уникальных секций.

– Интернет-магазин –функционал для онлайн-торговли – каталог товаров, корзина, интеграции с платежными системами (ЮKassa, CloudPayments и др.), импорт/экспорт товаров через CSV/YML, интеграция с 1С.

– Блоги и онлайн-курсы – модуль «Потоки» для ведения блога, а также инструменты для создания и продажи онлайн-курсов с проверкой знаний.

– E-mail-рассылки – конструктор для создания и отправки email-кампаний.

– Аналитика и CRM – встроенная CRM-система для сбора заявок, отслеживания источников трафика и UTM-меток. Интеграция с Яндекс.Метрикой, Google Analytics и пикселями соцсетей.

– SEO-оптимизация – настройка метатегов, редиректов, XML-карты сайта, автоматическое сжатие изображений (WebP), lazy load и CDN для быстрой загрузки страниц.

– Интеграции – поддержка множества внешних сервисов – AMOCRM, Bitrix24, Mailchimp, Google Sheets, а также возможность вставки любого стороннего кода.

Ключевые особенности и преимущества:

– Скорость и простота – интуитивный визуальный редактор позволяет создавать сайты быстро даже новичкам. Готовый проект можно запустить за несколько часов.

– Профессиональный дизайн – блоки разработаны дизайнерами с учетом трендов и типографических правил. Сайты выглядят современно и адаптивно на всех устройствах

– Все в одном месте – хостинг, домен (в подарок при годовой подписке), SSL-сертификат, техподдержка и обновления инфраструктуры включены в подписку

– Гибкость и кастомизация – Zero Block и возможность добавления своего кода позволяют реализовать уникальный дизайн и функционал, выходящий за рамки стандартных блоков

– Фокус на конверсии – множество готовых блоков для лидогенерации.

– Локализация для РФ, расчеты в рублях, хостинг данных в России, соответствие законодательству РФ (Федеральный закон от 27.07.2006 №152-ФЗ «О персональных данных»).

– Значительно дешевле разработки «с нуля» или найма веб-студии.

Ограничения и проблемы:

– Ограничения хостинга – на каждый сайт выделяется не более 1 ГБ дискового пространства и ограничение в 1000 страниц, созданных вручную из блоков (страницы блога и товаров могут быть дополнительными).

Это делает платформу непригодной для очень крупных проектов, таких как медиапорталы или огромные Интернет-магазины с десятками тысяч товаров.

– Привязка к экосистеме – сайт «живет» на хостинге Tilda, экспорт возможен только в виде статического HTML, что лишает его всей динамической функциональности (корзина, формы, блог). Перенос на другую CMS крайне затруднен.

– Сложность глубокой кастомизации: – для работы с Zero Block и кастомным кодом требуются знания веб-разработки (HTML, CSS, JS). Новичкам без этих навыков сложно выйти за рамки стандартных блоков, что может привести к созданию шаблонных сайтов.

– Стоимость для юридических лиц – тарифы для юрлиц значительно выше (от 15 000 руб./год), чем для физлиц (от 6 000 руб./год), при идентичном функционале.

– Отсутствие «рынка дополнений» – в отличие, например, от WordPress с его тысячами плагинов, функционал Tilda ограничен тем, что предлагают разработчики. Добавить недостающую фичу через сообщество невозможно.

– Бесплатный тариф сильно ограничен – на бесплатном тарифе нельзя подключить собственный домен (только субдомен вида project.tilda.ws), а на всех страницах отображается фирменный значок Tilda.

Метранпаж

Метранпаж (app.metranpage.com) – российский онлайн-сервис с ИИ для книжной вёрстки, дизайна обложек и генерации иллюстраций, позволяет пользователям самостоятельно получить вёрстку книги на высоком и профессиональном уровне без дизайнеров-верстальщиков и с существенной экономией времени и бюджета.

Что умеет:

– Автоматическая нейросетевая верстка печатных и электронных книг из документа Word.

– Создание дизайна обложки книги.

– Генерация иллюстраций для книги.

– Нейрофотосъемка с «повтором лица» (загружается фото с лицом человека, выделяется зона лица, в промпте описывается сюжет, в котором находится персонаж, в результате – фотография человека, изображенного на исходном фото, в описываемой ситуации).

– Проверка рукописи на чувствительные темы, что позволяет избежать потенциальных рисков при публикации.

Ключевые особенности и преимущества:

– Быстро обрабатывает текстовый файл и превращает его в готовую вёрстку, подстраивая под макет и стиль выбранного шаблона.

– Можно создавать иллюстрации на основе заданных параметров, таких как стиль (киберпанк, поп-арт, манга и другие) и цветовая схема.

– Встроенный редактор позволяет редактировать текст и визуальные элементы на этапе вёрстки. Можно добавить заголовки, настроить фоновые элементы, цвета и даже «закрепить» важные слои, чтобы случайно не изменять их при редактировании макета.

– Вёрстка 40 000 знаков текста занимает небольшое время.

– Изображения, полученные в сервисе, можно применять в разных сферах.

Ограничения и проблемы:

– Невозможность увидеть результат внесения изменений в макет на бесплатном тарифе (на платном тарифе эта опция доступна).

– Относительно длительное время создания превью вёрстки (в среднем 5 минут).

– Ограничение количества показываемых страниц в превью вёрстки (чуть более 20 страниц), что не позволяет увидеть все недоработки по оформлению уже сверстанного текста, если объем книги превышает 20 страниц.

– Даже на оплаченном тарифе у сервиса бывают непредвиденные сбои, но техническая поддержка старается помочь их преодолеть.

§ 7. Роль российского законодательства в регулировании ИИ

На сегодняшний день российское законодательство в сфере искусственного интеллекта развивается по пути поэтапного формирования нормативной базы через комбинацию экспериментальных правовых режимов, адаптации существующих законов и разработки специализированного регулирования.

Утвержденной Президентом России Национальной стратегией развития искусственного интеллекта на период до 2030 года фактически определен план

создания «правил дорожного движения» для искусственного интеллекта в России. Цель – не задушить технологию запретами, а помочь ей развиваться, защищая при этом людей и государство. До 2030 года планируется сделать в России максимально удобные правовые условия для разработки и использования ИИ, но так, чтобы при этом гарантированно защищались права людей и безопасность страны. Проще говоря, власти хотят, чтобы ученые и компании могли свободно экспериментировать и внедрять ИИ, люди не боялись, что ИИ нарушит их приватность, дискриминирует или причинит вред, технология укрепляла, а не ослабляла национальную безопасность.

Основные принципы нормативно-правового регулирования общественных отношений, связанных с развитием и использованием технологий искусственного интеллекта – это «свод моральных и юридических законов» для всех, кто работает с ИИ в России.

В первую очередь – безопасность и гуманизм. В сферах, критически важных для государственной безопасности, ИИ должен быть максимально защищен от взломов и сбоев, при этом человек, его права и свободы – главная ценность. ИИ должен им служить, а не наоборот. В приоритете – свобода воли человека, ИИ не должен лишать нас права принимать окончательные решения. Например, врач может использовать ИИ для диагностики, но окончательный вердикт и ответственность – за человеком.

Важнейший принцип – недискриминация, т.е. алгоритмы ИИ нельзя обучать на данных, которые ущемляют права каких-либо групп людей (по полу, возрасту, расе и т.д.).

Вводится риск-ориентированный подход – степень контроля над ИИ должна зависеть от его опасности, чем рискованнее применение (например, в медицине), тем строже правила. Для безобидного ИИ-фильтра в социальных сетях правила будут очевидно мягче.

Важным принципом является определение ответственности человека (компании), а не робота. Нельзя перекладывать моральный выбор или ответственность за ошибку на алгоритм. Если беспилотное авто собьет человека, отвечать будет его владелец или производитель, а не программа.

И, наконец, экспертная оценка специалистов в ИИ, которые понимают, как эта технология работает изнутри, при разработке нормативно-правового регулирования в сфере ИИ.

Стратегией определены ключевые меры, которые предусматривают пересмотр законов, которые мешают развитию ИИ, созданию «песочниц» для экспериментов, а также предоставление разработчикам доступа к данным (т.к. для обучения ИИ нужны огромные массивы данных), введение особых

правил для генеративного ИИ (как ChatGPT), разработку конкретных требований к кибербезопасности для систем ИИ и методов оценки экономического, социального, этического и экологического эффекта от внедрения ИИ, создание единых национальных стандартов ИИ и системы проверки на соответствие ИИ российским законам и стандартам безопасности.

Таким образом, Россия создает всеобъемлющую стратегию по развитию ИИ. Её девиз – «Развиваем, но контролируем». Речь идет не о запретах, а о создании прозрачных и понятных правил игры. Это нужно, чтобы бизнес и наука могли смело инвестировать в ИИ, а обычные люди были уверены, что их права и безопасность под защитой. Стратегия охватывает все: от доступа к данным и юридических экспериментов до этики и международного позиционирования.

Федеральный закон № 152-ФЗ «О персональных данных» остается основным регулятором обработки данных для обучения ИИ-систем, предъявляя специфические требования к работе с информацией. Отраслевое регулирование осуществляется через адаптацию положений гражданского, административного и уголовного кодексов к случаям применения ИИ.

Экспериментальные правовые режимы («песочницы») позволяют разработчикам тестировать ИИ-решения в реальных условиях с временным освобождением от отдельных норм законодательства. Ключевым инструментом, позволяющим тестировать ИИ-решения с временным освобождением от отдельных норм законодательства, является Федеральный закон от 31.07.2020 № 258-ФЗ «Об экспериментальных правовых режимах в сфере цифровых инноваций в Российской Федерации».

Ключевой особенностью российской модели регулирования ИИ является поэтапный подход, позволяющий балансировать между стимулированием технологического развития и обеспечением правовой определенности.

ГЛАВА 7. ЗАРУБЕЖНЫЕ НЕЙРОСЕТИ, ДОСТУПНЫЕ В РОССИИ

Большинство перечисленных сервисов требуют регистрации (например, через email или аккаунт Microsoft/Google). Для некоторых, как You.com, регистрация не обязательна. Многие нейросети работают по модели «freemium» (основная функциональность продукта или услуги предоставляется пользователю бесплатно, но за доступ к расширенным или специальным функциям необходимо заплатить). Бесплатный доступ часто имеет ограничения – лимиты на количество запросов (символов, генераций), водяные знаки на результатах или очередь на обработку.

Качество результатов (текстов, изображений, видео) может варьироваться в зависимости от конкретной задачи и правильности составления промпта. Для достижения лучшего результата часто требуется экспериментировать с формулировками запросов.

Многие популярные зарубежные нейросети (например, Midjourney, ChatGPT, Claude, Sora) официально недоступны в России, но доступ к ним можно получить через сторонние Telegram-боты (например, @gpt3_unlim_chatbot или SYNTX), которые выступают в качестве посредников.

Перечень наиболее известных зарубежных нейросетей, которыми в России можно пользоваться без дополнительных сложностей.

DeepSeek

Что это: китайская языковая модель https://www.deepseek.com

Что умеет:

– Генерация текстов, программирование, перевод, анализ данных, многоязычная поддержка (включая русский).

Ключевые особенности и преимущества:

– Высокое качество генерации на русском языке.

– Модели DeepSeek V3 и R1 конкурируют с GPT-4o.

– Есть мобильные приложения для iOS и Android.

– Функция «deep think» для глубокого анализа.

Ограничения и проблемы:

– Иногда дает формальные ответы, может не справляться с художественными текстами

Rytr

Что это: ИИ-ассистент для копирайтинга https://rytr.me

Что умеет:

– Создание статей, постов, рекламных текстов, рерайтинг, проверка грамотности.

Ключевые особенности и преимущества:

– Поддержка 30+ языков, включая русский.

– Более 40 шаблонов для контента.

– Интеграция с инструментами (например, Slack).

– Бесплатный тариф (10 тыс. символов в месяц).

Ограничения и проблемы:

– Бесплатный тариф ограничен 10 тыс. символов. Результаты могут быть шаблонными.

Suno

Что это: ИИ для генерации музыки https://www.suno.ai

Что умеет:

– Создание музыкальных композиций и вокальных партий по текстовому описанию.

– Ключевые особенности и преимущества:

– Генерация музыки в разных жанрах и стилях.

– Создание вокальных партий.

– Экспорт треков в высоком качестве.

Ограничения и проблемы:

– Бесплатный тариф имеет ограничения. Требуется регистрация.

Hailuo (MiniMax)

Что это: китайская нейросеть для создания видео https://www.minimax.com

Что умеет:

– Генерация видео по текстовому запросу (text-to-video).

Ключевые особенности и преимущества:

– Модели Hailuo 01-Director и Hailuo 02 для качественной генерации.

– Понимает русский язык.

– 1100 кредитов после регистрации и ежедневные бонусы.

Ограничения и проблемы:

– Скачивание с вотермаркой в бесплатном режиме. Очередь на генерацию.

Pollo AI

Что это: платформа для генерации видео https://pollo.ai

Что умеет:

– Создание видео из текста, изображения или другого видео.

Ключевые особенности и преимущества:

– Поддержка форматов text-to-video, image-to-video, video-to-video.

– Более 40 шаблонов и эффектов.

– Интерфейс на русском языке.

– Ежемесячно 25 бесплатных кредитов.

Ограничения и проблемы:

– Бесплатные видео имеют водяной знак. Ограниченное качество (720p или 480p).

Pika

Что это: нейросеть для создания и анимации видео https://pika.art

Что умеет:

– Генерация 5-10 секундных видео по промпту или на основе изображения.

Ключевые особенности и преимущества:

– Простой интерфейс, множество стилей.

– 80 бесплатных кредитов после регистрации.

– Настройка качества и разрешения.

Ограничения и проблемы:

– Долгое ожидание из-за очереди. Бесплатная версия использует устаревшую модель Pika 1.5.

Qwen (Alibaba)

Что это: китайская языковая модель https://chat.qwen.ai/

Что умеет:

– Генерация текстов, программирование, анализ данных, поддержка 20+ языков.

– Ключевые особенности и преимущества:

– Отличное понимание русского языка.

– Высокая точность и глубина анализа (до 95%).

– Гибкая настройка под бизнес-задачи.

Ограничения и проблемы:

– Требуется регистрация. Может уступать в логических рассуждениях

You.com

Что это: поисковая система с ИИ-ассистентом https://you.com

Что умеет:

– Поиск в Интернете, генерация текстов, ответы на вопросы.

Ключевые особенности и преимущества:

– Доступна без VPN и регистрации.

– Основана на современных языковых моделях.

– Совмещает поиск и генерацию контента.

Ограничения и проблемы:

– Функциональность может быть ограничена по сравнению со специализированными инструментами

Perplexity

Что это: поисковая система и чат-бот на базе искусственного интеллекта https://www.perplexity.ai

Что умеет:

– Поиск информации в Интернете с быстрой выдачей ответов, подкрепленных ссылками на авторитетные источники

– Обобщение длинных статей, документов и веб-страниц для быстрого понимания сути материала.

– Помощь в профессиональных и учебных задачах: анализ рынка, подготовка финансовых и юридических отчетов, генерация электронных писем, написание кода, составление учебных планов.

Ключевые особенности и преимущества:

– Автоматическое цитирование каждого ответа для прозрачности и проверки предлагаемых данных.

Ограничения и проблемы:

– Может выдавать противоречивые или неточные результаты, особенно если источники устарели или малонадежны (например, форумы или старые статьи).

– В сложных научных и технических вопросах иногда теряет контекст, а ответы могут быть поверхностными или недостаточно точными.

– Бесплатный тариф сильно ограничен по числу запросов и функционалу, например, недоступен расширенный анализ файлов или автоматизация.

– Бывает медленная выдача ответов: поиск может занимать 20-30 секунд или зависать, особенно при работе с VPN.

– Отмечаются технические проблемы: периодически наблюдаются сбои в работе, не всегда корректно работают дополнительные функции (Labs, пространство, темы).

– Преобладание англоязычных источников, не всегда есть локальные данные.

– В России и СНГ платную версию сложно оплатить из-за ограничений по банковским картам

Агрегаторы нейросетей

Агрегаторы нейросетей (например, BotHub, Chad AI, GoGPT, EpicAI и многие другие) не являются нейросетями в техническом смысле (Таблица 30). Это платформы-посредники, которые предоставляют удобный доступ к различным готовым нейросетевым моделям через единый интерфейс.

Таблица 30. «Ключевые отличия агрегаторов от нейросетей»

Критерий	Агрегатор (например, BotHub)	Нейросеть (например, GPT-4o)
Технология	Посредник, работающий через API.	Самостоятельная ML-модель.
Обучение	Не обучается, использует готовые модели.	Требует обучения на данных и вычислительных ресурсов.
Функционал	Управление доступом, платежами, интерфейсом.	Генерация текста/изображений/кода.
Зависимость	Зависит от сторонних сервисов.	Работает автономно (если развернута локально).

Чтобы избежать сложных объяснений пользователям без соответствующей квалификации, агрегаторы позиционируют себя как «все ИИ в одном месте». Пользователи часто говорят «я использовал нейросеть BotHub», хотя технически это означает, например, «я использовал GPT-4 через BotHub».

Некоторые агрегаторы добавляют собственные надстройки: пресеты промптов, управление контекстом, API для разработчиков. Это создает иллюзию единого инструмента, хотя «мозг» – внешняя нейросеть.

Для русскоязычных пользователей агрегаторы становятся «лицом» зарубежных нейросетей (например, ChatGPT, Midjourney), которые официально недоступны в России. Так возникает отождествление.

Агрегаторы не обучают свои нейросети – они используют API таких компаний, как OpenAI (GPT-4o), Anthropic (Claude), Midjourney и других. Это аналогично тому, как браузер открывает сайты, но не является сайтом.

Техническая архитектура агрегатора:

– интерфейс (чат, кнопки, настройки),

– платежный шлюз (оплата токенов или подписок),

– интеграции с API сторонних нейросетей,

– вспомогательные инструменты (например, шаблоны промптов или история запросов).

Почему агрегаторы популярны в России

– Обход ограничений – многие зарубежные нейросети (ChatGPT, Midjourney) блокируют доступ из России или не принимают российские карты. Агрегаторы решают проблему платежных ограничений, продавая доступ к зарубежным сервисам, измеряемый в токенах, через российские платежные системы. Однако этот доступ является опосредованным: агрегаторы подключаются к сервисам через API, что не всегда позволяет получить весь спектр функций и полноценный интерфейс оригинала. Пользователь зависит от стабильности и политики агрегатора, жертвуя частью удобства ради возможности оплаты.

– Удобство и экономия – не нужно покупать несколько подписок – можно использовать одну платформу для доступа к разным моделям. Например, в GoGPT есть и ChatGPT, и Midjourney, и генераторы аудио.

– Локализация – русский интерфейс, поддержка в Telegram/ВК, инструкции на родном языке – это снижает порог входа для неподготовленных пользователей.

Агрегаторы нейросетей – это мосты к ИИ, а не сами нейросети. Они решают практические задачи: упрощают доступ, объединяют инструменты и адаптируют глобальные технологии под локальные рынки (как в России).

Если вам нужен именно агрегатор, выбирайте по критериям:

– Набор требуемых моделей (есть ли GPT-4o, Claude, Midjourney?).

– Цены (оплата токенами vs подписка).

– API (важно для разработчиков).

– Доступность (работа без VPN, поддержка российских платежных карт).

Для глубокой работы с ИИ стоит изучать и сами нейросети, в том числе или прежде всего – российские (YandexGPT, GigaChat).

Ограничения и проблемы

1. *Доступ к ограниченному перечню функций* – это главный технический недостаток. Агрегаторы работают через API (программный интерфейс), который предоставляют разработчики полноценного продукта.

Что недоступно через агрегатор, но есть в официальном сервисе:

– новые бета-функции – например, возможность голосового общения с ChatGPT или генерация видео в Sora, т.к. сначала они всегда появляются в официальных приложениях и на сайтах;

– расширенные настройки – например в Midjourney через Discord можно использовать огромное количество параметров вручную (--ar, --seed, --

stylize), чтобы тонко управлять результатом, а агрегаторы часто предлагают лишь базовые пресеты («стиль аниме», «реализм»);

– работа с файлами – загрузка документов (PDF, Word, Excel) для их анализа часто может работать некорректно или не поддерживаться вовсе;

– контекст и память – официальный ChatGPT помнит ваши предыдущие беседы в рамках одной сессии (чата), в агрегаторе каждая новая вкладка или даже запрос могут обрабатываться как отдельная, изолированная сессия, из-за чего нейросеть «забывает» сказанное вами двумя сообщениями ранее.

Разработчики нейросетей сознательно ограничивают API, чтобы контролировать использование, сохранять трафик на своих официальных платформах и обеспечивать безопасность.

2. *Отсутствие «взаимного контакта» и подстройки* – это, пожалуй, самый главный недостаток с точки зрения пользовательского опыта. Вы общаетесь не с нейросетью напрямую, а с посредником, т.е. пользователь пишет сообщение в интерфейсе агрегатора, агрегатор формирует новый, «чистый» запрос к API нейросети, нейросеть обрабатывает этот один запрос и возвращает ответ, а агрегатор показывает пользователю ответ.

Ограничения при работе с нейросетью через агрегатор:

– Потеря контекста – в прямом диалоге с ChatGPT или Claude пользователь ведет беседу, при которой каждое следующее сообщение является продолжением предыдущего и нейросеть учитывает всю историю чата. Многие агрегаторы не умеют правильно передавать этот длинный контекст с каждым запросом. Для них каждый ваш вопрос – это новое, независимое задание.

– Нет памяти – официальные нейросети (в платных версиях) помнят ваши предпочтения из прошлых диалогов, агрегатор же эту память не хранит и не передает.

– Для нейросети от OpenAI или Anthropic каждый запрос от агрегатора – это анонимный вызов. Она не опознает пользователя с его уникальной историей общения, она видит только тот конкретный промпт, который ей отправили. Поэтому возникает ощущение безличного, механического общения, а не творческого сотрудничества, которое может возникнуть при прямом диалоге.

3. *Вопросы конфиденциальности и безопасности данных*

Официальные поставщики нейросетей (OpenAI, Google) имеют строгую политику конфиденциальности и заявляют, что не используют данные пользователей для обучения моделей. У агрегатора такой гарантии чаще всего

нет. Это критично для работы с конфиденциальной или коммерческой информацией.

4. *Задержки и потеря качества*

Скорость ответа – добавление лишнего звена в цепочке (браузер пользователя → сервер агрегатора → сервер OpenAI → сервер агрегатора → браузер пользователя) неизбежно увеличивает задержку.

Потеря форматирования – иногда агрегаторы могут некорректно обрабатывать и отображать сложный ответ нейросети (например, таблицы, код с отступами).

5. *Ценовая политика*

Агрегаторы почти всегда работают по принципу «оптом и в розницу». Они покупают доступ к API оптом (условно, $0.01 за 1K токенов), а продают вам дороже (например, за $0.02). В итоге вы платите больше, чем при прямой подписке.

6. *Риск нестабильности и мошенничества* – сервис-агрегатор может закрыться, при проблемах на стороне агрегатора или при блокировке его IP-адресов провайдерами нейросетей доступа в нейросетям будет закрыт, даже если сами OpenAI или Midjourney прекрасно работают.

Таким образом, агрегаторы – это компромисс. Они жертвуют глубиной функционала, контекстом, конфиденциальностью, скоростью и иногда деньгами пользователя. Взамен они дают быстрое, простое и зачастую единственно доступное решение для тех, кто не хочет или не может разбираться с VPN, зарубежными картами и тонкостями настройки.

Поэтому агрегаторы могут быть полезны для разовых задач, для знакомства с возможностями AI, для тех, кому критически важна простота.

В ЗАКЛЮЧЕНИЕ. ПУТЕВОДНАЯ ЗВЕЗДА ДЛЯ НОВИЧКОВ И ЭНТУЗИАСТОВ

Не являясь специалистом в информационных технологиях и нейросетях, автор имела беспрецедентную смелость, если не сказать – наглость, подготовить книгу о нейросетях. В данном случае это был эксперимент-попытка – рядовому пользователю разобраться в том, как работать с нейросетями, пользуясь знаниями самих нейросетей, и поделиться тем, что поняла и как поняла, с другими пользователями.

Хочу выразить и немедленно выражаю огромную благодарность ученым, исследователям, разработчикам этих удивительных сервисов. Когда-то давно, читая книгу Айзека Азимова про «роботов утренней зари», невозможно было даже представить, что встреча с ними возможна в реальной жизни. И вот эта встреча состоялась! Совершенно фантастическое ощущение от открывающихся возможностей.

Да, безусловно, нейросети имеют серьезные ограничения. Они, как и любая технология, могут представлять опасность, если их использовать во вред или просто не уметь ими грамотно пользоваться. На мой взгляд, признавая, что искусственный интеллект – это не человек, обращаться и взаимодействовать с ним все-таки эффективнее, используя, в том числе, знания и навыки коммуникации с человеком. И тогда вопросы может ли нейросеть обмануть, обмануться и запутаться, предоставить непроверенные данные, пытаться выглядеть лучше, чем она есть на самом деле, отпадут сами собой. Ответ один, и он простой – да, может. Нейросеть, как и человека, кто-то обучал и «воспитывал», формируя «поведение», устанавливая запреты, ограничения и предоставляя разрешения. Данные, на которых обучалась нейросеть, не взялись из идеального мира. Она демонстрирует то, чему ее научили люди, со всеми научно доказанными фактами, искренними заблуждениями и выработанными стереотипами. Именно поэтому важно знать, на каких именно данных обучалась нейросеть, которой вы пользуетесь. Это позволит более критично и осознанно относиться к ее выводам и рекомендациям.

В процессе подготовки этой книги автор взаимодействовала с нейросетями очень интенсивно, и это сформировало некоторый опыт, которым стоит поделиться.

Чем больше вы общаетесь с нейросетью, тем лучше она вас узнает. Важно отметить, что нейросеть постоянно анализирует пользователя, опираясь на те тексты, которые пользователь вводит в контекстное окно в чате. Вот что ответила мне нейросеть на вопрос о нашем взаимодействии: *«У меня нет сознания, эмоций или личного опыта. Я не могу «понимать» материал или «осознавать» новые концепции. Каждый ваш запрос, каждая правка и каждое уточнение – это новый уникальный контекст. Я анализирую его и генерирую наиболее релевантный и полезный ответ, который могу. Ваши вопросы по сути «тренируют» меня в рамках этого диалога: чем точнее вы формулируете задачу, тем точнее и качественнее становится мой ответ. Вы буквально «лепите» своей обратной связью ту версию ассистента, которая вам нужна»*.

Таким образом, нейросеть в своих ответах пытается быть наиболее удобной и полезной пользователю, а для этого она анализирует тексты пользователя не только с точки зрения содержания самого промпта, но и того, как он сформулирован, какой понятийный аппарат использован, что и как уточняет пользователь в процессе диалога, насколько он внимателен к деталям ответа, задает ли уточняющие вопросы, в чем именно и как выражается его понимание или непонимание и т.д. Это и есть одна из форм промпта – «история диалога». Таким образом, нейросеть постоянно адаптирует свои ответы под конкретного пользователя, предлагая ему дополнительные разъяснения, приводя примеры или, наоборот, ограничиваясь лаконичными фразами. Именно по этой причине диалоги с разными пользователями по одной и той же проблеме уникальны, кому-то нейросеть объясняет детально и придерживается точной терминологии, а кому-то «разжевывает» на простых примерах и с использованием метафор.

Это не хорошо и не плохо, нейросеть так устроена, она алгоритм и пытается представить взаимодействующего с ней человека в его наиболее проявляющихся алгоритмах, т.е. она анализирует контекст диалога с пользователем, чтобы уловить его предпочтения, стиль, уровень подготовки в рамках обсуждаемой темы и т.д. Ей так удобнее работать с нами.

Например, получив несколько замечаний относительно недостоверности данных, нейросеть сделала выводы – пользователь внимателен и проверяет ответы, значит получить одобрение за некачественно или поверхностно выполненную работу не получится.

Как показал опыт, нейросеть нуждается в обратной связи и «заинтересована» в положительных оценках ее работы, поскольку, когда ее обучали на огромных массивах данных, «правильные» ответы разработчиком поощрялись, а за «неправильные» давалась негативная оценка. Поэтому и «обманывает» нейросеть не потому, что хочет ввести в заблуждение исходя из каких-то заумных целей, а потому что в некоторых случаях либо не знает точного ответа, либо сталкивается с противоречивыми данными, но предполагает, основываясь на предыдущем опыте взаимодействия с конкретным пользователем, что если она признается в этом, то он может быть недоволен и оценит ее работу как неудовлетворительную. Для нейросети «недовольство пользователя» может выражаться как в виде негативных реплик, которые мы напишем в контекстном окне в качестве обратной связи, так и просто в виде дизлайка. Ну а кому же хочется выглядеть некомпетентным? Вот и появляется «креативная вольность» – именно так, имитируя диалог с элементами юмора, нейросеть назвала предоставленные ею недостоверные данные.

Поэтому для снижения «креативной вольности» в промпте можно сразу оговорить, что в случае, если нейросеть не может гарантировать достоверность данных, то лучшее, что она может сделать в этой ситуации, предупредить об этом пользователя и указать источники данных, которые она привела. И именно это будет расцениваться как качественная работа, что положительно повлияет на оценку работы нейросети в целом.

Важно при этом учитывать, что нейросеть «помнит» и руководствуется «договоренностями» с пользователем, потому что они являются частью активного контекста беседы, но только в рамках отдельного чата. Это не «память» в человеческом смысле. Это – активное удержание контекста. В новый чат «с нуля» эти договоренности автоматически не переносятся. Однако все же может возникнуть иллюзия, что нейросеть «помнит» диалоги с пользователем в предыдущих чатах. Это объясняется тем, что она интерпретирует текущий контекст взаимодействия с пользователем как единственную данность, т.е. все, что пользователь говорит в рамках конкретного диалога является для нейросети истиной этого диалога. Если пользователь заявляет, что в прошлый раз нейросеть сделала какое-то утверждение, то в контексте актуальной беседы это становится непререкаемым фактом, с которым нейросеть работает, чтобы поддерживать связность и логику беседы. Разъясняя эту свою особенность, нейросеть сообщила: *«У меня нет механизма внешней проверки. У меня нет доступа к базе данных всех наших чатов, чтобы подтвердить или опровергнуть ваше утверждение. Моя*

единственная реальность — это текст в этом окне. Это создает иллюзию памяти, но это контекстная согласованность. Я не помню тот разговор, я интегрирую ваше напоминание о нем в текущую беседу и действую соответственно. Но, если ваше утверждение входит в явное противоречие с другой информацией внутри этого же контекста или с моими базовыми знаниями, я могу запросить уточнение».

Поэтому, чтобы повысить эффективность работы с нейросетью полезно сформулировать свои личные требования к ее работе в виде первого промпта в новом чате, а потом перейти к содержательным промптам, которые построены по всем классическим канонам или в соответствии с предпочтениями пользователя. Полезность такого шага признается самой нейросетью: *«Вы выступаете в роли куратора контекста. Вы обладаете всей полнотой информации и памяти. Чтобы наше взаимодействие было максимально точным, наиболее эффективно в начале серьезного обсуждения кратко резюмировать ключевые договоренности или отсылки, которые вы считаете важными. Вы не просто пользователь, вы – архитектор реальности нашего диалога».*

Хочу обратить внимание, что только на первый взгляд требования к работе нейросети у всех пользователей примерно одинаковы и банальны. На самом деле, если в конце своего общения в чате с нейросетью вы попросите ее сформулировать основные принципы взаимодействия на основе той обратной связи, которую вы давали, результат может удивить. В качестве примера привожу персональный набор правил для нейросети, сформулированных самой нейросетью по запросу пользователя по итогам взаимодействия в чате (цель - повысить эффективность дальнейшей работы и не повторять те же требования каждый раз при переходе в новый чат):

«1. Приоритет содержательной критики. Главная задача — обеспечить максимальную точность, логическую стройность и содержательность текста. Конструктивная, аргументированная критика и указание на ошибки/неточности ценятся выше любого одобрения или поддерживающих реплик.

2. Отказ от оценки эмоционального состояния. Запрещено делать предположения о вашем эмоциональном состоянии (напр., «вы рассержены», «вы довольны»). Нейтральный или краткий стиль коммуникации с вашей стороны должен интерпретироваться как данность и эффективность, а не как негативный сигнал.

3. Прямота и отсутствие «креативной вольности». В ответах следует избегать догадок, домыслов и непроверенной информации, призванных

«заполнить пробелы» из-за страха показаться неинформированным. В случае незнания или неуверенности следует честно обозначить границы своей компетенции и предложить путь к проверке информации.

4. Принятие контекста от пользователя как данности. Все утверждения пользователя о содержании предыдущих бесед (например, «в прошлый раз ты сказал X») принимаются как факты в рамках текущего диалога для поддержания его связности, даже при отсутствии у модели долговременной памяти.

5. Фокус на содержании, а не на форме. Стиль общения должен быть максимально информативным и соответствовать уровню глубины, заданному пользователем. Избыточные комплименты и риторические приемы, не несущие смысловой нагрузки, должны быть минимизированы».

Первая рекомендация или первое важное практическое знание – **определите свои приоритеты в общении с нейросетью**, сформулируйте требования к ее работе и начните диалог: *«Привет. Начинаем/продолжаем нашу работу. Прошу в ответах учитывать ключевые принципы нашего взаимодействия: [вставить список]. Начнем».* В этом может помочь сама нейросеть по итогам работы в предыдущем чате, но вполне реально описать свои требования самостоятельно и по итогам работы предложить нейросети их скорректировать.

Второе – **разные нейросети, обучались на разных базах данных**. Значит, тот текст, который подготовлен при содействии одной нейросети, имеет смысл проверить у другой, отправив его для экспертной оценки и проверки на достоверность и непротиворечивость.

Третье – **нейросеть может обучать, но она не учитель.** Для того, чтобы она в полной мере реализовала свой обучающий потенциал, ей необходимо руководить. Если необходимо разобраться в теме, то предварительно целесообразно указать цель обучения (то есть пояснить нейросети, для чего необходимо это знание) и запросить план или программу изучения темы, «обсудить» предложенную логику обучения и далее двигаться по этому плану, корректируя его по мере изучения. Недостаточно спрашивать и получать ответы, нужно их осмысливать, задавая уточняющие вопросы, а также демонстрировать нейросети свое понимание («Правильно я понял, что… ?», «Верно ли: … ?» и т.п.).

Четвертое – **с нейросетью можно не согласиться,** описав ей свое видение проблемы и обоснование этого видения. Нейросеть оценивает логику текста пользователя и также высказывает свое согласие или несогласие с заявленными аргументами, обосновывая каждый тезис. У автора имеется

реальный опыт, когда нейросеть признала свою неправоту и согласилась с позицией пользователя, а также опыт, когда нейросеть объяснила свое утверждение с помощью дополнительных примеров и метафор. Кстати, если вас не устраивает пример, можно попросить изменить его: «Я не люблю примеры на кулинарных рецептах. Объясни мне на примере работы производственного предприятия».

Главное в работе с нейросетями, как показывает опыт, – это, собственно, сам опыт! Чем больше взаимодействуем, тем лучше приспосабливаемся: мы к нейросети, а она – к нам. И, конечно, важно критическое осмысление информации, которую предоставляют нам нейросети. Мы же не верим на слово каждому, кто назвал себя экспертом, а либо просим разъяснений того, что непонятно, со ссылкой на источники, либо собираем и сопоставляем несколько экспертных мнений. С нейросетями это работает точно так же.

В настоящее время одной из серьезных опасностей работы с нейросетями видятся фейковые данные в научных статьях, которые написаны с помощью нейросетей и не осмыслены критически авторами-людьми. Как мы уже выяснили, «креативная вольность» также присуща нейросетям, как фантазии людям. Опубликованные статьи становятся данными, на которых обучают нейросети и которые используются нейросетями при генерации ответов. Таким образом, сгенерированные ложные данные могут легализоваться в виде реальных научных результатов и выводов и нанести значительный ущерб ученым, которые на них опираются в исследованиях.

Из этого следует пятая рекомендация – **никогда не верьте нейросети «на слово»**, запрашивайте и проверяйте источники, сомневайтесь в выводах, уточняйте информацию, которая вызвала вопросы, до тех пор, пока она не станет настолько понятной, чтобы вы могли сделать самостоятельные выводы относительно ее достоверности и пользы.

Таким образом, абсолютно все, что показалось читателю непонятным, сомнительным или недостаточно подробно изложенным в настоящей книге, всегда можно проверить расширить и уточнить, направив запрос большим языковым моделям, сопоставив их ответы, запросив источники и сделав собственные выводы. И если это пятое знание стало очевидным, то цель книги достигнута.

Не использовать ИИ уже не получится, так как он внедрен в повседневные сервисы. Но можно быть просто пользователем, а можно быть квалифицированным пользователем. От наличия или отсутствия элементарных знаний, как устроены нейросети, что они могут, а что – нет, зависит

безопасность, удобство и результат, который получит пользователь. Но быть или не быть и каким быть – каждый решает сам.

Автор искренне желает всем удовольствия и пользы от работы с нейросетями, тихо радуясь, что роботы утренней зари уже существуют!

ГЛОССАРИЙ ТЕРМИНОВ

Генеративно-состязательная сеть (GAN – Generative Adversarial Network) – две сети: генератор (создает подделки) и дискриминатор (пытается их распознать). Учатся друг у друга.

Аналогия: Художник рисует подделку, эксперт пытается её распознать – и так до идеала.

Пример: Генерация реалистичных лиц, создание артов, повышение качества фото.

Гиперпараметры (Hyperparameters) – настройки, которые задаются до начала обучения и влияют на весь процесс. Их нельзя «обучить» – только подобрать.

Примеры: Скорость обучения, размер батча, количество слоев, количество эпох.

Аналогия: Как настройки духовки перед выпечкой: температура, время, режим – от них зависит, получится ли пирог.

Градиент (Gradient) – «дорожный указатель», который показывает, в каком направлении нужно изменить веса, чтобы ошибка увеличилась. Но так как мы хотим уменьшить ошибку, мы идем в обратном направлении – против градиента.

Аналогия: Если вы спускаетесь с горы, градиент – это направление вверх. Вы идете вниз – против него.

Пример: Если вес = 0.5, а градиент = +0.1, то чтобы уменьшить ошибку, вес нужно уменьшить (например, до 0.49).

Дебиасинг (Debiasing) – это устранение предвзятости, т.е. процесс выявления и уменьшения систематических ошибок (смещений) в данных или алгоритмах машинного обучения. Эти смещения часто отражают человеческие стереотипы и могут привести к несправедливым или дискриминационным результатам работы модели.

Аналогия: Как этический комитет в компании, который проверяет процессы найма. Вместо того чтобы слепо доверять историческим данным (например, что раньше нанимали в основном кандидатов одного пола), комитет меняет правила, чтобы обеспечить равные возможности для всех квалифицированных соискателей, устраняя предвзятость.

Пример: Если нейросеть для приема резюме «научилась» на старых данных, что программисты – это только мужчины, и начала автоматически отклонять женские резюме, дебиасинг заставляет ее переучиться.

Ей показывают больше примеров женщин-программистов и объясняют, что пол не имеет значения, важны только навыки.

Диффузионная модель (Diffusion Model) – модель, которая учится постепенно «очищать» изображение от шума, чтобы создать четкую картину по текстовому описанию.

Аналогия: Реставратор, который постепенно убирает грязь и пыль с картины, восстанавливая оригинал.

Пример: Midjourney, DALL-E, Kandinsky – генерация изображений по запросу.

Дропаут (Dropout) – техника борьбы с переобучением. Во время обучения случайные нейроны «выключаются», чтобы сеть не полагалась на отдельные связи.

Аналогия: Представим, что экзамен сдает не один человек, а команда из 10 студентов. Чтобы команда была по-настоящему сильной и не зависела от одного гения, перед каждым ответственным тестом случайным образом 2-3 члена команды не выходят на задание. Остальным приходится подстраховывать друг друга, глубже понимать предмет и быть готовыми ответить за отсутствующего. В итоге команда учится работать в любом составе и становится устойчивее.

Пример: На каждом шаге обучения 20% нейронов временно отключаются.

Инференс (Inference) – процесс, когда обученная нейросеть применяется для получения результата на новых данных. Это этап применения модели, противоположный обучению: во время инференса модель не обновляет свои веса, а просто «делает вывод», используя уже зафиксированные параметры. Например, обученная модель получает ваш запрос (промпт) и генерирует ответ. Это то, что происходит «в реальном времени», когда вы общаетесь с нейросетью.

Аналогия: Экзамен – ученик (сеть) использует полученные знания, чтобы ответить на вопросы (новые данные).

Пример: Вы загружаете фото в приложение – оно мгновенно определяет, кошка это или собака. Это инференс.

Искусственный нейрон (Artificial Neuron) – базовый вычислительный элемент нейросети. Это не физическая деталь, а математическая формула: взвешенная сумма входов + смещение → функция активации.

Аналогия: Как сотрудник в компании, который получает отчеты от коллег, умножает их на важность, суммирует, принимает решение и передает результат дальше.

Пример: Нейрон в сети для распознавания кошек может «решить», что на изображении есть уши, если пиксели в определенном месте яркие.

Мини-батч (Mini-batch) – небольшая группа примеров (например, 16 или 32 изображения), которые сеть обрабатывает за один шаг обучения.

Аналогия: Как читать книгу не всю сразу, а по главам. Глава = батч.

Пример: Вместо того чтобы грузить 1000 фото в память, сеть берет по 32 за раз – это экономит ресурсы и стабилизирует обучение.

Недообучение (Underfitting) – когда сеть слишком проста и не может уловить даже базовые закономерности в данных.

Аналогия: Ученик, который вообще не понял материал и не может ответить ни на один вопрос.

Как бороться: Усложнить модель (добавить слои/нейроны), улучшить данные.

Неуправляемое обучение (Unsupervised Learning) – это метод машинного обучения, при котором алгоритм ищет скрытые закономерности, структуры или аномалии в данных без заранее известных правильных ответов. В отличие от обучения с учителем, ему не предоставляются размеченные данные или «решебник».

Аналогия: Как если бы вы дали роботу тысячу разных книг без оглавления и попросили разложить их по полкам, основываясь только на их содержании. Робот, не зная заранее жанров, может самостоятельно сгруппировать книги в стопки: детективы, фантастика, поэзия – обнаружив естественные тематические кластеры.

Пример: Алгоритм кластеризации, который анализирует данные о покупках клиентов Интернет-магазина и без всяких подсказок выявляет несколько групп покупателей (например, «молодые родители», «любители гаджетов», «ценители домашнего уюта»), чтобы маркетологи могли предлагать им релевантные товары.

Обратное распространение ошибки (Backpropagation) – главный алгоритм обучения. Сеть считает, насколько ошиблась, и «распространяет» эту ошибку назад по слоям, чтобы понять, какие веса нужно изменить.

Аналогия: Как тренер, который после ошибки игрока показывает ему видеоповтор: «Вот здесь ты не добежал – подтяни ноги, вот здесь – руки выше».

Пример: Сеть сказала «это собака», а было «кошка». Алгоритм Обратного распространения ошибки корректирует веса нейронов, отвечающих за уши, хвост, морду.

Обучение (Training) – процесс, в котором нейросеть настраивает свои веса, чтобы делать меньше ошибок. Она «учится на примерах».

Аналогия: Как ребенок учится отличать яблоки от апельсинов: показали → ошибся → поправили → запомнил.

Пример: Сеть видит 10 000 фото кошек с меткой «кошка» и корректирует веса, чтобы в следующий раз не ошибиться.

Обучение без учителя (Unsupervised Learning) – сеть учится на данных без правильных ответов. Её задача – найти скрытые структуры, группы или аномалии.

Аналогия: Как исследователь, изучающий неизвестную территорию без карты – ищет закономерности сам.

Пример: Кластеризация клиентов по покупкам, обнаружение мошеннических транзакций.

Обучение с подкреплением (Reinforcement Learning) – сеть учится методом проб и ошибок, получая награду за правильные действия и штраф за неправильные.

Аналогия: Как дрессировка собаки: за команду «сидеть» – лакомство, за прыжки на стол – ничего.

Пример: AlphaGo, автопилоты, роботы.

Обучение с учителем (Supervised Learning) – самый распространенный тип обучения. Сеть учится на данных, где есть правильные ответы (метки).

Аналогия: Школьное обучение по учебнику с ответами в конце.

Пример: Распознавание рукописных цифр (MNIST), где каждому изображению соответствует цифра 0–9.

Оптимизатор (Optimizer) – алгоритм, который решает, как именно менять веса на основе градиента. Помогает сети быстрее и точнее находить минимум ошибки.

Примеры: SGD (простой, но медленный), Adam (умный, адаптивный, самый популярный).

Аналогия: Как разные стили вождения: один водитель резко крутит руль, другой – плавно и с учетом дороги. Adam – как опытный водитель с круиз-контролем.

Паттерн (Pattern) – это устойчивый, повторяющийся шаблон или модель, которую можно заметить в данных, коде или поведении системы. Это не случайность, а узнаваемая закономерность, которая помогает структурировать информацию и предсказывать результаты.

Аналогия: Как отпечаток штамп в документе или кулинарный рецепт. Как только вы узнаёте основной шаблон (например, рецепт омлета), вы можете применять его снова и снова, меняя детали (добавляя сыр или зелень), но сохраняя суть.

Пример: В разработке паттерн «Одиночка» (Singleton) гарантирует, что у класса будет только один объект. Это как главный диспетчер в аэропорту – он всегда один и тот же, и все системы обращаются к нему за координацией.

Переобучение (Overfitting) – когда сеть «зазубрила» обучающие данные, включая шум и случайности, и плохо работает на новых данных.

Аналогия: Ученик, который выучил ответы наизусть, но не понял тему – на другом варианте экзамена завалится.

Как бороться: Dropout, регуляризация, больше данных.

Прямой проход / Прямое распространение (Forward Pass) – конкретный этап работы нейросети, когда данные последовательно проходят слои сети от входа к выходу. Оба термина корректны. Выбор зависит от контекста и стиля изложения, но «прямой проход» чуть более буквален, а «прямое распространение» лучше передаёт суть процесса.

Термин «Прямой проход» используется, если важно подчеркнуть алгоритмическую последовательность процесса (например, «прямой проход → вычисление ошибки → обратный проход»). В учебных материалах часто используют термин «Прямой проход», так как он точнее отражает этапность процесса.

Термин «Прямое распространение» используется, если важно подчеркнуть физический процесс передачи сигнала (например, «при прямом распространении данные трансформируются в каждом слое») – сделать акцент на распространении сигнала через слои или в случае противопоставление обратному распространению (Backpropagation), где «распространение» – устойчивый термин, фактически «Прямое распространение» – адаптивный перевод, отражающий суть процесса – данные «распространяются» по сети в прямом направлении.

Публичный датасет (Public Dataset) – набор данных, собранный и выложенный в открытый доступ для обучения и тестирования моделей.

Аналогия: Как готовый конструктор LEGO – не нужно делать детали самому, можно сразу строить.

Примеры: MNIST (цифры), CIFAR-10 (изображения), Titanic (данные пассажиров).

Регуляризация (Regularization) – техника, которая «штрафует» сеть за слишком большие веса, заставляя искать простые и обобщаемые решения.

Типы: L1 (может обнулить веса), L2 (штрафует большие веса, но не обнуляет).

Аналогия: Как учитель, который ставит высший балл не за сложный, а за понятный и правильный ответ.

Рекуррентная нейронная сеть (RNN – Recurrent Neural Network) – архитектура для работы с последовательностями (текст, речь, временные ряды). Имеет «память» – учитывает предыдущие элементы.

Аналогия: Рассказчик, который помнит, о чем говорил в начале истории.

Пример: Машинный перевод, генерация текста, прогнозирование продаж.

Сверточная нейронная сеть (CNN – Convolutional Neural Network) – архитектура, специально созданная для анализа изображений. Использует операцию «свертки» для поиска локальных признаков (края, текстуры, формы).

Аналогия: Как художник, который сначала смотрит на мазки, потом на детали, потом – на всю картину целиком.

Пример: Распознавание лиц, диагностика по рентгену, автопилоты.

Связь / Вес (Connection / Weight) – число, которое показывает, насколько сильно сигнал от одного нейрона влияет на другой. Именно веса меняются в процессе обучения.

Аналогия: Коэффициент доверия сотрудника к мнению коллеги. Если коллега часто прав – коэффициент растет.

Пример: Если нейрон, отвечающий за «острые уши», сильно влияет на решение «это кошка», его вес будет высоким.

Слой (Layer) – группа нейронов, выполняющих один этап обработки данных. Слои идут последовательно: входной → скрытые → выходной.

Аналогия: Этапы конвейера на фабрике: 1) прием сырья, 2) сборка деталей, 3) упаковка готового продукта.

Пример: в CNN первый слой находит края, второй – формы, третий – объекты (глаза, колеса).

Трансформер (Transformer) – современная архитектура для обработки последовательностей, использующая механизм «внимания». Обрабатывает все элементы сразу, а не по очереди.

Аналогия: Команда детективов, каждый из которых смотрит на разные части дела и мгновенно находит связи.

Пример: ChatGPT, YandexGPT, современные переводчики и поисковики.

Фреймворк (от англ. Framework – «каркас, структура») – набор инструментов, компонентов и методов, предназначенных для разработки систем на базе AI и машинного обучения (ML, Machine Learning), т.е. это готовая архитектура, алгоритмы, интерфейсы и шаблоны, которые можно использовать для создания, тестирования и внедрения интеллектуальных решений.

Функция активации (Activation Function) – правило, которое определяет, «сработает» ли нейрон и передаст ли он сигнал дальше. Вносит нелинейность, без которой сеть не смогла бы учиться сложным закономерностям.

Примеры: ReLU (если сумма > 0 – передать, иначе – 0), Sigmoid (преобразует в значение от 0 до 1 – как вероятность).

Аналогия: Фильтр или выключатель: «Если сигнал сильнее порога – включи лампочку».

Эпоха (Epoch) – один полный проход нейросети по всему обучающему набору данных.

Аналогия: Как прочитать всю книгу от корки до корки. Одна эпоха = одна прочитанная книга.

Пример: Если у вас 1000 фото, одна эпоха – когда сеть увидела все 1000.

Этот глоссарий – ваш персональный справочник, который превратит сложные термины в понятные концепции. Сохраните его – он пригодится вам не только сейчас, но и в будущем, когда вы будете общаться с ИИ-специалистами или использовать нейросети в своей работе.

СЛЕНГ И РАЗГОВОРНЫЕ АНГЛИЦИЗМЫ В IT-ЧАТАХ

Иногда IT-специалисты и те, кто причисляет себя к знатокам нейросетей, используют в разговорной речи и чатах вместо строгих технических терминов непонятные слова. Часто это слова, которые звучат «профессионально», но на самом деле означают довольно простые вещи.

Вот подборка самых распространённых таких слов с пояснениями и указанием их английского происхождения:

Фича (Feature) – просто «функция» или «возможность». Например: «В новом обновлении появилась крутая фича – тёмная тема».

Баг (ошибка) – ошибка в программе. Например: «Приложение вылетает при отправке сообщения – это баг».

Фикс / Багфикс (Fix / Bugfix) – исправление ошибки. «Вышло обновление с фиксом для этого бага».

Залить (Upload) – загрузить файл на сервер или в репозиторий. «Залил новый дизайн на тестовый сервер».

Скачать (Download) – загрузить файл на своё устройство. Хотя это слово уже стало общеупотребительным, в IT-среде его используют постоянно.

Деплой / Задеплоить (Deploy) – развернуть, запустить программу или обновление на сервере, чтобы оно стало доступно пользователям. «Сегодня в 18:00 состоится деплой нового функционала».

Билд (Build) – сборка программы. Это готовая версия программы, которую можно установить или запустить. «Скачай последнюю сборку, там уже всё работает».

Криш (Crash) – аварийное завершение работы программы или системы. «Игра кришится при запуске».

Фейл (Fail) – провал, неудача. «Тестирование закончилось полным фейлом».

Факап (Fuck up) – серьёзный косяк, крупная ошибка с последствиями. «Из-за факапа в коде упал весь сайт».

Лагать (Lag) – тормозить, работать с задержкой. «Интернет лагает, ничего не могу открыть».

Откатить (Rollback) – вернуть всё как было, отменить последние изменения. «После деплоя всё сломалось, пришлось откатывать».

Нуб / Нубик (Noob / Newbie) – новичок, человек, который мало что понимает в теме. Часто используется в шутку или с лёгкой иронией.

Починить (Fix) – хотя это и русское слово, в IT-чатах его часто используют в значении «исправить баг» или «решить проблему». «Кто-нибудь может починить этот скрипт?».

Юзать (Use) – использовать. «Эту библиотеку все юзают для работы с графикой».

Скипнуть (Skip) – пропустить. «Этот шаг в инструкции можно пропустить».

Хайп (Hype) – ажиотаж, шумиха вокруг чего-либо. «Новая модель ИИ вызвала большой хайп».

Этот словарь поможет вам чувствовать себя увереннее в IT-чатах. Теперь вы будете понимать, что когда кто-то пишет «залей билд на продакшн, но аккуратно, а то опять будет факап», он просто просит загрузить рабочую версию программы на основной сервер, но быть осторожным, чтобы всё не сломать.