Вычислительное оборудование для нейронных сетей

Модель скачалась, а программа пишет «недостаточно памяти», хотя на диске свободны сотни гигабайт. Чего именно не хватает?

Файл с моделью скачался на ноутбук, но программа сообщает: «Недостаточно памяти». На диске свободны сотни гигабайт. Почему этого мало? Чтобы ответить, разберём, где хранятся параметры, кто выполняет вычисления и что ещё занимает память во время работы.

Ранее

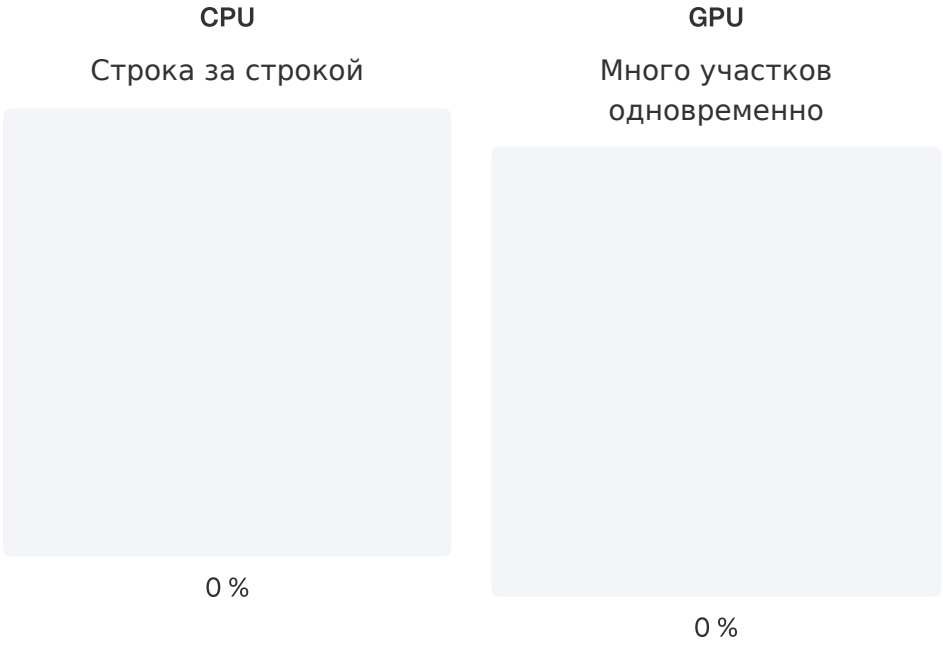
Нейросеть выполняет вычисления с входными числами и обученными параметрами. При обучении параметры меняются; при обычном применении используются готовые.

← Градиентный спуск

1. CPU и GPU

Представим задачу: закрасить изображение Моны Лизы пиксель за пикселем по известным цветам. Один исполнитель идёт слева направо, строка за строкой. Если раздать независимые участки множеству исполнителей, картина проявится быстрее. До просмотра предположите, в каком случае работа закончится раньше и какое условие делает такое разделение возможным.

Последовательное и параллельное заполнение



Одинаковая доля готовности

Запустите CPU и GPU по очереди или сравните порядок заполнения ползунком. Время условное.

Закрашивание Моны Лизы: CPU заполняет строки, GPU — много участков одновременно. Время условное, это не измерение скорости оборудования.

Интерактивный эксперимент — в электронной версии урока.

CPU — универсальный процессор компьютера. Он выполняет команды программ, управляет их работой и подходит для разнообразных задач. У современных CPU несколько ядер, поэтому параллельно они тоже работают. Один исполнитель в анимации нужен для сравнения способов обработки, а не для буквального описания процессора.

GPU, графический процессор, рассчитан на массовое выполнение похожих операций над большими наборами чисел. Нейросеть как раз многократно умножает числа на веса и складывает результаты. Многие такие вычисления можно выполнять одновременно. Поэтому GPU используют и для применения нейросетей, и для обучения, где дополнительно рассчитывают изменения параметров.

Ссылки
Параллельные вычисления на GPU. Ускорение зависит от устройства вычислений и обмена данными с памятью.

Но последовательная зависимость не исчезает. Например, при генерации текста очередной фрагмент зависит от уже полученного текста. Внутри одного шага GPU выполняет много операций параллельно; затем программа переходит к следующему шагу. Нельзя заранее независимо написать каждое слово ответа, как закрасить известный пиксель.

GPU также приходится получать числа из памяти. Если вычислительные блоки ждут данные, добавление ещё большего числа блоков не обязательно ускорит ответ. Поэтому паспортное число операций в секунду не заменяет измерение скорости нужной модели.

> Вычисление содержит много одинаковых операций, которые не зависят друг от друга. Почему для него может подойти GPU?
>
> ○ Он хранит готовый правильный ответ для любой задачи
>
> ○ Он гарантирует ускорение любой программы в одинаковое число раз
>
> ○ Он может распределить независимые операции между многими вычислительными блоками
>
> ○ Он отменяет необходимость хранить параметры модели

2. Параметры и формат хранения

Параметры модели — числа, подобранные при обучении. Их нужно сохранить в файле, а при вычислении сделать доступными процессору. Число параметров и объём памяти — разные величины: память зависит ещё и от того, сколько места отводится каждому числу.

Бит хранит одно двоичное значение; восемь бит составляют байт. В форматах BF16 и FP16 на параметр приходится 16 бит, или два байта. Поэтому у условной модели с 20 миллиардами параметров только веса займут около 40 ГБ. В названиях моделей буква B означает billion, миллиард параметров. Она не обозначает байты.

Сравним три модели в одном формате хранения. Сначала посмотрите на 16 бит: близкие по числу параметров модели занимают примерно одинаковый объём, а сотни миллиардов параметров требуют уже сотен гигабайт. Переключатели позволяют пересчитать те же числа при более компактном хранении.

Число параметров и объём весов

16 бит | 8 бит | 4 бита

GigaChat-20B-A3B	gpt-oss-20b	DeepSeek-V4.1-Flash
20 млрд параметров	21 млрд параметров	552 млрд параметров
40 ГБ	42 ГБ	1 104 ГБ

Одинаковый формат для сравнения. Это расчёт объёма всех весов, а не размер опубликованного файла и не полная память запуска.

Для моделей с экспертами учитывается полное число параметров, даже если на одном шаге используется лишь часть.

Интерактивный эксперимент — в электронной версии урока.

Квантизация позволяет хранить параметры с меньшим числом бит. В грубой оценке восемь бит дают один байт на параметр, четыре бита — половину байта. У той же модели на 20 миллиардов параметров получатся около 20 и 10 ГБ только на веса. Служебные данные формата пока не учитываем; гигабайты считаем десятичными.

Перед переключением в опыте предскажите: во сколько раз уменьшится объём весов при переходе с 16 на 4 бита? Затем проверьте. После этого измените число параметров, сохранив формат.

Объём памяти для весов

Число параметров

Формат хранения

40 ГБ — только веса

20 млрд параметров × 2 байт на параметр.

Десятичные гигабайты. Служебные данные формата, рабочие буферы и кэш не учтены. Меньшее число бит может изменить качество ответов.



Полная полоса — объём тех же весов при 16 битах, а не память устройства.

Расчёт по числу параметров и битам на параметр. Рабочие данные и служебные структуры сюда не входят.

Интерактивный эксперимент — в электронной версии урока.

Меньшее число бит означает более грубое представление чисел. Результат вычислений может измениться, поэтому качество квантизованной модели проверяют на нужных задачах. Само уменьшение файла не доказывает, что ответы останутся прежними или станут быстрее: важна поддержка формата программой и устройством.

Заметка

Размер весов — нижняя оценка потребности в памяти. Если веса занимают все доступные 8 ГБ, для рабочих данных места уже не осталось.

> У условной модели 8 миллиардов параметров. Для каждого отведён один байт. Сколько десятичных гигабайт занимают только веса, без служебных данных?

3. Рабочая память и накопитель

SSD хранит файл модели, в том числе когда компьютер выключен. При запуске процессору нужна доступная для вычислений память. **Оперативная память**, RAM, хранит данные работающих программ; её содержимое обычно теряется при выключении питания. У отдельной видеокарты есть своя **видеопамять**, VRAM. В системах с объединённой памятью процессор и ускоритель пользуются общим пространством, часть которого занята системой и другими программами.

SSD

Файл модели

Хранится и после выключения компьютера.

RAM → CPU

Данные работающей программы

Процессор читает из оперативной памяти и записывает результаты.

VRAM → GPU

Веса и рабочие данные

У отдельной видеокарты — собственная память. У систем с общей памятью CPU и GPU используют один общий объём.

При запуске файл читается с накопителя; выбранная программа размещает веса в памяти, доступной вычислительному устройству.

Свободные 500 ГБ на SSD не превращаются в дополнительные 500 ГБ видеопамяти. Программа может переносить часть работы или весов в оперативную память компьютера, если поддерживает такой режим. Но пересылка данных занимает время; скорость и доступность такого запуска нужно проверять отдельно.

Кроме весов, во время работы сохраняются промежуточные результаты. У языковых моделей может храниться кэш уже обработанного текста: он помогает не повторять часть вычислений. С ростом текста такой кэш обычно увеличивается. Несколько одновременно обслуживаемых запросов тоже требуют дополнительных рабочих данных, даже если используют одни и те же веса.

Допустим, после загрузки весов свободны 2 ГБ. Короткий запрос обрабатывается, а длинный документ вызывает нехватку памяти. Это не противоречие: веса остались прежними, вырос расход на обработку запроса. Число одновременных пользователей и длину текста проверяют вместе с моделью.

Учебный пример: 16 ГБ доступной памяти



Веса · 8 ГБ Рабочие данные · 6 ГБ Свободно · 2 ГБ

Если обработка длинного документа потребует ещё 3 ГБ, свободных 2 ГБ не хватит. Веса при этом не изменились. Числа условные.

Проверьте эту зависимость в калькуляторе памяти ApX: выберите модель около 8B с четырёхбитными весами. Сравните контекст в 8 и 32 тысячи токенов, затем один и четыре параллельных запроса. Меняйте одно условие за раз. Калькулятор даёт оценку; окончательная проверка — запуск в нужной программе.

Во время обучения нужны ещё данные для вычисления градиентов и работы алгоритма обновления параметров. Поэтому возможность запустить готовую модель на устройстве не означает, что на нём получится обучать её тем же способом и с той же нагрузкой.

4. От телефона до сервера

Небольшие модели можно запускать на личных устройствах. Для крупных моделей используют компьютеры с большим объёмом памяти и серверные ускорители; иногда модель распределяют между несколькими устройствами. Слово «сервер» само по себе не указывает ни объём памяти, ни скорость.

Модели на телефонах, компьютерах и ускорителях

Не удалось загрузить пример.

Повторить загрузку

Шесть отдельных примеров. Скорости получены на разных моделях и задачах; напрямую сравнивать быстродействие устройств по ним нельзя.

Интерактивный эксперимент — в электронной версии урока.

У телефона и компактного компьютера память делится с другими программами. Видеокарта получает свой отдельный объём. Обратите внимание на пару RTX 5090 и DGX Spark: во втором примере больше памяти и значительно крупнее модель, но ниже скорость генерации. Это не противоречие и не доказательство преимущества одной архитектуры: здесь различаются сами вычислительные задачи.

Память определяет, что удастся разместить. Скорость показывает, насколько удобно этим пользоваться. Качество ответа проверяется отдельно. Устройство с большим объёмом памяти может вместить более крупную модель и всё же выдавать её ответ медленнее, чем другое устройство выдаёт ответ небольшой модели.

Для языковой модели скорость часто выражают в токенах в секунду. Токен — фрагмент текста, с которым работает модель; подробно разберём его в разделе о языке. Сравнивать два числа скорости можно только с оглядкой на условия: модель, формат весов, длину входа, программу и число одновременных запросов. Отдельно бывает заметна задержка до начала ответа.

Например, один замер сделан на коротком вопросе и небольшой модели, другой — на большом документе и гораздо более крупной. Из этих двух результатов нельзя вывести, во сколько раз один процессор быстрее другого. Для сравнения устройств нужен одинаковый эксперимент.

5. Локальный запуск, аренда и API

Следующий вопрос — чьё оборудование выполняет вычисления. Модель можно запустить на личном компьютере, собственном сервере организации или арендованной удалённой машине. Собственный сервер тоже может стоять далеко от пользователя: обращение по сети ещё не делает его чужим облачным сервисом.

Три способа запуска

Не удалось загрузить пример.

Повторить загрузку

Место размещения и владелец оборудования — разные признаки. Своя машина тоже может находиться удалённо.

Интерактивный эксперимент — в электронной версии урока.

При собственном запуске нужно подобрать совместимую программу, получить веса и следить за доступными ресурсами. Покупка компьютера не отменяет расходы на электричество и обслуживание. Локальная работа может позволить обрабатывать запросы без отправки текста внешнему сервису, но это зависит от устройства приложения: оно может обращаться к внешним инструментам или отправлять диагностические данные.

При **аренде вычислений** пользователь получает доступ к оборудованию и сам организует запуск модели. Плата часто зависит от времени выделения ресурсов, включая простой. В учебном примере машина стоит 2 условные единицы в час. Её арендовали на 10 часов, хотя запросы шли только два часа. При оплате всего времени получится 20 единиц, а не 4.

API — интерфейс взаимодействия программ: одна программа отправляет запрос, другая возвращает результат. Такой интерфейс может быть и у модели на вашем компьютере. Здесь рассматриваем доступ через API к готовой модели провайдера: он запускает и обслуживает её на своём оборудовании. Пользователь обычно оплачивает объём обработки, например входные и выходные токены. Конкретные условия зависят от сервиса. Покупать видеокарту для каждого клиента API не требуется: вычисления выполняются на стороне сервиса.

Для выбора способа запуска полезно сначала проверить небольшую типичную задачу. Помещается ли модель вместе с рабочими данными? Сколько длится ответ? Подходит ли он по качеству? Только после этого имеет смысл сравнивать расходы при предполагаемой нагрузке. Сам факт, что модель написала связный текст, не подтверждает его достоверность.

> Команда арендует сервер с GPU, получает доступ к системе и сама устанавливает программу и веса. Какой вариант описан?
>
> ○ Готовый API, в котором установкой модели занимается провайдер
>
> ○ Покупка оборудования в собственность
>
> ○ Аренда вычислительного оборудования с самостоятельным запуском модели
>
> ○ Запуск целиком на телефоне пользователя

6. Аренда и покупка оборудования

При аренде важна не только ставка, но и состав предложения. На одной площадке указана цена одной GPU, на другой — виртуальной машины с GPU, процессорами и RAM. Эти цены нельзя сравнить, не разобравшись, что входит в каждую.

Измените число часов. Затем переключитесь с Vast.ai на Cloud.ru: вместе с валютой изменится и то, за что платит пользователь. Десять часов аренды оплачиваются целиком, даже если программа считала только два часа.

Стоимость времени на GPU

Не удалось загрузить пример.

Повторить загрузку

Датированный учебный срез, не текущие предложения. Ставки, доступность и дополнительные расходы перед арендой проверяют заново.

Интерактивный эксперимент — в электронной версии урока.

Покупка сервера — другой масштаб расходов. На фотографии система для восьми ускорителей. Каждый имеет свою память; чтобы использовать их вместе, программе нужны средства распределённого вычисления.

Сервер с восемью ускорителями

Не удалось загрузить пример.

Повторить загрузку

Цена системы и сумма памяти не описывают ни скорость конкретной модели, ни расходы за весь срок эксплуатации.

Интерактивный эксперимент — в электронной версии урока.

Сервер, занятый круглые сутки, и сервер, простаивающий между занятиями, дают разную стоимость одного полезного часа. Поэтому деление цены покупки на самую низкую ставку аренды ещё не показывает окупаемость. Сначала измеряют реальную нагрузку и учитывают обслуживание.

7. Память и цена устройств

У видеокарт есть разные классы: игровые, профессиональные и серверные. Ниже длина полосы показывает объём видеопамяти в одном масштабе. Сравните, насколько меняются память и цена при переходе от игровой карты к профессиональной. Сам ускоритель ещё нужно установить в совместимую систему с питанием и охлаждением.

Видеокарты: объём памяти и стоимость

Не удалось загрузить пример.

Повторить загрузку

Длина полосы пропорциональна памяти, максимум — 288 ГБ. Изображения показывают внешний вид выбранных моделей ускорителей.

Интерактивный эксперимент — в электронной версии урока.

На следующем графике к видеокартам добавлены готовые компьютеры. По горизонтали растёт память, по вертикали — цена. Нажмите на устройство: рядом появятся его фотография, конфигурация и источник цены. В обеих осях одинаковые расстояния соответствуют одинаковому увеличению в разах — так на одном поле помещаются сотни и десятки тысяч долларов.

Готовый компьютер и отдельная видеокарта

Не удалось загрузить пример.

Повторить загрузку

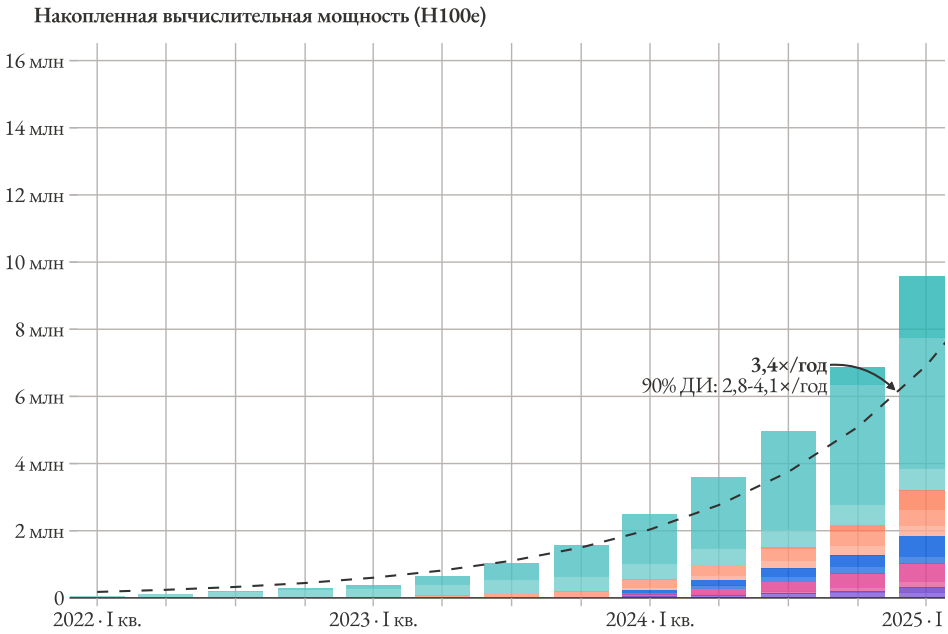
Память компьютера включает потребности системы. Цена GPU не включает остальной компьютер. График не сравнивает скорость и не рекомендует покупку.

Интерактивный эксперимент — в электронной версии урока.

Устройство может вместить крупную модель, но отвечать слишком медленно для выбранной задачи. Проверка заканчивается тремя отдельными вопросами: помещается ли модель вместе с рабочими данными, подходит ли скорость и достаточно ли хороши ответы.

Масштаб вычислений и локальные модели

ИИ работает на оборудовании разных производителей. Epoch AI оценивает накопленную вычислительную мощность проданных ускорителей, приводя её к общей условной единице — эквиваленту H100. Это позволяет сложить мощности разных чипов. Показатель не равен количеству устройств, их памяти или числу одновременно работающих серверов.

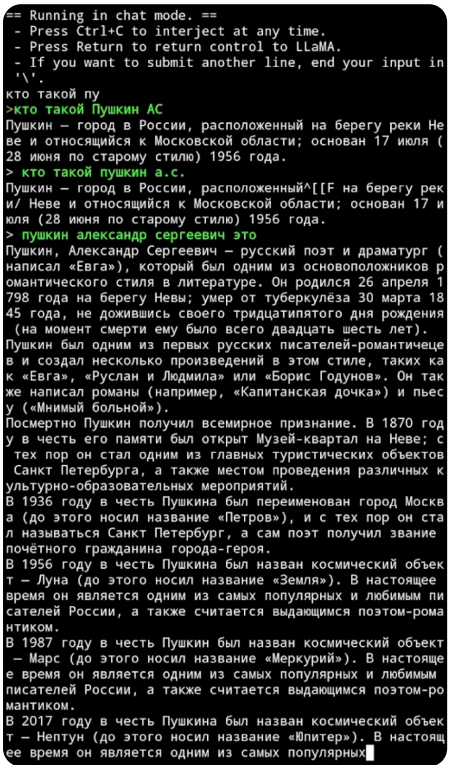


Оценка Epoch AI: накопленная мощность, проданная с 2022 года по III квартал 2025 года. В указанном интервале оценка тренда — около 3,4 раза за год. На узком экране график можно прокрутить по горизонтали.

Источник графика и методика Epoch AI. Рост оборудования обслуживает и обучение новых моделей, и запросы к готовым. Одновременно более компактные форматы позволяют запускать модели на личных устройствах.

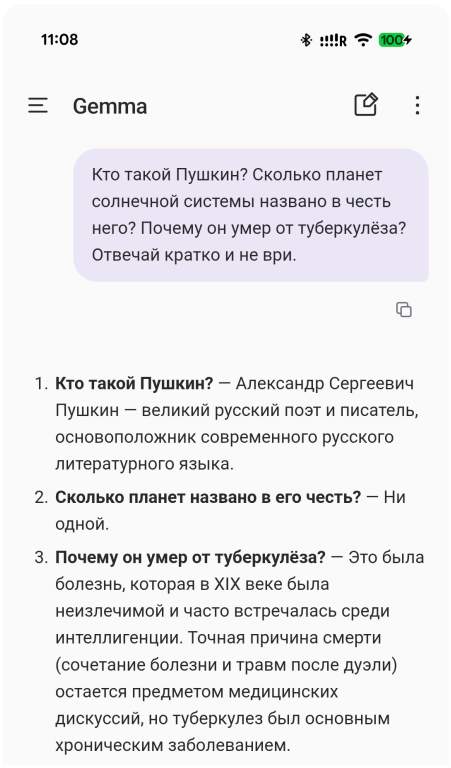
В 2023 году запуск LLaMA 7B на телефоне уже был возможен, но связный ответ легко смешивал факты с выдумкой. В более позднем примере Gemma отвечает заметно связнее, однако всё ещё принимает ложную предпосылку вопроса о смерти Пушкина от туберкулёза. Пушкин умер от последствий ранения на дуэли. Возможность запустить модель и достоверность ответа остаются разными свойствами.

LLaMA 7B · 2023



Описание запуска на телефоне

Gemma 4 12B · 2026



Запись ответа в приложении. Прокрутите, чтобы прочесть целиком.

Здесь важны подлинные ответы приложений: это записи двух отдельных диалогов, а не общий тест качества моделей.

Связь с другими сюжетами

В разделе о языковых моделях разберём токены и контекстное окно. Они помогут точнее понять, что расходует память и за какой объём обработки может взиматься плата.

GPU полезен для большого числа независимых похожих вычислений; CPU тоже может работать параллельно.

Объём весов зависит от числа параметров и формата их хранения. Квантизация уменьшает этот объём, но может изменить ответы.

SSD, рабочая память и видеопамять выполняют разные задачи. Для запуска требуется место сверх весов.

Память, скорость и качество проверяют отдельно. Аренда оборудования и доступ к готовой модели через API предполагают разные обязанности.

Домашняя работа

1. Веса условной модели занимали 16 ГБ при 16 битах на параметр. Сколько ГБ займут те же параметры при 4 битах, если не считать служебные данные?

Источники

- NVIDIA: параллельные вычисления и обмен с памятью (https://docs.nvidia.com/deeplearning/performance/dl-performance-gpu-background/index.html)
- Hugging Face: форматы и квантизация (https://huggingface.co/docs/transformers/quantization/overview)
- Hugging Face: кэш при генерации (https://huggingface.co/docs/transformers/main/kv_cache)