DOTE 6635: Artificial Intelligence for Business Research

Inference

Renyu (Philip) Zhang

1

OpenAI API Prices

OpenAI Prices: https://openai.com/api/pricing/

Similar pricing scheme for Claude, Grok, DeepSeek, Qwen, etc.

OpenAI o1
Frontier reasoning model that supports tools, Structured Outputs, and vision | 200k context length

Price
Input:
$15.00 / 1M tokens
Cached input:
$7.50 / 1M tokens
Output:
$60.00 / 1M tokens

OpenAI o3-mini
Small cost-efficient reasoning model that's optimized for coding, math, and science, and supports tools and Structured Outputs | 200k context length

Price
Input:
$1.10 / 1M tokens
Cached input:
$0.55 / 1M tokens
Output:
$4.40 / 1M tokens

Save 50% on inputs and outputs with the Batch API and run tasks asynchronously over 24 hours.

GPT-4.5
Largest GPT model designed for creative tasks and agentic planning, currently available in a research preview. | 128k context length

Price
Input:
$75.00 / 1M tokens
Cached input:
$37.50 / 1M tokens
Output:
$150.00 / 1M tokens

GPT-4o
High-intelligence model for complex tasks | 128k context length

Price
Input:
$2.50 / 1M tokens
Cached input:
$1.25 / 1M tokens
Output:
$10.00 / 1M tokens

GPT-4o mini
Affordable small model for fast, everyday tasks | 128k context length

Price
Input:
$0.150 / 1M tokens
Cached input:
$0.075 / 1M tokens
Output:
$0.600 / 1M tokens

ChatGPT's User Experience: What is Behind the Decline in Intelligence?
ChatGPT Bugs chatgpt

daixin0906 Jan 6

Since the beginning of this year, I have noticed some significant changes in the functionality and performance of ChatGPT, especially in terms of its intelligence and depth of reasoning. Once, whether as a work assistant or a daily conversational partner, ChatGPT left a deep impression on me. But now, with the use of the GPT-4o model and GPT-01, I can't help but feel that their performance is far below the previous versions. This article will discuss this change from several perspectives.

Jan 6
1 / 19
Jan 6
Back

https://community.openai.com/t/chatgpts-user-experience-what-is-behind-the-decline-in-intelligence/1081511

- **Key question**: How to make LLM inference more efficient and cost-effective?

2

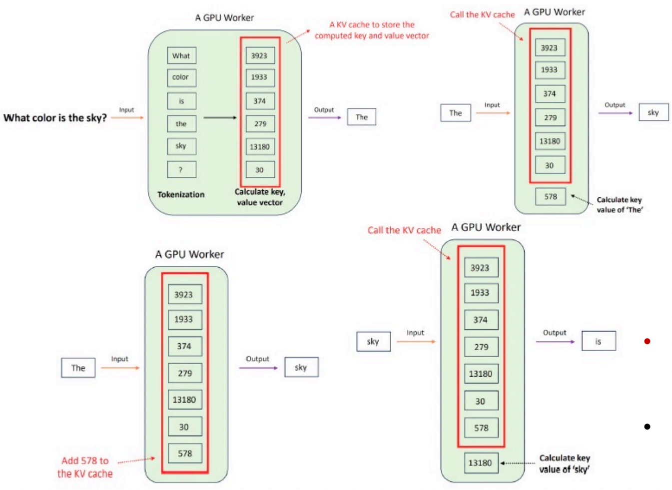
2

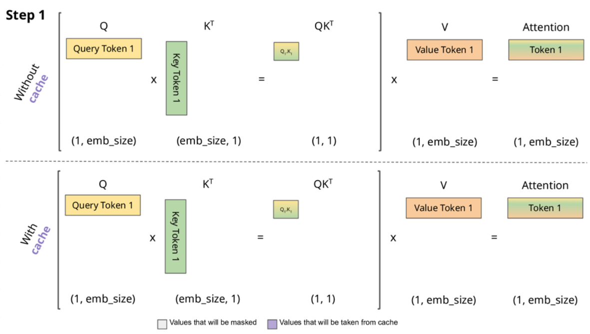
Agenda

- KV-Cache
- Quantization
- DeepSeek Inference System
- OR for LLM Inference

KV Cache

Hugging Face KV Cache Intro: https://huggingface.co/blog/not-lain/kv-caching
https://medium.com/@plienhar/llm-inference-series-2-the-two-phase-process-behind-llms-responses-1ff1ff021cd5

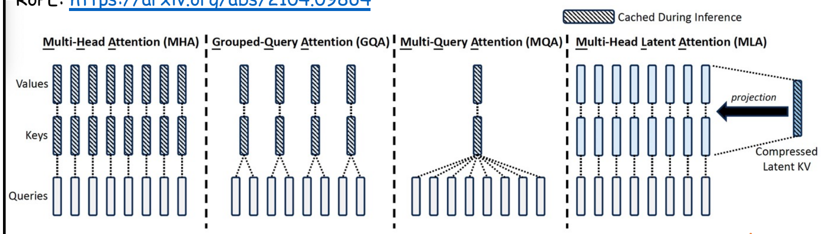




- **Essence of KV Cache**: Use memory to substitute compute.
- KV Cache is linear in context window length.

Multi-Head Latent Attention

DeepSeek-V3: https://arxiv.org/abs/2412.19437v1; https://www.bilibili.com/video/BV1HqFQezEMt
RoPE: https://arxiv.org/abs/2104.09864



Down-Projection Matrix

$$\mathbf{c}_t^{KV} = W^{DKV}\mathbf{h}_t,$$
$$\mathbf{k}_t^{C} = W^{UK}\mathbf{c}_t^{KV},$$
$$\mathbf{v}_t^{C} = W^{UV}\mathbf{c}_t^{KV},$$

Up-Projection Matrix

A low-rank joint compression of keys and values.

Compress the query to reduce activation memory.

$$\mathbf{c}_t^{Q} = W^{DQ}\mathbf{h}_t,$$
$$[\mathbf{q}_{t,1}^{C};\mathbf{q}_{t,2}^{C};\ldots;\mathbf{q}_{t,n_h}^{C}] = \mathbf{q}_t^{C} = W^{UQ}\mathbf{c}_t^{Q},$$
$$[\mathbf{q}_{t,1}^{R};\mathbf{q}_{t,2}^{R};\ldots;\mathbf{q}_{t,n_h}^{R}] = \mathbf{q}_t^{R} = \mathrm{RoPE}(W^{QR}\mathbf{c}_t^{Q}),$$
$$\mathbf{q}_{t,i} = [\mathbf{q}_{t,i}^{C};\mathbf{q}_{t,i}^{R}],$$

Attention Mechanism	KV Cache per Token (# Element)	Capability
Multi-Head Attention (MHA)	$2n_h d_h l$	Strong
Grouped-Query Attention (GQA)	$2n_g d_h l$	Moderate
Multi-Query Attention (MQA)	$2d_h l$	Weak
MLA (Ours)	$(d_c + d_h^R)l \approx \frac{9}{2}d_h l$	Stronger

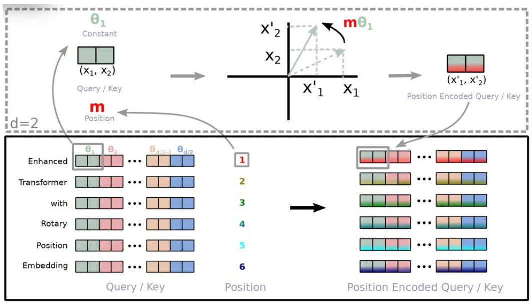


Figure 1: Implementation of Rotary Position Embedding(RoPE).

5

Agenda

- KV-Cache
- Quantization
- DeepSeek Inference System
- OR for LLM Inference

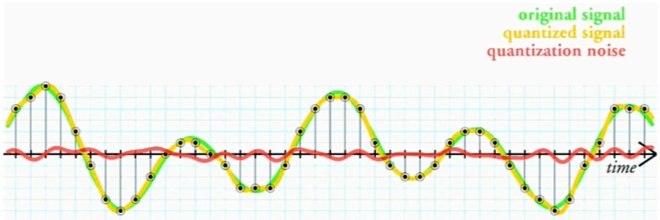
6

6

Quantization

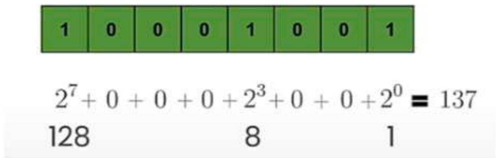
MIT Efficient DL Computing: https://hanlab.mit.edu/courses/2024-fall-65940
Quantization Fundamentals with Hugging Face: https://learn.deeplearning.ai/courses/quantization-fundamentals/

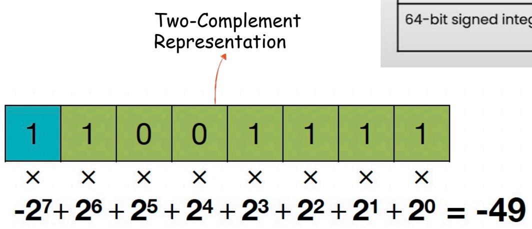
- Quantization: Mapping an input from a large (and continuous) set of values to a smaller (and discrete) set of values.
  - We do quantization to save memory and energy and accelerate compute, especially for LLM inference.



Data Type	torch.dtype
8-bit signed integer	torch.int8
8-bit unsigned integer	torch.uint8
16-bit signed integer	torch.int16
32-bit signed integer	torch.int32
64-bit signed integer	torch.int64

- For unsigned integer data types, $[0, 2^n-1]$.
- For signed integer data types, $[-2^n, 2^{n-1}-1]$.





7

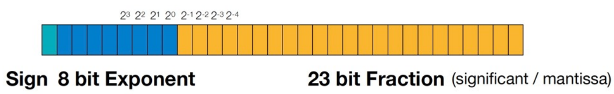
7

Floating Number Representations

MIT Efficient DL Computing: https://hanlab.mit.edu/courses/2024-fall-65940
Quantization Fundamentals with Hugging Face: https://learn.deeplearning.ai/courses/quantization-fundamentals/

- Floating-point numbers:
  - Sign: +/-
  - Exponent: Range
  - Fraction/mantissa: Precision

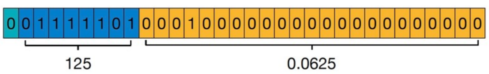
Data Type	torch.dtype	torch.dtype alias
16-bit floating point	torch.float16	torch.half
16-bit brain floating point	torch.bfloat16	
32-bit floating point	torch.float32	torch.float
64-bit floating point	torch.float64	torch.double

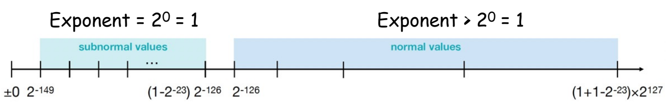


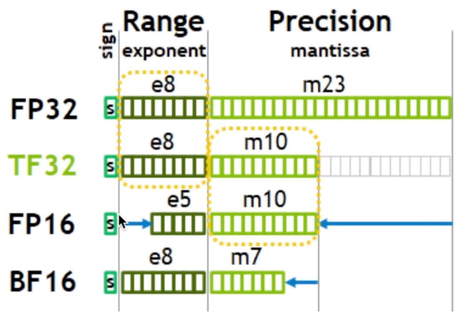
$(-1)^{\text{sign}} \times (1 + \textbf{Fraction}) \times 2^{\textbf{Exponent}-127}$ ⟵ Exponent Bias = 127 = $2^{8-1}-1$

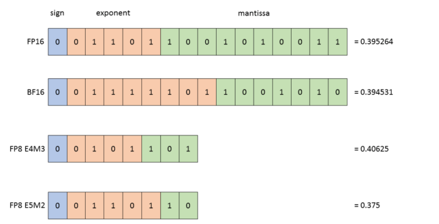
How to represent **0.265625**?

0.265625 = 1.0625 × 2^{-2} = (1 + 0.0625) × $2^{125-127}$









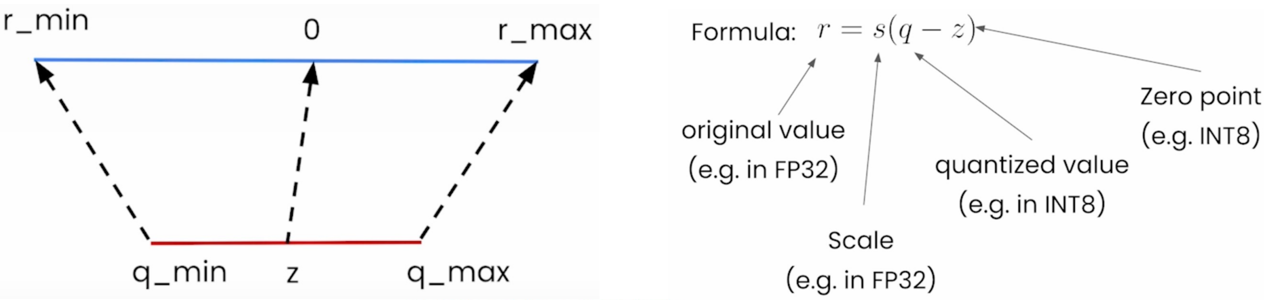
8

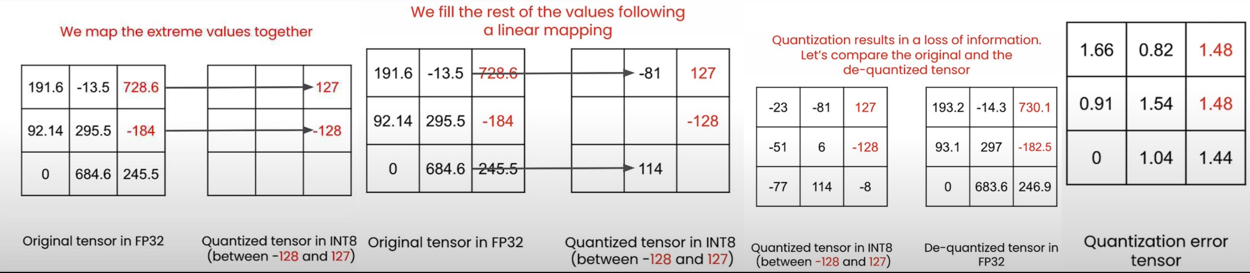
8

Linear Quantization

MIT Efficient DL Computing: https://hanlab.mit.edu/courses/2024-fall-65940
Quantization Fundamentals with Hugging Face: https://learn.deeplearning.ai/courses/quantization-fundamentals/

- Use a linear mapping to represent a number in high-precision type (FP32) in low-precision type (INT8).



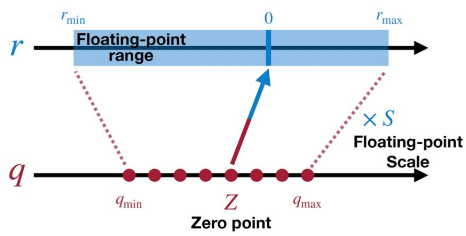


9

Linear Quantization

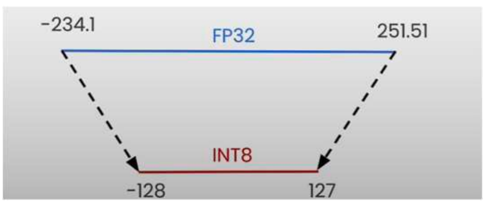
MIT Efficient DL Computing: https://hanlab.mit.edu/courses/2024-fall-65940
Quantization Fundamentals with Hugging Face: https://learn.deeplearning.ai/courses/quantization-fundamentals/

- How do we determine the scale s and zero point z?

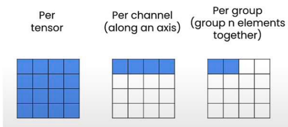


$$r_{\max} = S\left(q_{\max} - Z\right)$$
$$r_{\min} = S\left(q_{\min} - Z\right)$$
$$\downarrow$$
$$r_{\max} - r_{\min} = S\left(q_{\max} - q_{\min}\right)$$
$$S = \frac{r_{\max} - r_{\min}}{q_{\max} - q_{\min}}$$

$$r_{\min} = S\left(q_{\min} - Z\right)$$
$$\downarrow$$
$$Z = q_{\min} - \frac{r_{\min}}{S}$$
$$\downarrow$$
$$Z = \text{round}\left(q_{\min} - \frac{r_{\min}}{S}\right)$$



- Per-channel and per-group quantization.
- Quantization of weights and activations.
- Quantization-aware training.



10

Quantization + LoRA: QLoRA

Stanford CS224N: https://web.stanford.edu/class/cs224n/
QLoRA Paper: https://arxiv.org/abs/2305.14314

Qlora: Efficient finetuning of quantized llms
T Dettmers, A Pagnoni, A Holtzman… - Advances in neural …, 2023 - proceedings.neurips.cc
… We present **QLORA**, an efficient finetuning approach that reduces … **QLORA** backpropagates gradients through a frozen, 4-bit … **QLORA** introduces a number of innovations to save memory …
☆ Save 99 Cite Cited by 2449 Related articles All 9 versions

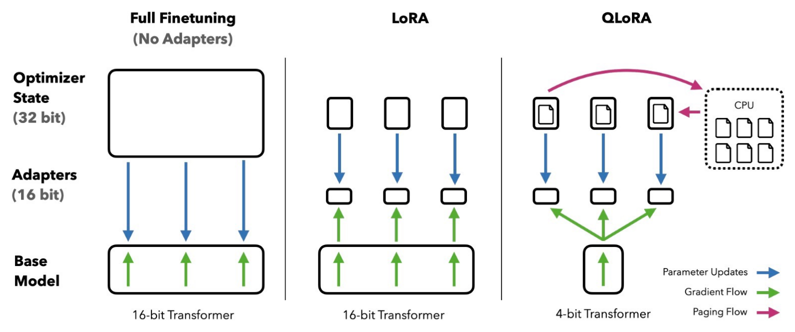
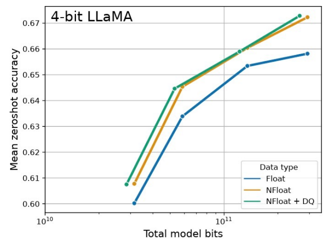


Figure 1: Different finetuning methods and their memory requirements. QLoRA improves over LoRA by quantizing the transformer model to 4-bit precision and using paged optimizers to handle memory spikes.

To further save memory, adopt double-quantization (DQ).

- QLoRA improves over LoRA by quantizing the transformer to 4-bit precision and using paged optimizer to handle memory.
- 4-bit NormalFloat (NF4)
  - Data type suitable for normally distributed weights.



11

11

Agenda

- KV-Cache
- Quantization
- DeepSeek Inference System
- OR for LLM Inference

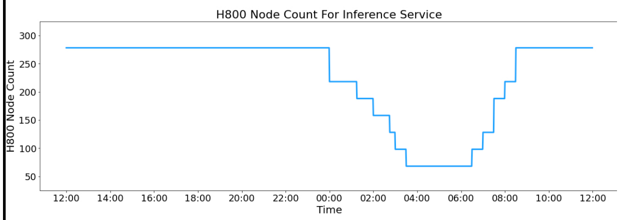
12

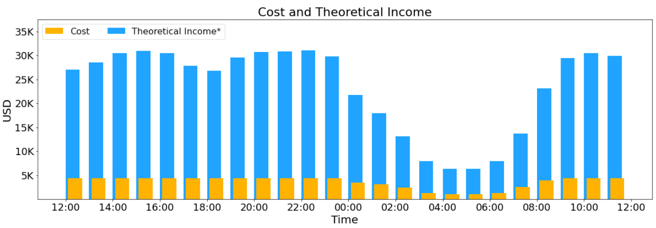
12

DeepSeek-V3/R1 Inference System

GitHub: https://github.com/deepseek-ai/open-infra-index/blob/main/202502OpenSourceWeek/day_6_one_more_thing_deepseekV3R1_inference_system_overview.md

- Extremely optimized for high throughput and low latency: cross-node Expert Parallelism (EP).
  - Leverage EP to scale batch size.
  - Hide communication latency behind computation.
  - Perform load balancing.
- Served with 278 8-H800 GPU nodes; average occupancy 226.75 nodes; each with throughput ~73.7k tokens/s for input during prefilling and ~14.8k tokens/s for output during decoding.
- Daily input tokens: 608B (342B hit the KV cache)
- Daily output tokens: 168B; 20-22 tokens/s; average KV-cache length per output: 4,989 tokens.





* The theoretical income is calculated based on R1's standard API pricing, taking into account all tokens across web, APP, and API. It is not our actual income.

Daily cost = $87,072; Daily Revenue under the R1-API pricing = $562,027, i.e., 545% profit margin

13

13

Agenda

- KV-Cache
- Quantization
- DeepSeek Inference System
- OR for LLM Inference

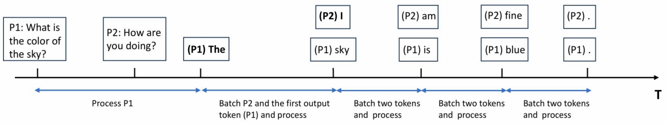
14

14

OR for LLM Inference

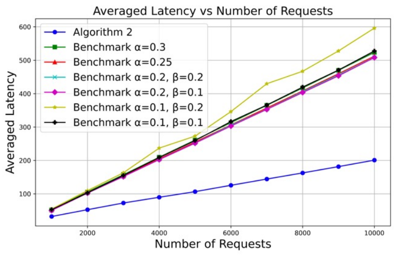
Fundamental Modeling for LLM Inference with Exploding KV Cache Demands

Patrick Jaillet
Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, jaillet@mit.edu
Jiashuo Jiang
HKUST, jsjiang@ust.hk
Chara Podimata
Sloan School of Management, Massachusetts Institute of Technology podimata@mit.edu
Zijie Zhou*
Operations Research Center, Massachusetts Institute of Technology, zhou98@mit.edu



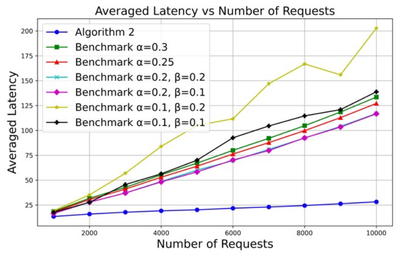
- Given the KV-cache memory limit, how to batch different prompts and output tokens to minimize total end-to-end latency.

- Proposed scheduling algorithm (with provable constant regret): Prioritize the prompt with the smallest predicted number of output tokens, subject to the KV-cache limit constraint.
  - Benchmark: alpha-protection first-come-first-serve, beta-clearing when reaching KV-cache limit.



High-Demand

Low-Demand



15

15