Finetuning LLM

|  |  |  |
| --- | --- | --- |
|  | **Prompting** | **Finetuning** |
| Pros | No data to get started | Nearly unlimited data fits |
| Smaller upfront cost | Learn new information |
| No technical knowledge needed | Correct incorrect information |
| Connect data through retieval (RAG) | Less cost afterwards if smaller model |
|  | Use RAG too |
| Cons | Much less data fits | More high quality data |
| Forgets data | Upfront compute cost |
| Hallucinations | Needs some technical knowledge, esp data |
| RAG misses, or gets incorrect data |  |
| Use case | Generic,side projects,prototypes | Domain-specific,enterprise,production usage, -privacy |

|  |  |
| --- | --- |
| Benefits of finetuining your own LLM |  |
|  |  |
| **Performance** | stop hallucination |
| increase consistancy |
| reduce unwanted info |
| **Privacy** | on-prem or VPC |
| prevent leakage |
| no breaches |
| **Cost** | lower cost per request |
| increased transperancy |
| greater control |
| **Reliability** | control uptime |
| lower latency |
| moderation |

|  |  |
| --- | --- |
| Techonologies to finetune |  |
| Pythorch(Meta) | basic |
| Huggingface | intermediate |
| Llama library (Lamini) | high level |

Finetuning fits in after a step called Pretraining

Pretraining

* Model at the start:
  + Zero knowledge about the world
  + Can’t form English words
* Can’t predict next token
* Trained on giant corpus of text data
* Often scraped from internet: “unlabeled
* Self-supervised learning
* After training
  + Learns language
  + Learns knowledge

What is “data scraped from the internet”?

* Often not publicized how to pretrain
* Open-source pretraining data : “The Pile” (by ElutherAI)
* Expensive and time-consuming to train

LLMs are base models pretrained on data from internet as explained above

The pretarined LLMs are not used to chatbot interface.

Eg. In\p What is the the capital of Mexico?

LLM o\p what is the capital of Hungary?

**Finetuning after pretraining**

To get it to chatbot interface finetuning is required.

We can do further finetuning on finetuned models

* Finetuning usually refers to training further
  + Can also be self-supervised unlabeled data
  + Can be “labeled” data you curated
  + Much less data needed
  + Tool in your toolbox
* Finetuning for generative tasks is not well-defined:
  + Unlike other finetuning, it updates entire model, not just part of it
  + Same training objective: next token prediction
  + More advance ways to reduce how much to update model

What is finetuning doing for you?

* Behavior change
  + Learning to respond more consistently
  + Learning to focus, e.g. moderation
  + Teasing out capability, eg. Better at conversation
* Gain knowledge
  + Increasing knowledge of new specific concepts
  + Correcting old incorrect information
* Both

**Tasks to finetune**

* For LLM just text-in, text-out
  + Extraction: text in, less text out
    - Reading
    - Keywords, topics, routing,agents (planning,reasoning,self-citric,tool use),etc
  + Expansion: text in , more text out
    - Writing
    - Chat, write emails, write code
* Task clarity is key indicator of success
* Clarity means knowing what’s bad vs good vs better

**First time finetuning**

1. Identify task(s) by prompt-engineering a large LLM
2. Find tasks that you see an LLM doing ~OK at
3. Pick one task
4. Get ~1000 inputs and outputs for the task
5. Finetune a small LLM on this data

Instruction Finetuning to make gpt3 to chat\_gpt3

Types of Finetuning:-

Resoning

Routing

Copilot

Chat, agents

What is instruction finetuning

* AKA “instruction-tuned” or “instruction-following” LLMs
* Teaches model to behave more like a chatbot
* Better user interface for model interaction
  + Turned GPT-3 into ChatGPT
  + Increase AI adoption

Instruction – following datasets

* Some existing data is ready as-is
* FAQs
* Customer support conversations
* Slack messages

If no data is available:-

LLM Data Generation Non -OnA data can also be converted to QnA

Read Me - Using a prompt template

To authenticate, retrieve - Using another LLM

The API key from the

Settings page

LLM - ChatGPT (Alpaca)

Generation - Open-source models

Pipeline

How do you authenticate your request?

You must retrieve the API key

From the settings page

Instruction Finetuning Generalization

* Can access model’s pre-existing knowledge
* Generalize following instructions to other data, not in finetuning dataset

What kind of data?

Better:- Lower: -

Higher Quality Lower Quality

Diversity Homogeneity

Real Generated

More Less

Steps to prepare your data

* Collect instruction-response pairs
* Concatenate pairs (add prompt template, if applicable)
* Tokenize
* Split into train/test

Tokenizing your data

* Tokenize the data
* Converts text into tokens and token to number (Encoder)
* Converts numbers to tokens (Decoder)

There are multiple popular tokenizers:

* Use the tokenizer associated with your model

Training: same as other NN

What’s going on?

Add training data

Once upon a Calculate loss

Midnight Backprop through model

Dreary while I Update weights

pondered

Hyperparameters

LLM Learning rate

Learning rate scheduler

Update

weight

Sd!!@ upon Optimizer hyperparameters

Loss

Run through general chunks of training process in code

For epoch in range(num\_epochs)

For batch in train\_dataloader:

output = model(\*\*batch)

loss = output.loss

loss.backward()

optimizer.step()

Evaluating generative models is notoriously difficult

* Human expert evaluation is most reliable
* Good test data is crucial
  + High – quality
  + Accurate
  + Generalized
  + Not seen in training data
* Elo comparisons also popular