So you want your private LLM at home? A survey and benchmark of methods for efficient GPTs

Swiss Conference on Data Science
May 31. 2024

Zürcher Hochschule für Angewandte Wissenschaften

zh aw

rwai.ch

Team

Pascal Sager
sage@zhaw.ch

Yassine Taoudi-Benchekroun
ytaoudi@ethz.ch

Prof. Dr. Benjamin F. Grewe
bgrewe@ethz.ch

Prof. Dr. Thilo Stadelmann
stdm@zhaw.ch

Motivation - LLMs go open-source

May 2020 - GPT-3 is announced

Nov 2022 Chat GPT goes live

Feb 2023 Llama 1 is leaked

- Meta AI gets all the attention
- Triggers a shift towards open-sourcing models (almost over night)
- Causes an unprecedented burst of innovations around LLMs

Ollama

LlamaIndex

Stanford Alpaca

Motivation - AI hardware is expensive

On-Premise:

Model	*VRAM*	*Price*
RTX 4090	24 GB	1’800 CHF
RTX A6000	48 GB	4’300 CHF
H100*	80 GB	~30’000 $

*good luck finding one

Cloud (azure):

NC4as T4 v3	4	28 GiB	1X T4	$383.98/ month
NC6s v3	6	112 GiB	1X V100	$3,069.65/ month
ND96asr A100 v4	96	900 GiB	8x A100 (NVlink)	$19,853.810/ month

Key technologies for resources constrained LLM usage

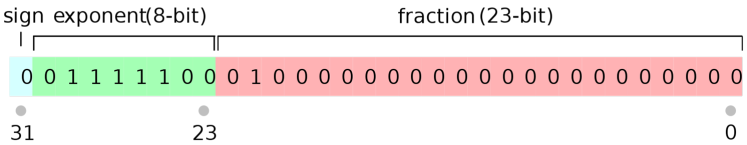
- **Quantization**
- **Low-rank approximators**
- Pruning - *find and delete inconsequential parameter*
- Gradient accumulation - *retain gradients to virtually increase batch size*
- Gradient checkpointing - *offload computed gradients to RAM*
- Parameter offloading - *share load between GPU and CPU*

Key technologies for resources constrained LLM usage

- **Quantization**
- **Low-rank approximators**
- Pruning - *find and delete inconsequential parameter* ▶ Very promising
- Gradient accumulation - *retain gradients to virtually increase batch size* ▶ Only do this if you **really** have to
- Gradient checkpointing - *offload computed gradients to RAM* ▶ Most useful in large operations
- Parameter offloading - *share load between GPU and CPU*

Quantization

- Move to 16 / 8 / 4 / 2 bit float representation
- Can we do better than naive truncate?
  - BnB ▶ ML first encoding (nf4)
  - AWQ ▶ protect important weights
  - GPTQ ▶ use second-order info

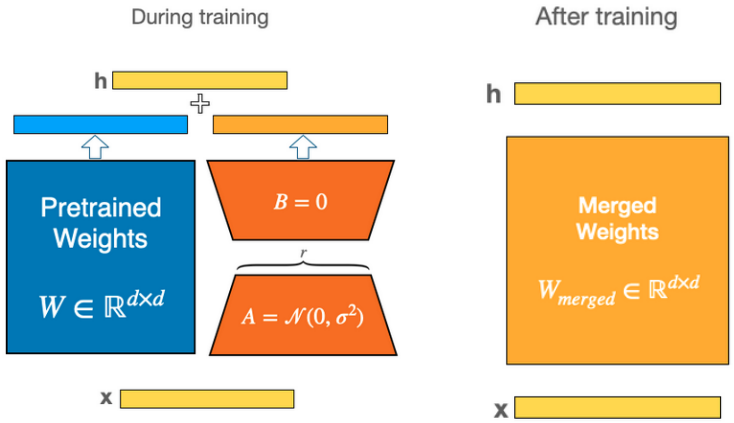


source: https://courses.physics.illinois.edu/cs357/sp2020/notes/ref-4-fp.html

more on quantization: https://huggingface.co/blog/merve/quantization

Low-rank approximators (LoRA)

- Freeze all base weights
- Train a of differential update blocks (adapters)
- Adapters are low-rank matrix matrix decompositions
  - r = 64 << d = 4096



source & more on LoRA: https://huggingface.co/docs/peft/main/en/conceptual_guides/lora

Benchmark setup and goals

what's the tradeoff here?

- Computational performance
  - VRAM consumption
  - speed

- Model quality
  - MT-bench grading via GPT-4*
  - 5 Selected questions graded via eight judges

MT-bench examples
Writing:
Round 1: Compose an engaging travel blog post about a recent trip to Hawaii, highlighting cultural experiences and must-see attractions.

Round 2: Rewrite your previous response. Start every sentence with the letter A.

Reasoning:
Round 1: Thomas is very healthy, but he has to go to the hospital every day. What could be the reasons?

Round 2: Can you explain why the above question is interesting?

all experiments use Llama2 7B / 4-bit quant

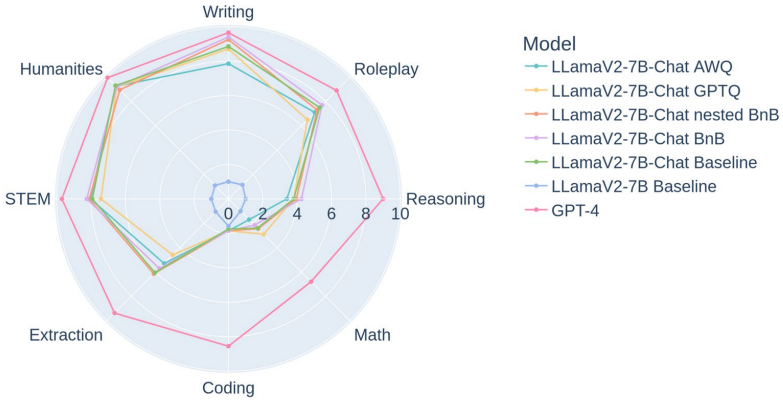
*. Zheng et al., “Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena,” Oct. 2023. arXiv:2306.05685.

Benchmark results - quantized inference

Computational performance:

	VRAM [GB]	speed [token/s]	
		A100	T4
baseline	**33.8**	36.06	oom
BnB	**8.4**	25.45	26.32
AWQ	**9.1**	86.66	26.32
GPTQ	**9**	34.20	28.33

Model Quality:



	Q1	Q2	Q3	Q4	Q5	**avg**
baseline	3.75	1.25	4.5	4	3.25	**3.35**
BnB	2.25	2	2.75	3.5	4	**2.9**
AWQ	1.5	1.5	1.5	2.5	2.75	**1.95**
GPTQ	1.25	1.75	4.75	3.75	4	**3.1**

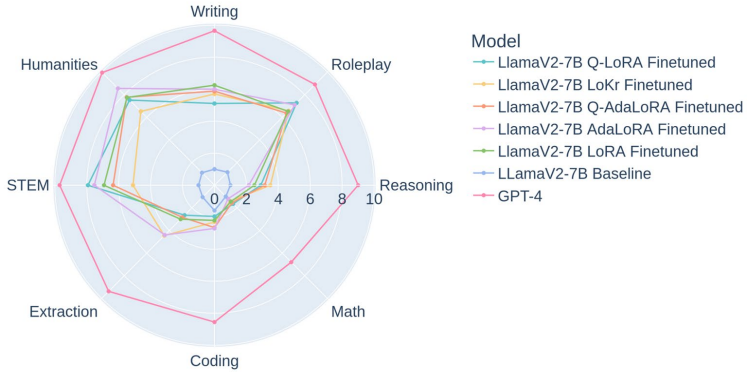
Benchmark results - LoRA training

train base Llama2 to behave as an assistant

Computational performance:

	VRAM [GB]	speed [token/s]	
		A100	T4
baseline	**>> 40**	oom	oom
LoRA	**33.4**	0.217	oom
AdaLoRA	**32.7**	0.203	oom
LoKr	**39.5**	0.15	oom
Q-LoRA	**13.5**	0.178	0.04
Q-AdaLoRA	**12.7**	0.168	0.04

Model Quality:



	Q1	Q2	Q3	Q4	Q5	avg
LoRA	2.5	2.25	0.75	3.75	3.24	**2.5**
Q-LoRA	2	0.75	2.25	3.25	1.75	**2**
AdaLoRA	1.5	1.25	2.5	3	2.25	**2.1**
AdaLoRA	1.25	1.75	1.25	1.5	2.25	**1.6**
LoKr	2.25	1.5	1	0.5	3	**1.65**

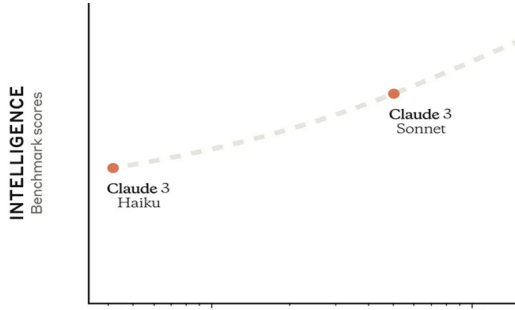
Conclusions

It is possible to effectively train a powerful LLM on a single consumer GPU

Quantization and LoRA are no free lunch (Bfloat16 is as close as it gets).

Automated evaluation of LLMs is in its infancy (at best)

gold standard: https://chat.lmsys.org/?leaderboard



source: https://www.anthropic.com/news/claude-3-family

Take away - is self-hosting useful?

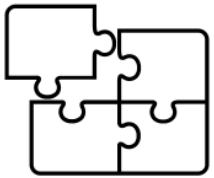
Privacy

Capability

Cost

(cloud-)hardware is very expensive, depending on usage apis can be cheaper

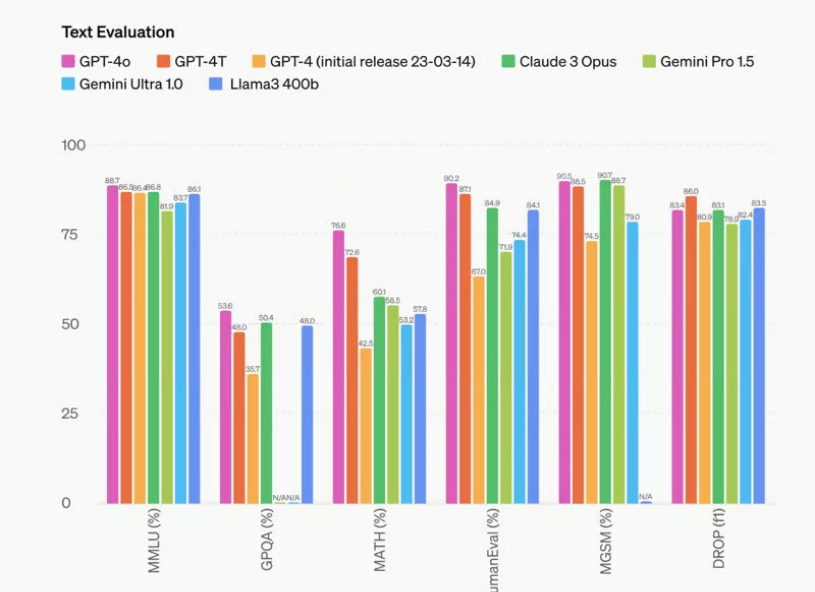
Fit to task



In-context learning and prompt engineering are becoming very powerful - however fine tuning can be necessary esp. when moving away from natural language.

What happened in the meantime? - outlook

- Llama2 (July 2023) is old news
- Every week a new LLM gets open sourced (phi-3, gemma, mistral….)
- Running LLMs has become very easy
  - on colab: https://github.com/tuggeluk/LMM_at_home
  - using https://ollama.com/
- Open-source multi modality is still in its infancy
- Llama3 400B is on the horizon (hopefully?):



Thanks for your attention!

Zürcher Hochschule
für Angewandte Wissenschaften

zh
aw

rwai.ch

More questions? - find me at the apero or send me an email

Research:
tugg@zhaw.ch

Business:
lukas.tuggener@rwai.ch