Finetuning LLMs on custom datasets

Aniket Maurya, Developer Advocate at Lightning AI

October 2023

Agenda

- Overview of LLMs
- Parameter efficient finetuning with instruction dataset
- Training on consumer GPUs

What are LLMs

What are LLMs

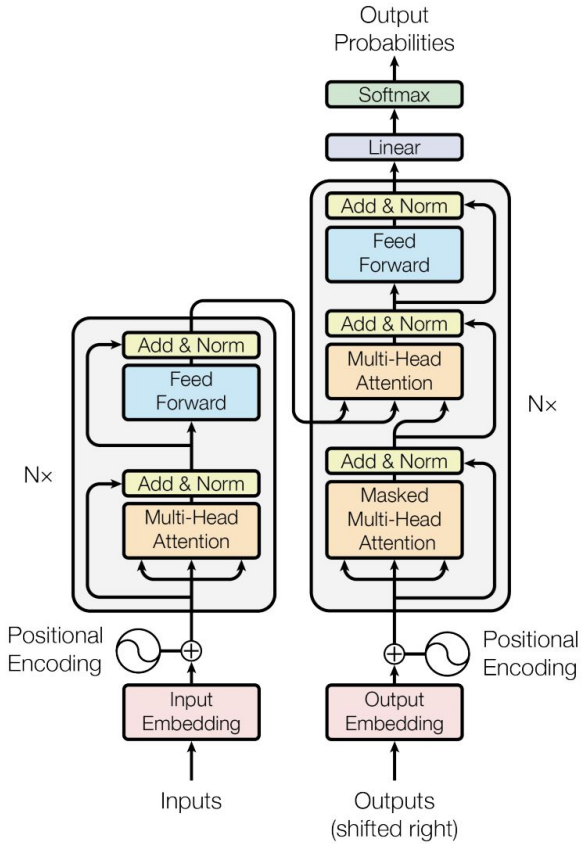
```
query = "Capital of"

output = "Capital of"
for i in range(MAX_GENERATED_TOKENS):
    output = output + LLM(output)
```

0. output = Capital of
1. output = Capital of France
2. output = Capital of France is
3. output = Capital of France is Paris

What are LLMs

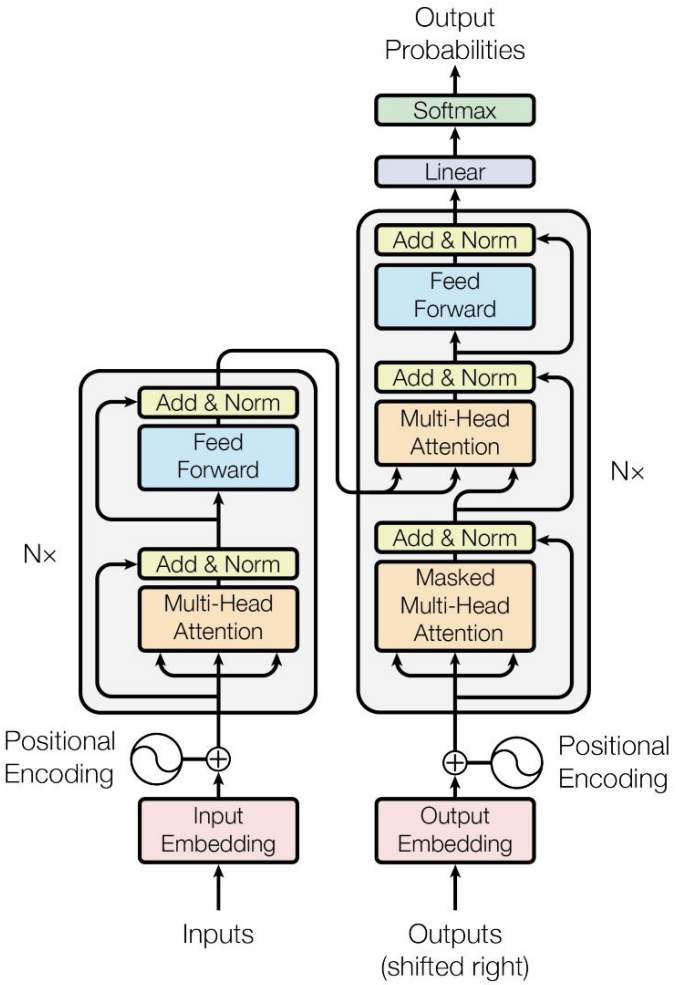
```
query = "Capital of"

output = "Capital of"
for i in range(MAX_GENERATED_TOKENS):
    output = output + LLM(output)
```

0. output = Capital of
1. output = Capital of France
2. output = Capital of France is
3. output = Capital of France is Paris

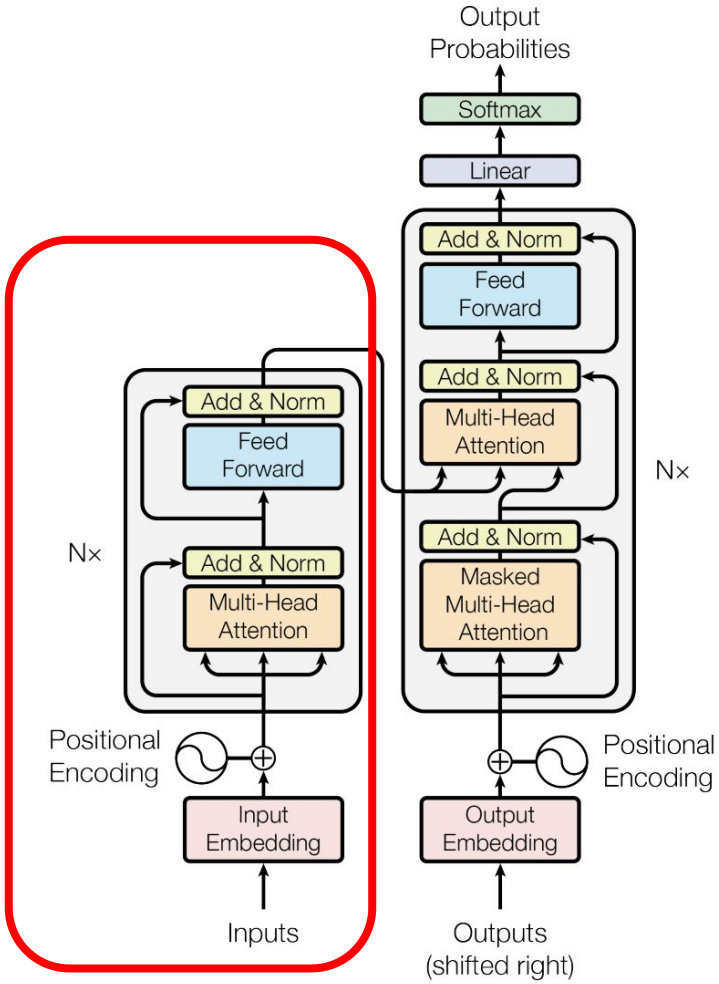


Source: Attention is All you Need

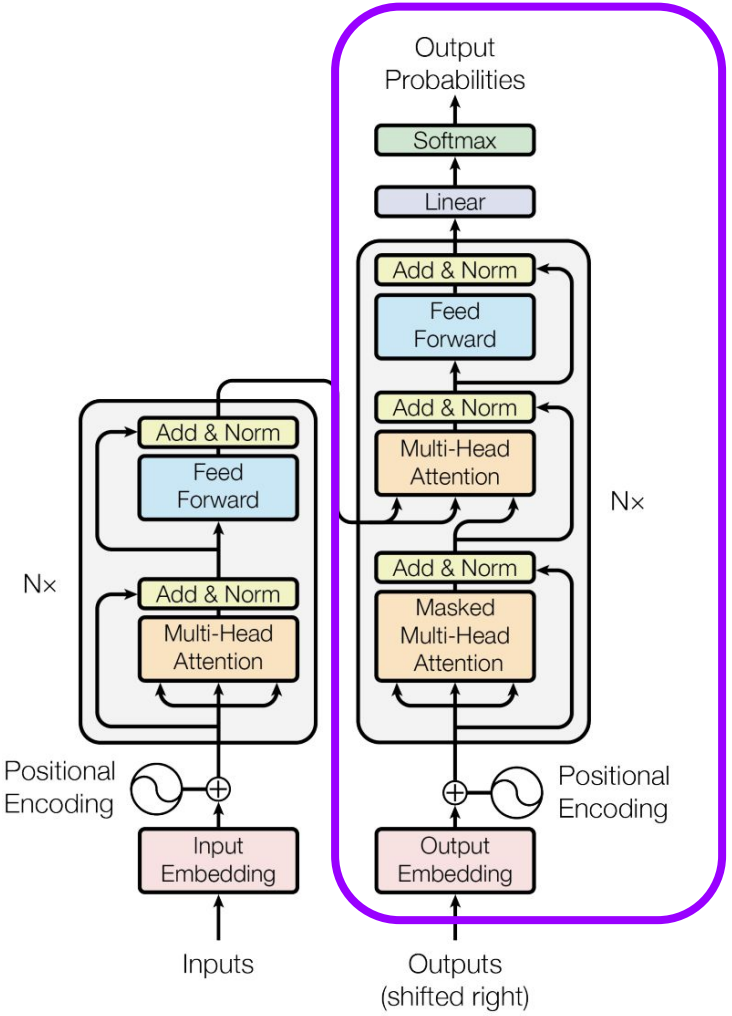
What are LLMs



What are LLMs

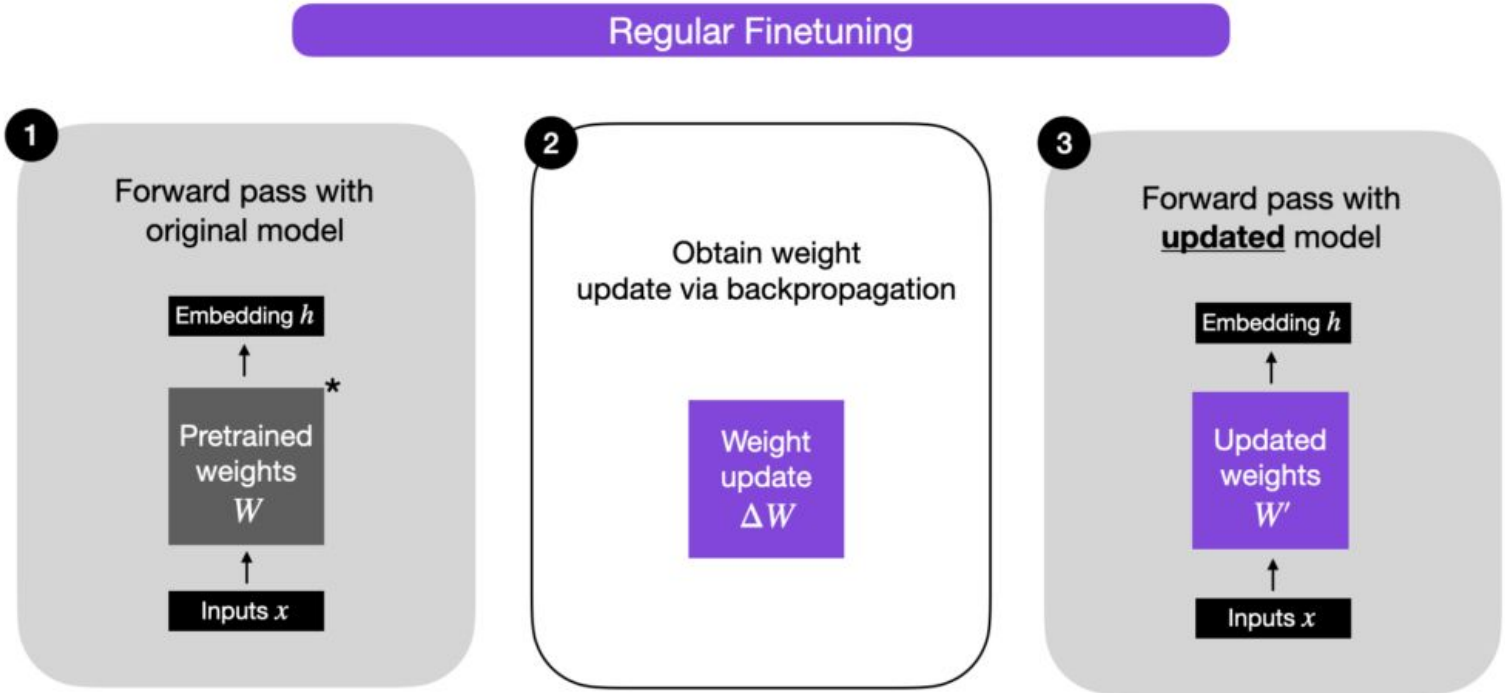


What are LLMs



*Decoder

Finetuning LLMs

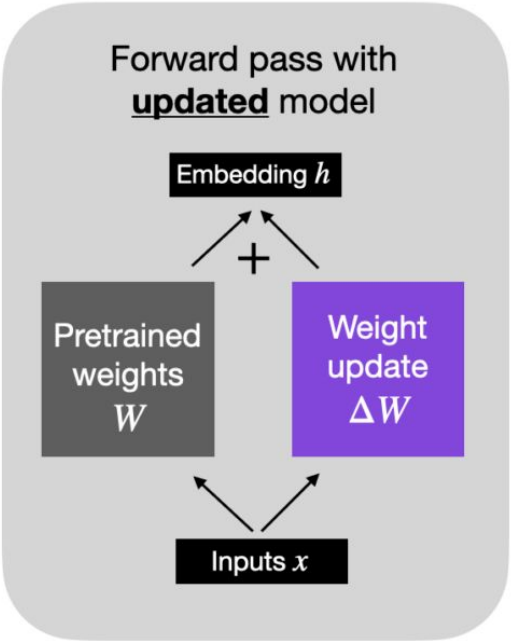


* The pretrained model could be any LLM, e.g., an encoder-style LLM (like BERT) or a generative decoder-style LLM (like GPT)

Source : https://lightning.ai/pages/community/tutorial/lora-llm/

Finetuning LLMs

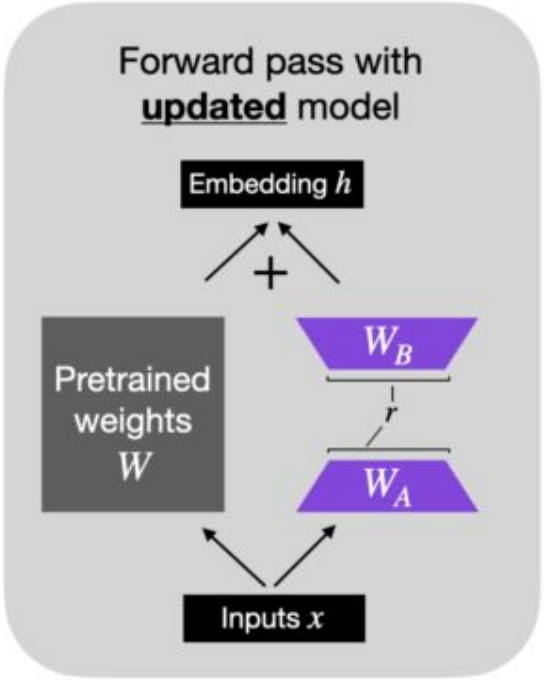
Alternative formulation (regular finetuning)



Source : https://lightning.ai/pages/community/tutorial/lora-llm/

Parameter Efficient Finetuning

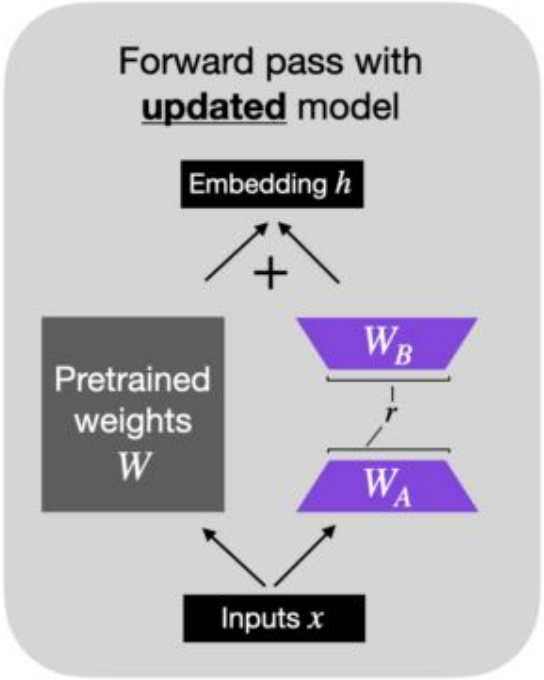
LoRA weights, W_A and W_B, represent ΔW



Source : https://lightning.ai/pages/community/tutorial/lora-llm/

Parameter Efficient Finetuning

LoRA weights, W_A and W_B, represent ΔW

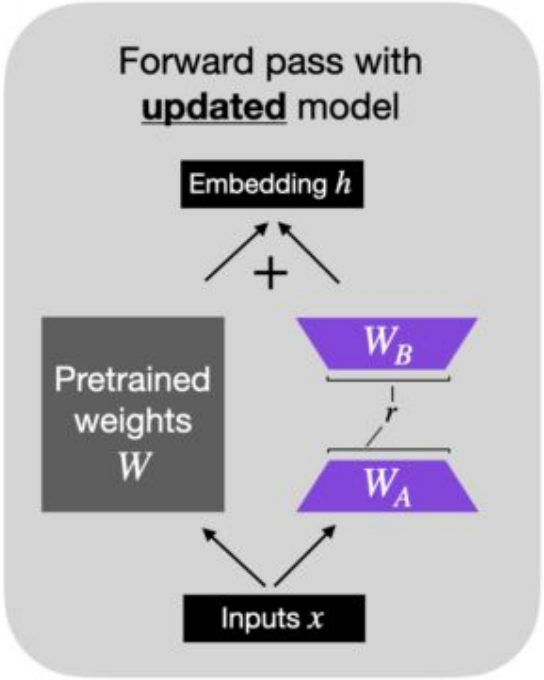


- W = 100 x 500
- Wa = 100x5, Wb=5x500
- r = 5

Source : https://lightning.ai/pages/community/tutorial/lora-llm/

Parameter Efficient Finetuning

LoRA weights, W_A and W_B, represent ΔW



- W = 100 x 500
- Wa = 100x5, Wb=5x500
- r = 5

Old parameters = 50, 000
New parameters = 3,000

Parameter Efficient Finetuning

LoRA can even outperform full finetuning training only 2% of the parameters

Full finetuning

Only tune bias vectors

Prompt tuning

Prefix tuning

Model&Method	# Trainable Parameters	WikiSQL Acc. (%)	MNLI-m Acc. (%)	SAMSum R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 ($\text{Adapter}^{\text{H}}$)	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 ($\text{Adapter}^{\text{H}}$)	40.1M	73.2	**91.5**	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1

ROUGE scores

Table 4: Performance of different adaptation methods on GPT-3 175B. We report the logical form validation accuracy on WikiSQL, validation accuracy on MultiNLI-matched, and Rouge-1/2/L on SAMSum. LoRA performs better than prior approaches, including full fine-tuning. The results on WikiSQL have a fluctuation around $\pm 0.5\%$, MNLI-m around $\pm 0.1\%$, and SAMSum around $\pm 0.2/\pm 0.2/\pm 0.1$ for the three metrics.

Why Finetune LLMs

- Remove untruthfulness and toxicity
- Customize the output and tone of language
- Privacy and control

Finetuning Challenges

- Dataset not available
- Computationally expensive
- Need to re-train with time

Finetuning LLMs on instruction dataset

```
[
    {
        "instruction": "Write a limerick about a
                        pelican.”,
        "input": "",
        "output": "There once was a pelican so fine,
                   \nHis beak was as colorful as
                   sunshine,\nHe would fish all day,\nIn
                   a very unique way,\nThis pelican was
                   truly divine!\n\n\n"
    },
    {
        "instruction": "Identify the odd one out from
                        the group.",
        "input": "Carrot, Apple, Banana, Grape",
        "output": "Carrot\n\n"
    },
]
```

Finetuning LLMs

- Setup model
- Prepare data
- Finetune the model

Finetuning LLMs

- Setup model
- Prepare data
- Finetune the model

Finetuning LLMs

- Setup model
- Prepare data
- Finetune the model

lit-gpt

Public

Edit Pins | Watch 47 | Fork 327 | Starred 3.4k

main | 27 branches | 0 tags

Go to file | Add file | Code

Andrei-Aksionov Fix unused arguments (#657) 6ed6a15 18 hours ago 532 commits

.github	Use all requirements file in CI (#621)	2 weeks ago
chat	Type hints (#633)	2 weeks ago
eval	Use Fabric's quantization (#596)	3 weeks ago
finetune	Fix unused arguments (#657)	18 hours ago
generate	Fix --quantization -> --quantize (#656)	4 days ago
lit_gpt	Automated code fixes with type annotations check. (#646)	last week
notebooks	Torch 2.1 installation instructions (#611)	3 weeks ago
pretrain	Fix typing annotation	last week
quantize	Use Fabric's quantization (#596)	3 weeks ago
scripts	Fix unused arguments (#657)	18 hours ago
tests	Fix unused arguments (#657)	18 hours ago

About

Hackable implementation of state-of-the-art open-source LLMs based on nanoGPT. Supports flash attention, 4-bit and 8-bit quantization, LoRA and LLaMA-Adapter fine-tuning, pre-training. Apache 2.0-licensed.

Readme

Apache-2.0 license

Activity

3.4k stars

47 watching

327 forks

Report repository

Releases

No releases published

Create a new release

Lit-GPT

Created by Lightning AI

Model and usage	Model size	Reference
Meta AI Llama 2	7B, 13B, 70B	Touvron et al. 2023
Stability AI FreeWilly2 (Stable Beluga 2)	70B	Stability AI 2023
Stability AI StableCode	3B	Stability AI 2023
TII UAE Falcon	7B, 40B, 180B	TII 2023
OpenLM Research OpenLLaMA	3B, 7B, 13B	Geng & Liu 2023
LMSYS Vicuna	7B, 13B, 33B	Li et al. 2023
LMSYS LongChat	7B, 13B	LongChat Team 2023
Together RedPajama-INCITE	3B, 7B	Together 2023
EleutherAI Pythia	{70,160,410}M, {1,1.4,2.8,6.9,12}B	Biderman et al. 2023
StabilityAI StableLM	3B, 7B	Stability AI 2023
Platypus	7B, 13B, 70B	Lee, Hunter, and Ruiz 2023
NousResearch Nous-Hermes	7B, 13B, 70B	Org page
Meta AI Code Llama	7B, 13B, 34B	Rozière et al. 2023
Microsoft Research phi-1.5	1.3B	Li et al. 2023
Mistral AI Mistral	7B	Mistral website

Lit-GPT

- 4-bit quantized finetuning and inference
- Minimal code, easy to debug and hack
- TPU support
- Flash-Attention 2

Finetuning Llama on instruction dataset

instruction	input	output
When did Virgin Australia start operating?	Virgin Australia, the trading name of Virgin Australia Airlines Pty Ltd, is an Australian-based airline. It is the largest airline by fleet size to use the Virgin brand. It commenced services on 31 August 2000 as Virgin Blue, with two aircraft on a single route. It suddenly found itself as a major airline in Australia's domestic market after the collapse of Ansett Australia in September 2001. The airline has since grown to directly serve 32 cities in Australia, from hubs in Brisbane, Melbourne and Sydney.	Virgin Australia commenced services on 31 August 2000 as Virgin Blue, with two aircraft on a single route.
Which is a species of fish? Tope or Rope		Tope
Why can camels survive for long without water?		Camels use the fat in their humps to keep them filled with energy and hydration for long periods of time.
Alice's parents have three daughters: Amy, Jessy, and what's the name of the third daughter?		The name of the third daughter is Alice
When was Tomoaki Komorida born?	Komorida was born in Kumamoto Prefecture on July 10, 1981. After graduating from high school, he joined the J1 League club Avispa Fukuoka in 2000. Although he debuted as a midfielder in 2001, he did not play much and the club was relegated to the J2 League at the end of the 2001 season. In 2002, he moved to the J2 club Oita Trinita. He became a regular player as a defensive midfielder and the club won the championship in 2002 and was promoted in 2003. He played many matches until 2005. In September 2005, he moved to the J2 club Montedio Yamagata. In 2006, he moved to the J2 club Vissel Kobe. Although he became a regular player as a defensive midfielder, his gradually was played less during the summer. In 2007, he moved to the Japan Football League club Rosso Kumamoto (later Roasso Kumamoto) based in his local region. He played as a regular player and the club was promoted to J2 in 2008. Although he did not play as much, he still played in many matches. In 2010, he moved to Indonesia and joined Persela Lamongan. In July 2010, he returned to Japan and joined the J2 club Giravanz Kitakyushu. He played often as a defensive midfielder and center back until 2012 when he retired.	Tomoaki Komorida was born on July 10,1981.

Setup Model

```
python scripts/download.py \
    --repo_id meta-llama/Llama-2-7b-hf

python scripts/convert_hf_checkpoint.py \
    --checkpoint_dir checkpoints/meta-llama/Llama-2-7b-hf
```

Setup Model

```
(lit-gpt) aniket@hyperplane1:~/llms/lit-gpt$ ls /data/aniket/Llama-2-7b-hf
lit_config.json                  pytorch_model.bin.index.json
lit_model.pth                    tokenizer_config.json
pytorch_model-00001-of-00002.bin  tokenizer.json
pytorch_model-00002-of-00002.bin  tokenizer.model
(lit-gpt) aniket@hyperplane1:~/llms/lit-gpt$
```

Prepare Dataset

```
python scripts/prepare_csv.py \
    --csv_path "databricks-dolly-15k.csv" \
    --checkpoint_dir "/data/aniket/Llama-2-7b-hf" \
    --destination_path "data/dolly" \
    --max_seq_length 512
```

Finetune

```
python finetune/lora.py \
    --checkpoint_dir "/data/aniket/Llama-2-7b-hf" \
    --data_dir "data/dolly" \
    --out_dir "out/lora/dolly"
```

CUDA Out Of Memory

Memory Required to load Llama

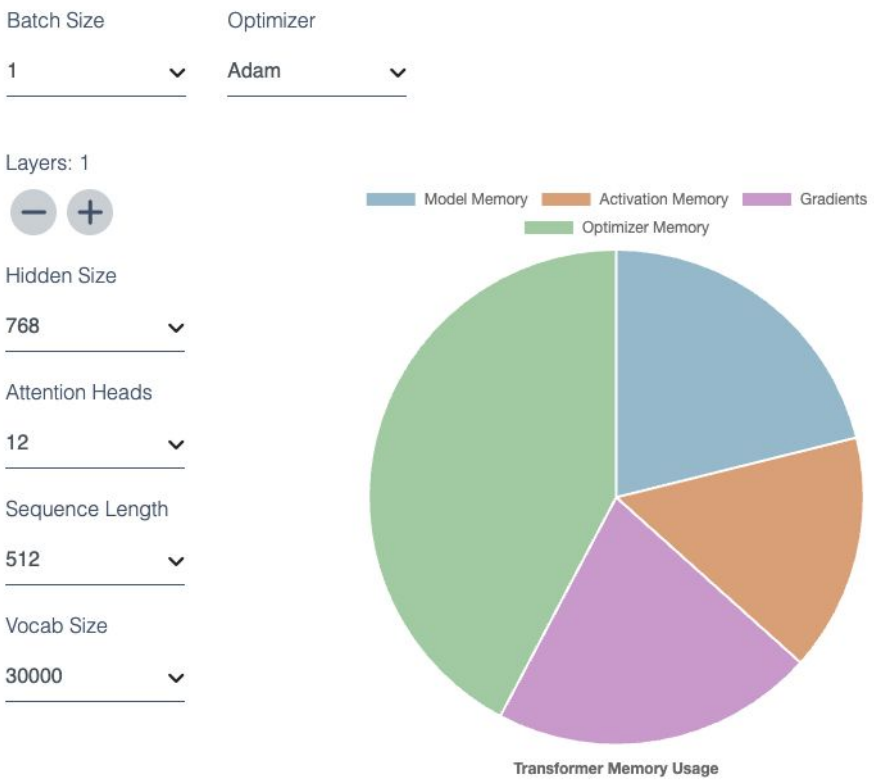
- **Llama 7B, fp32:** ~28GB
- **Llama 7B, fp16:** ~14GB

Memory Usage

- Model memory
- Activation memory
- Gradient memory
- Optimizer memory

Memory Usage

- Model memory
- Activation memory
- Gradient memory
- Optimizer memory



Source: https://tinkerd.net/blog/machine-learning/distributed-training/

main lit-gpt / finetune / lora.py

Code Blame 335 lines (278 loc) · 12.9 KB

```
31  eval_interval = 100
32  save_interval = 100
33  eval_iters = 100
34  eval_max_new_tokens = 100
35  log_interval = 1
36  devices = 1
37
38  # Hyperparameters
39  learning_rate = 3e-4
40  batch_size = 128
41  micro_batch_size = 4
42  gradient_accumulation_iters = batch_size // micro_batch_size
43  assert gradient_accumulation_iters > 0
44  max_iters = 50000  # train dataset size
45  weight_decay = 0.01
46  lora_r = 8
47  lora_alpha = 16
48  lora_dropout = 0.05
49  lora_query = True
50  lora_key = False
51  lora_value = True
52  lora_projection = False
53  lora_mlp = False
54  lora_head = False
55  warmup_steps = 100
```

Distributed finetuning

Control Hyperparameters

Avoid OOM

- **Reduce the micro batch size**

main lit-gpt / finetune / lora.py

Code Blame 335 lines (278 loc) · 12.9 KB

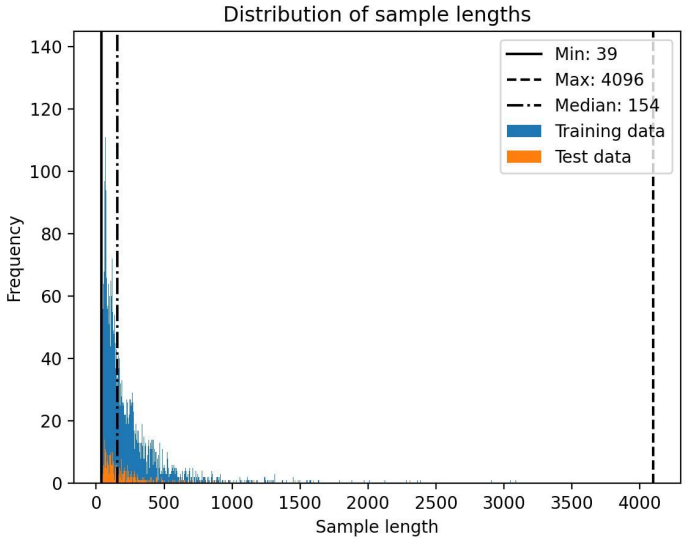
```
31  eval_interval = 100
32  save_interval = 100
33  eval_iters = 100
34  eval_max_new_tokens = 100
35  log_interval = 1
36  devices = 1
37
38  # Hyperparameters
39  learning_rate = 3e-4
40  batch_size = 128
41  micro_batch_size = 4
42  gradient_accumulation_iters = batch_size // micro_ba
43  assert gradient_accumulation_iters > 0
44  max_iters = 50000  # train dataset size
45  weight_decay = 0.01
46  lora_r = 8
47  lora_alpha = 16
48  lora_dropout = 0.05
49  lora_query = True
50  lora_key = False
51  lora_value = True
52  lora_projection = False
53  lora_mlp = False
54  lora_head = False
55  warmup_steps = 100
```

Avoid OOM

- Reduce the micro batch size
- **Reduce the model's context length**

```
python scripts/prepare_csv.py \
    --csv_path databricks-dolly-15k.csv \
    --checkpoint_dir /data/aniket/Llama-2-7b-hf \
    --destination_path data/dolly \
    --max_seq_length 512
```

Avoid OOM



- Reduce the micro batch size
- **Reduce the model's context length**

```
python scripts/prepare_csv.py \
    --csv_path databricks-dolly-15k.csv \
    --checkpoint_dir /data/aniket/Llama-2-7b-hf \
    --destination_path data/dolly \
    --max_seq_length 512
```

Avoid OOM

- Reduce the micro batch size
- Reduce the model's context length
- **Use lower precision**

```
python finetune/lora.py \
    --checkpoint_dir /data/aniket/Llama-2-7b-hf \
    --data_dir "data/dolly" \
    --out_dir "out/lora/dolly" \
    --precision bf16-true \
    --quantize bnb.fp4
```

Avoid OOM

- Reduce the micro batch size
- Reduce the model's context length
- Use lower precision
- **4-bit quantization**

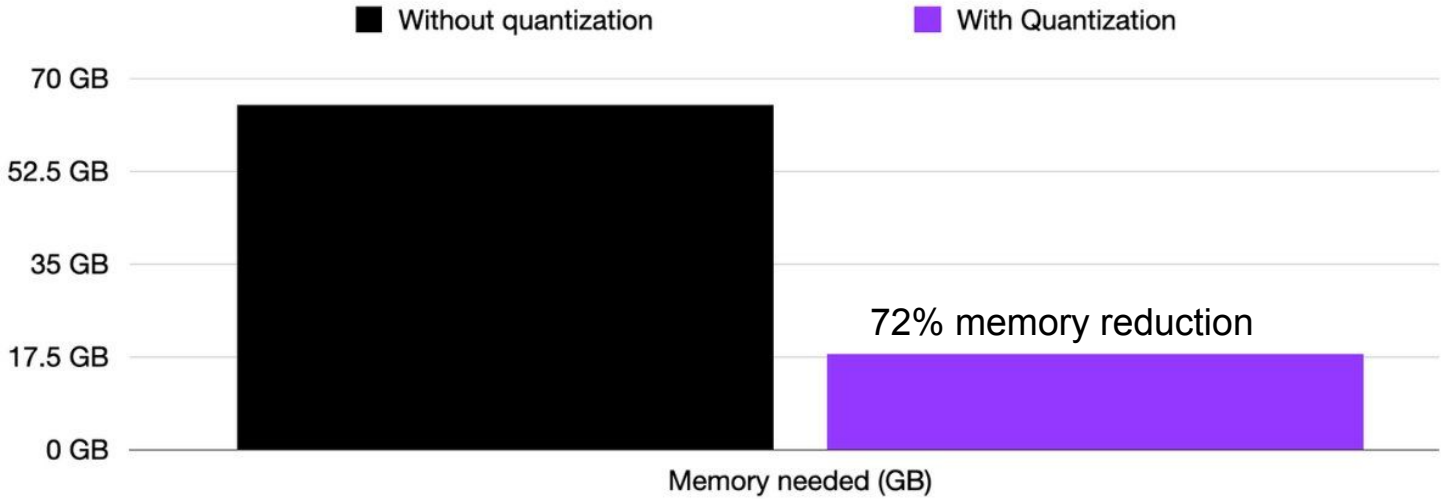
```
python finetune/lora.py \
    --checkpoint_dir /data/aniket/Llama-2-7b-hf \
    --data_dir "data/dolly" \
    --out_dir "out/lora/dolly" \
    --precision bf16-true \
    --quantize bnb.fp4
```

Without Quantization

```
python generate/base.py \
    --checkpoint_dir checkpoints/lmsys/vicuna-33b-v1.3 \
    --precision bf16-true
```

With Quantization

```
python generate/base.py \
    --checkpoint_dir checkpoints/lmsys/vicuna-33b-v1.3 \
    --precision bf16-true --quantize bnb.nf4-dq
```



Avoid OOM

- Reduce the micro batch size
- Reduce the model's context length
- Use lower precision
- 4-bit quantization
- **Do sharding across multiple GPUs**

main lit-gpt / finetune / lora.py

Code Blame 335 lines (278 loc) · 12.9 KB

```
31  eval_interval = 100
32  save_interval = 100
33  eval_iters = 100
34  eval_max_new_tokens = 100
35  log_interval = 1
36  devices = 1
37
38  # Hyperparameters
39  learning_rate = 3e-4
40  batch_size = 128
41  micro_batch_size = 4
42  gradient_accumulation_iters = batch_size // micro_ba
43  assert gradient_accumulation_iters > 0
44  max_iters = 50000  # train dataset size
45  weight_decay = 0.01
46  lora_r = 8
47  lora_alpha = 16
48  lora_dropout = 0.05
49  lora_query = True
50  lora_key = False
51  lora_value = True
52  lora_projection = False
53  lora_mlp = False
54  lora_head = False
55  warmup_steps = 100
```

Avoid OOM

main ▾ lit-gpt / finetune / lora.py

Code | Blame | 335 lines (278 loc) · 12.9 KB

```
60    def setup(
83            strategy = FSDPStrategy(
84                auto_wrap_policy={Block},
85                activation_checkpointing_policy={Block},
86                state_dict_type="full",
87                limit_all_gathers=True,
88                cpu_offload=False,
89            )
```

Avoid OOM

main lit-gpt / finetune / lora.py

Code Blame 335 lines (278 loc) · 12.9 KB

```
60    def setup(
83            strategy = FSDPStrategy(
84                auto_wrap_policy={Block},
85                activation_checkpointing_policy={Block},
86                state_dict_type="full",
87                limit_all_gathers=True,
88                cpu_offload=False,
89            )
```

Bonus: Evaluate LLMs

```
!python scripts/merge_lora.py \
    --checkpoint_dir "/data/aniket/Llama-2-7b-hf" \
    --lora_path "out/dolly/Llama-2-7b-hf/lit_model_lora_finetuned.pth" \
    --out_dir "out/dolly/Llama-2-7b-hf/"


!python eval/lm_eval_harness.py \
    --checkpoint_dir "/data/aniket/Llama-2-7b-hf" \
    --eval_tasks "[truthfulqa_mc]" \
    --precision "bf16-true" \
    --batch_size 4 \
    --save_filepath "results.json"
```

Aniket Maurya

We're hiring!

Lightning AI

Creators of PyTorch Lightning