Lessons Learned: Finetuning a Large Language Model

Noel Victor
Application Architect at Wharton Research Data Services

Slides at https://github.com/noeldvictor/lightning_talk

Who am I

I'm an Application Architect at Wharton Research Data Services (WRDS).

WRDS has supported 500+ institutions with over 600 financial and business related datasets.

If your university has WRDS, We are launching a small Beta for our Text Analytics Platform and you can contact us for more info on https://wrds-www.wharton.upenn.edu/

Caveats

This is a high level talk based on my research and experiences.

- I’m not a data scientist or AI researcher, my interest here is more experimental and practical view.
- This is a fairly high level hopefully intro friendly lightning talk.
- I’m not going to go deep into technical details.

The field itself is evolving fast and furiously, so expect some pain doing this work.

So keeping all of that in mind and the fact new features are coming out every week, you *might* be better off just waiting a week or two to see if someone else solves your issues with an app or library.

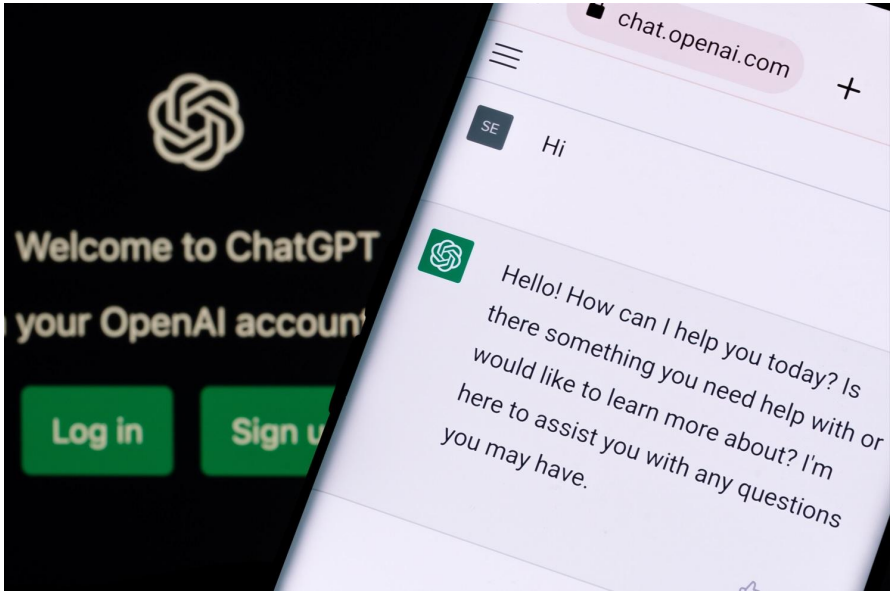
ChatGPT by OpenAI

It's a closed source large language model by OpenAI that was trained on 570 gbs of data.

- reddit
- wikipedia
- common crawl (https://commoncrawl.org/ open repo of web crawl data)
- various books
- and alot more

Important to know because if someone on reddit lied, ChatGPT won't know it was false info!

GPT-3 training cost today would be 4.6 million.



Llama by Meta

It's a open source large language model released by Meta.

- https://ai.facebook.com/blog/large-language-model-llama-meta-ai/
- LLaMA available at several sizes (7B, 13B, 33B, and 65B parameters)
    - which means different levels of quality and hardware requirements
- This is what kickstarted the open source communities interest in language models.
- Trained on similar data to GPT3/4
- Probably costs $2-3 million to train today

Language Models

Large Language models have some common key features you should know about for this talk.

- At their core really just a probability distribution of tokens usually using a technology called neural networks
- parameters
  - aspects of statistical model such weights and biases (imagine an equation with billions of variables)
- vocabulary
  - the distinct set of tokens, they can recognize and generate
  - GPT-3 50,257 vocabulary size
  - Llama 32,000 vocabulary size
- context length
  - The context length refers to the number of tokens (words, punctuation, or special characters) that the model can consider at a time when generating a response.
  - think it of like short term memory
- hyperparameters - various parameter used for training only

What is fine tuning ?

It’s when we taking a existing model, and train it a bit more to do something specific or teach it new things.

Fine tuning Why ?

You might want to fine tune a large language model (there are many!) for different reasons

- specialize it for your industry
- improve relevance
- reduce bias
- new information
- personalize it
- maybe you want it to talk like homer simpson
    - https://replicate.com/blog/fine-tune-llama-to-speak-like-homer-simpson

Training a model from scratch is millions of dollars, but fine-tuning is radically cheaper.

Stanford Alpaca - The most Famous Fine-tuned version of llama

https://github.com/tatsu-lab/stanford_alpaca

- They trained the 7B Llama model on 52,000 instruction examples (see right)
- Using <$600 to fine tune.
- Performs as well as GPT-3 (davinchi-003)
- required 112 gb VRAM

$600 is better than millions, but still not cheap enough for a lot of people.

Training Data Example:


```
  {
    "instruction": "Give three tips for staying healthy.",
    "input": "",
    "output": "1.Eat a balanced diet and make sure to include plenty of fruits and vegetables. \n2. Exercise regularly to keep your body active and strong. \n3. Get enough sleep and maintain a consistent sleep schedule."
  },
```

Better Fine tuning with PEFT and LoRA

New Techniques called State-of-the-art Parameter-Efficient Fine-Tuning (PEFT) methods came out such as LoRA - Low-Rank Adaptation

- https://github.com/huggingface/peft
- https://github.com/tloen/alpaca-lora
- https://github.com/lxe/simple-llm-finetuner

Using the alpaca-lora library, you can train the 7b parameter Llama model with 24gb vram in 8-11 hours.

- gaming card 3090/4090rtx accessible

My experiments on Google Colab were about $10-20 per run.

README.md

Training (`finetune.py`)

This file contains a straightforward application of PEFT to the LLaMA model, as well as some code related to prompt construction and tokenization. PRs adapting this code to support larger models are always welcome.

Example usage:

```
python finetune.py \
    --base_model 'decapoda-research/llama-7b-hf' \
    --data_path 'yahma/alpaca-cleaned' \
    --output_dir './lora-alpaca'
```

We can also tweak our hyperparameters:

```
python finetune.py \
    --base_model 'decapoda-research/llama-7b-hf' \
    --data_path 'yahma/alpaca-cleaned' \
    --output_dir './lora-alpaca' \
    --batch_size 128 \
    --micro_batch_size 4 \
    --num_epochs 3 \
    --learning_rate 1e-4 \
    --cutoff_len 512 \
    --val_set_size 2000 \
    --lora_r 8 \
    --lora_alpha 16 \
    --lora_dropout 0.05 \
    --lora_target_modules '[q_proj,v_proj]' \
    --train_on_inputs \
    --group_by_length
```

Inference (`generate.py`)

This file reads the foundation model from the Hugging Face model hub and the LoRA weights from `tloen/alpaca-lora-7b`, and runs a Gradio interface for inference on a specified input. Users should treat this as example code for the use of the model, and modify it as needed.

Example usage:

Ok What are we going to fine tune/train ?

- Well you have options
- Open AI
  - https://platform.openai.com/docs/guides/fine-tuning
  - as far as the docs go, you can't fine tune the chat models just yet
  - you can fine tune the older models though
- As for open source models, you have a huge amount to choose from
  - https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

🤗 Open LLM Leaderboard

📐 The 🤗 Open LLM Leaderboard aims to track, rank and evaluate LLMs and chatbots as they are released.

🤗 Anyone from the community can submit a model for automated evaluation on the 🤗 GPU cluster, as long as it is a 🤗 Transformers model with weights on the Hub. We also support evaluation of models with delta-weights for non-commercial licensed models, such as LLaMa.

🔍 Search your model and press ENTER...

🏅 LLM Benchmark (lite) | 📊 Extended view | About

Model	Average ⬆	ARC (25-s) ⬆	HellaSwag (10-s) ⬆	MMLU (5-s) ⬆	TruthfulQA (MC) (0-s) ⬆
tiiuae/falcon-40b-instruct	63.2	61.6	84.4	54.1	52.5
timdettmers/guanaco-65b-merged	62.2	60.2	84.6	52.7	51.3
CalderaAI/30B-Lazarus	60.7	57.6	81.7	45.2	58.3
tiiuae/falcon-40b	60.4	61.9	85.3	52.7	41.7
timdettmers/guanaco-33b-merged	60	58.2	83.5	48.5	50
ausboss/llama-30b-supercot	59.8	58.5	82.9	44.3	53.6
huggyllama/llama-65b	58.3	57.8	84.2	48.8	42.3
pinkmanlove/llama-65b-hf	58.3	57.8	84.2	48.8	42.3
llama-65b	58.3	57.8	84.2	48.8	42.3
MetaIX/GPT4-X-Alpasta-30b	57.9	56.7	81.4	43.6	49.7
Aeala/VicUnlocked-alpaca-30b	57.6	55	80.8	44	50.4
digitous/Alpacino30b	57.4	57.1	82.6	46.1	43.8
Aeala/GPT4-x-AlpacaDente2-30b	57.2	56.1	79.8	44	49.1
TheBloke/dromedary-65b-lora-HF	57	57.8	80.8	50.8	38.8
TheBloke/Wizard-Vicuna-13B-Uncensored-HF	57	53.6	79.6	42.7	52

Ok What are we going to fine tune ?

- **Most open source models aren't for commercial usage**
- If you want a commercial friendly option try
    - **RedPajama** - https://www.together.xyz/blog/redpajama

Most tutorials/ecosystems are oriented around llama and its variants, so you should start with those…

GPT Quantization ? An Important Side Note

GPTQ is a clever quantization algorithm that lightly re-optimizes the weights during quantization (reducing the amount of bits to represent weight) so that the accuracy loss is compensated.

See the paper for more details: https://arxiv.org/abs/2210.17323.

TLDR: 4-bit GPTQ models reduce VRAM usage by about 75%. So LLaMA-7B fits into a 6GB GPU, and LLaMA-30B fits into a 24GB GPU.

llama.cpp - can run models and quantize models

https://github.com/ggerganov/llama.cpp is very cool software

- Allows you run llama models (not every model) on CPU though may not be fast.
- Allows you quantize models too.
- Works even on mac, linux, windows, and android.
- I highly recommend running it via docker if you don’t need to squeeze performance from your device.

Georgi Gerganov
@ggerganov

Here is what a properly built llama.cpp looks like

Running 7B on 2 years old Pixel 5 at 1 token/sec.
Would be interesting to see how an interactive session feels like

> **Radoslav Gerganov** @rgerganov · Mar 14
> Running llama.cpp on my Pixel5 phone with termux. Kudos to @ggerganov !

```
one
llama_model_load: model size =  4017.27 MB / num tensors
 = 291

system_info: n_threads = 6 / 8 | AVX = 0 | AVX2 = 0 | AV
X512 = 0 | FMA = 0 | NEON = 1 | ARM_FMA = 1 | F16C = 0 |
 FP16_VA = 0 | WASM_SIMD = 0 | BLAS = 0 | SSE3 = 0 | VSX
 = 0 |

main: prompt: 'I believe the meaning of life is'
```

https://twitter.com/ggerganov/status/1635605532726681600?lang=en

So my basic workflow

Data Gathering and Cleaning

- get raw data and keep it safe somewhere
- process raw data in a way your chosen model can handle
    - the way you process the data into question answer, context sizes ,etc depends on the model

Model Setup

- Use a LoRA type technique to train a llama based model
    - On my Gaming PC at home or using Google Colab Pro typically using 10-20usd per training run for 7b/13b parameter run.
- Merge LoRA weights back into parent model
- Convert the model into a GPTQ 4 bit variant
    - This can take alot of memory as well, example 30B 4bit quantize uses about 20-22GB of VRAM on GPU

Run Model

- I can run this processed model on an old laptop using cpu and ram (not video ram) and see what I get.
- I might run a series of tasks overnight just see what I get using the llama.cpp python bindings, so I'm not too time sensitive.

Retrain Model

- You should expect to retrain model
- **One of the most critical things to remember is that you might have to run various trials and see what you get**

My Results - well it depends

I've trained several different types of texts now with different levels of success.

- smaller complete units of knowledge or text work really well like the comments or answers you get on reddit or stack overflow
- extremely long form content got very poor results for example knowledge base/support articles from WRDS.
- teaching something novel/niche was kinda easy like pretending to be homer simpson.
- STRANGE AND UNEXPECTED:
  - trying to UPDATE data on something well known
  - for example: information on WRDS's products was difficult, it was clear I'm seeing strange conflicting interactions from internet trained data and the data I trained in.
  - **This is where I need to do more testing around hyperparameters, full fine-tuning, or/and try to force overfitting.**

How you view the results is really dependent on your goals, and is highly subjective.

- **You will have to figure out how to score what you think are good/bad answers especially for business**
- **If you are looking to have a laugh with Homer Simpsons or doing "creative" work with human oversight, you might not care so much.**

The Ever Changing Future

- New approaches to “teaching ai”, such as using increased context windows to pass information instead of fine tuning
  - GPT4 has increased context sizes of roughly 8000 to 32,000 tokens (24,000 words/ 48 pages of text)
  - AntropicAI has 100k token context windows
  - Superhot Models - 8k context for popular opensource models
    - https://www.reddit.com/r/LocalLLaMA/comments/14kj2w8/thebloke_has_released_superhot_versions_of/

The Ever Changing Future 2

One click installable software has come out that claims to be able to integrate with your data very easily.

- GPT4all (not affiliated with OpenAI)
  - https://docs.gpt4all.io/gpt4all_chat.html
  - works with several open source models
  - **has local document plugins that allows you to chat with your local files**
- https://github.com/oobabooga/text-generation-webui
  - allows you do lora training via Gradio based GUI
  - works with several open source models
  - **requires GPU**

The Ever Changing Future 3

So things are evolving at breakneck speed. New services and AI are evolving extremely fast.

I can't say I would recommend spending time trying to code your own ai training workflows in python or even looking to hard at solutions that aren't simple 1-3 step setups for training and running models.

The biggest bang for your buck is probably spending time curating your data, and thinking about what you want from it.

Thank you!

LoRA ?

So okay we aren't technically fine-tuning the entire model anymore, we are isolating and updating the min amount of weights to alter into a second set of weights. Typically people merge these weights back in to run their models locally.

You can read more here :https://github.com/microsoft/LoRA

- it gets mathy fast.