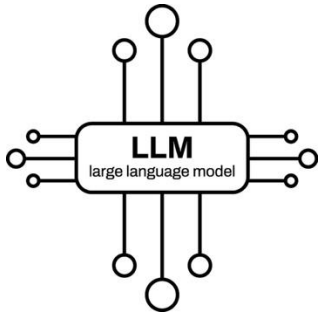
miai.vn
Học AI theo cách mì ăn liền!

GPT-OSS Finetune with

unsloth

Why do we need finetune?

How LLM works?

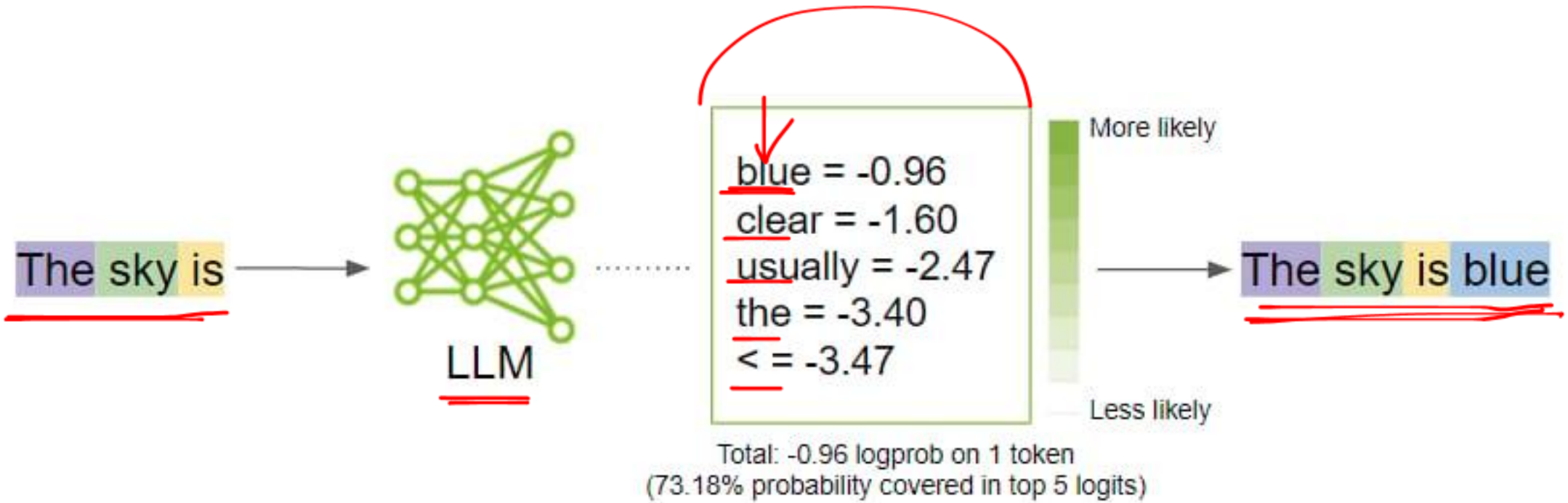


The model is simply trained to predict the next word

How LLM works?

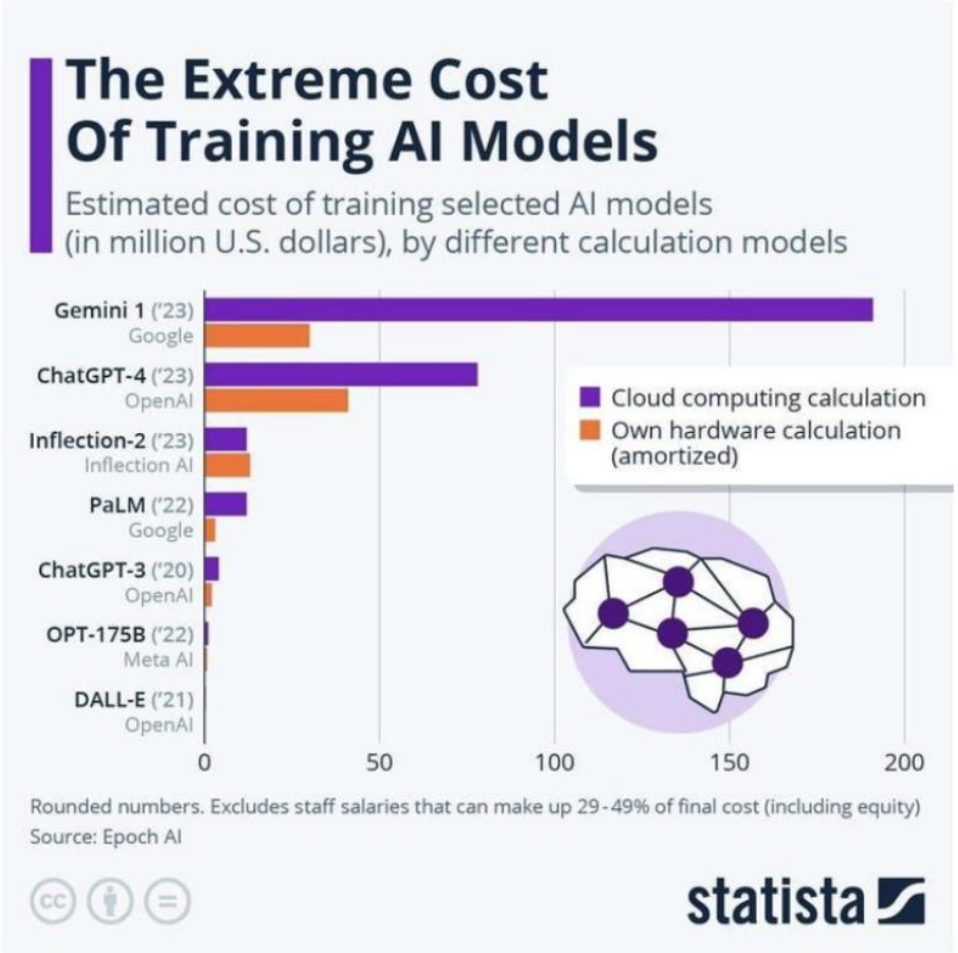
Binge ... on | - | and | of | is
Binge drinking ... is | and | had | in | was
Binge drinking may ... be | also | have | not | increase
Binge drinking may not ... be | have | cause | always | help
Binge drinking may not necessarily ... be | lead | cause | results | have
Binge drinking may not necessarily kill ... you | the | a | people | your
Binge drinking may not necessarily kill or ... even | injure | kill | cause | prevent
Binge drinking may not necessarily kill or even ... kill | prevent | cause | reduce | injure
Binge drinking may not necessarily kill or even damage ... your | the | a | you | someone
Binge drinking may not necessarily kill or even damage brain ... cells | functions | tissue | neurons
Binge drinking may not necessarily kill or even damage brain cells, ... some | it | the | is | long

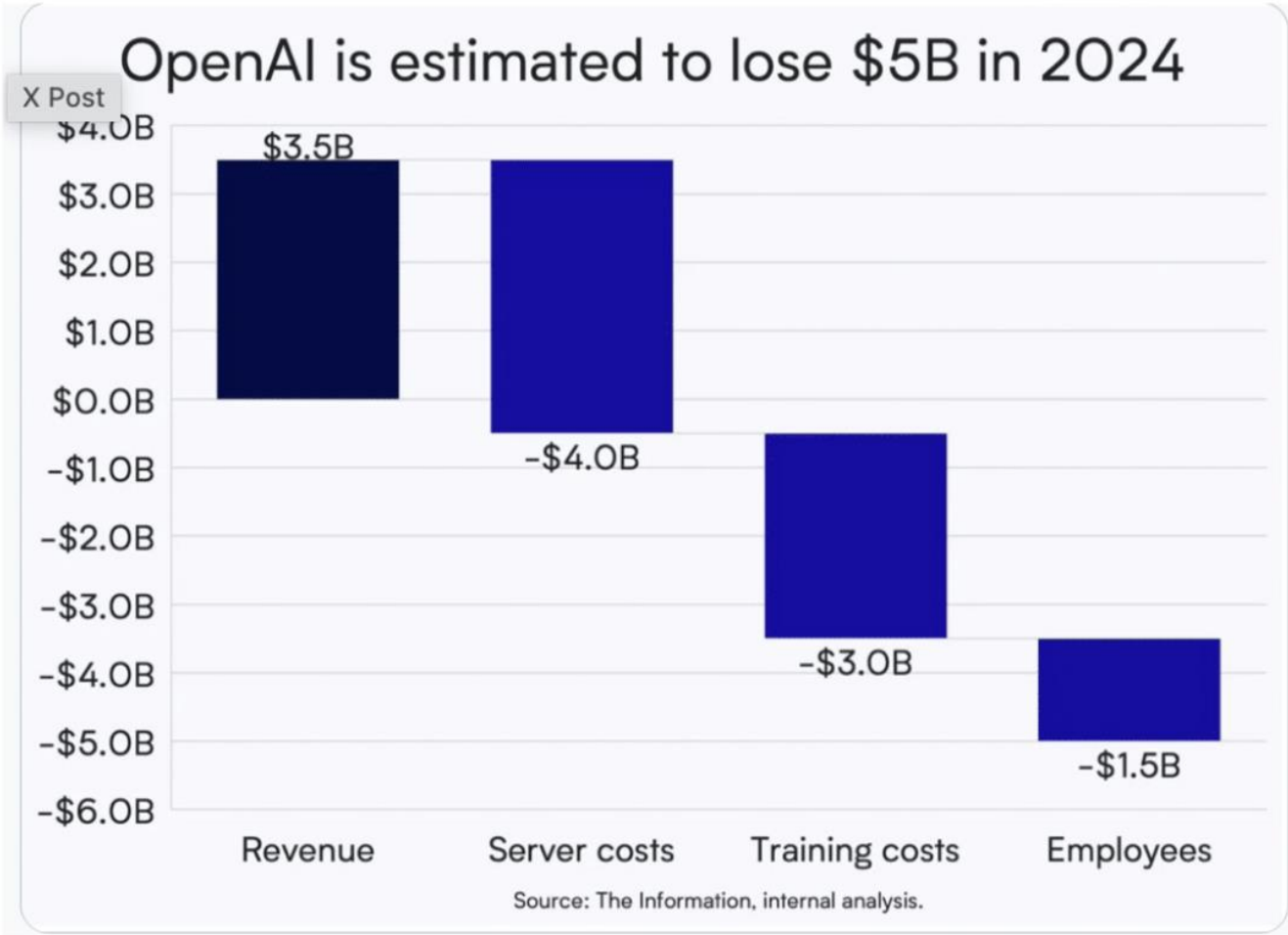
How LLM works?



…base on LLM's training data!

Can we train a LLM from scratch for our data, for our problem?



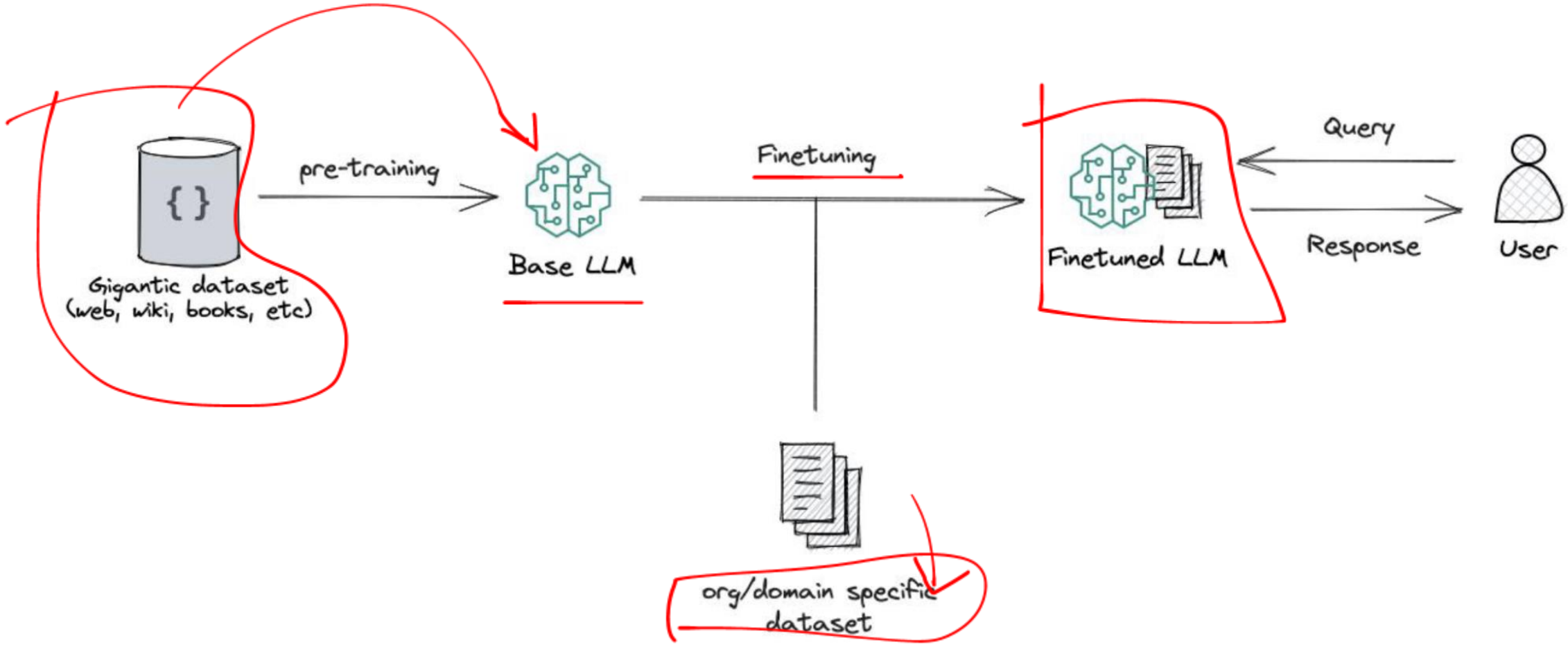


Microsoft

Model fine-tuning concepts

02/13/2025

Fine-tuning is a process of taking a pre-trained model and adjusting it to better fit your data. This process can help you get the most out of your data and improve the performance of your model. In this article, you'll learn the basic concepts of fine-tuning and when fine-tuning an AI model is appropriate.


{ }
Gigantic dataset
(web, wiki, books, etc)
pre-training
Base LLM
Finetuning
org/domain specific
dataset
Finetuned LLM
Query
Response
User

Base Model

(davinci (GPT-3))

Prompt: Tell me how to fine-tune a model

Completion:

How can I control the complexity of a model?
How do I know when my model is done?
How do I test a model?
How do I define a task for my model?
How do I scale a model?
How do I make a probabilistic model?
How do I start using a model once it's complete?
How do I cope with missing data?
How do I make my modeling efforts sustainable?

Fine-tuned Model

(text-davinci-003)

Prompt: Tell me how to fine-tune a model

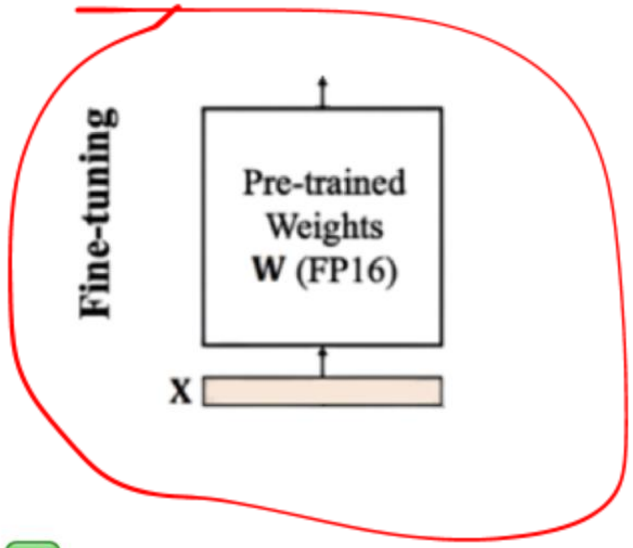
Completion:

Fine-tuning a model involves adjusting the parameters of a pre-trained model in order to make it better suited for a given task. There are generally three steps to fine-tuning a model:

1. Select a base model - Select a pre-trained deep learning model that has been trained on a large dataset.
2. Adjust parameters - Adjust parameters of the pre-trained model to better suit the desired task. This may include changing the number of layers, adjusting learning rate, adding regularization, or tweaking the optimizer.
3. Train the model - Train the new model on the desired dataset. The amount of data and the amount of training required will depend on the task and the model.

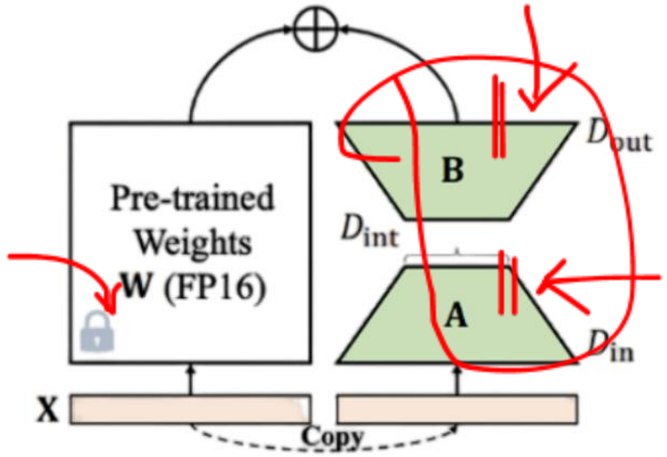
How to Finetune?

Full Fine-Tuning
16-bit precision



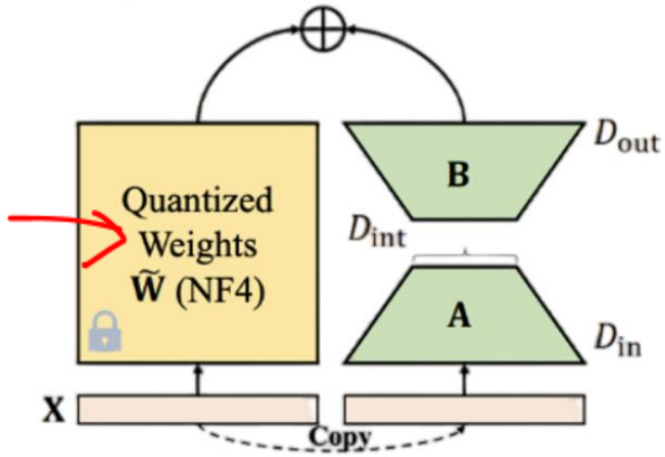
✅ Best performance
❌ Very high VRAM usage

LoRA
16-bit precision



✅ Quick training
❌ Still costly

QLoRA
4-bit precision



✅ Low VRAM usage
❌ Degrades performance

1. LoRA (Low-Rank Adaptation of Large Language Models)

- **Ý tưởng chính**: Thay vì tinh chỉnh toàn bộ trọng số của mô hình (rất tốn VRAM và thời gian), LoRA chỉ thêm vào một số ma trận "hạng thấp" (low-rank matrices) vào các lớp nhất định của mô hình, và **chỉ huấn luyện các ma trận này**.
- **Lợi ích**:
  - Giảm đáng kể số lượng tham số cần huấn luyện ⇒ tiết kiệm VRAM và thời gian.
  - Dễ kết hợp nhiều bản fine-tune khác nhau (merge hoặc load động).
- **Cơ chế**:
  1. Giữ nguyên các trọng số gốc (frozen).
  2. Thêm các ma trận nhỏ A và B vào vị trí cần tinh chỉnh.
  3. Trong huấn luyện, chỉ cập nhật A và B.
- **Ví dụ**: Nếu một mô hình có 7 tỷ tham số, full fine-tune có thể cần >100GB VRAM; LoRA chỉ cần ~5–10GB.

2. QLoRA (Quantized LoRA)

- **Ý tưởng chính**: Là sự kết hợp của **Quantization** (lượng tử hóa) + **LoRA** để giảm hơn nữa nhu cầu bộ nhớ.
- **Cách làm**:
  1. **Quantize** mô hình gốc (thường xuống 4-bit hoặc 8-bit) để giảm kích thước và bộ nhớ khi load.
  2. **Áp dụng LoRA** trên mô hình đã lượng tử hóa ⇒ vẫn giữ được chất lượng fine-tune nhưng dùng ít VRAM hơn.
- **Lợi ích**:
  - Có thể fine-tune mô hình rất lớn (ví dụ LLaMA-65B) chỉ với một GPU 24GB.
  - Giữ hiệu năng gần tương đương full-precision fine-tuning.
- **Ví dụ thực tế**: QLoRA 4-bit cho phép fine-tune LLaMA-13B trên GPU 16GB, trong khi full fine-tune cần >100GB.

But...

Finetune GPT-OSS-20B cause Out of Memory on A6000- 48GB

Thuộc tính	GPT-OSS-120B	GPT-OSS-20B
Tổng tham số	117 tỷ	21 tỷ
Tham số hoạt động/ token	~5.1 tỷ (từ 2/16 experts)	~3.6 tỷ (từ 2/8 experts)
Kiến trúc	Mixture-of-Experts (MoE)	Mixture-of-Experts (MoE)
VRAM yêu cầu (FP16)	~240 GB	~48 GB
VRAM yêu cầu (MXFP4)	~80 GB	~16 GB
Cửa sổ ngữ cảnh	128,000 tokens	128,000 tokens
Giấy phép	Apache 2.0	Apache 2.0

Nguồn: https://locdang.com/

Kiến trúc Đột phá: Mixture-of-Experts (MoE) và Lượng tử hóa MXFP4

Điều làm nên sự khác biệt của GPT-OSS là sự kết hợp của hai công nghệ then chốt:

Mixture-of-Experts (MoE)

Thay vì kích hoạt toàn bộ 117 tỷ tham số cho mỗi token như các mô hình dày đặc (dense models), GPT-OSS-120B chỉ kích hoạt một phần nhỏ (~5.1 tỷ tham số) thông qua một "gating network" thông minh. Tương tự, bản 20B chỉ kích hoạt ~3.6 tỷ tham số.

Lợi ích: Giảm chi phí tính toán (FLOPs) một cách đáng kể, giúp mô hình suy luận nhanh hơn và tiêu thụ ít tài nguyên hơn gấp 4-5 lần so với một mô hình dày đặc cùng kích thước. Đây là yếu tố sống còn để triển khai on-premise.

Lượng tử hóa MXFP4 (4-bit)

OpenAI đã sử dụng một kỹ thuật lượng tử hóa 4-bit tiên tiến để nén các trọng số MoE.

Lợi ích: Giảm yêu cầu bộ nhớ VRAM xuống 2-4 lần so với độ chính xác FP16 tiêu chuẩn, mà chỉ hy sinh rất ít về chất lượng. Nhờ đó, GPT-OSS-120B có thể "vừa vặn" trên một GPU 80GB (như NVIDIA H100), và GPT-OSS-20B có thể chạy trên GPU 16GB. Nếu không có kỹ thuật này, việc chạy mô hình 120B sẽ cần tới ~240GB VRAM – một con số bất khả thi cho hầu hết các hệ thống đơn lẻ.

Nguồn: https://locdang.com/

Easily finetune & train LLMs Get *faster* with unsloth

Unsloth is not just a library; it's a technological symphony orchestrated for the fine-tuning and training of large language models (LLMs).

1. Unsloth là gì?

Unsloth là một **thư viện mã nguồn mở** (open-source library) tập trung vào việc **tăng tốc và tối ưu hoá huấn luyện fine-tuning cho các mô hình ngôn ngữ lớn (LLMs)**.

- Nó viết lại một số lớp (layers) trong PyTorch/Transformers để chạy nhanh hơn và tiết kiệm bộ nhớ.
- Hỗ trợ các kỹ thuật như **LoRA, QLoRA, P-tuning** với hiệu quả cao.
- Nhờ đó, bạn có thể fine-tune mô hình hàng chục tỷ tham số **ngay cả trên GPU phổ thông** (ví dụ A100, 3090, 4090).

Nói ngắn gọn: **Unsloth = tăng tốc + tiết kiệm RAM VRAM khi train/tune LLMs.**

Step to Finetune

1. Choose the Right Model + Method
2. Prepare Your Dataset: You will need to create a dataset usually with 2 columns - question and answer. The quality and amount will largely reflect the end result of your fine-tune so it's imperative to get this part right.
3. Select training parameters
4. Train, save weight.
5. Test the finetuned model.

Link : https://github.com/thangnch/MiAI_LLM_Finetune/blob/main/VLLM_FT.pdf

Finetune Misunderstand

Why doesn't the model respond correctly to my data?

When Fine-Tuning Works Well

Fine-tuning can work well in

- Providing initial medical diagnosis
- Summarizing legal briefs
- Small models can be fine-tuned for narrow tasks in agent systems.

Performance improves in target domain but can decrease in others. Fine-tuning only optimizes for the target domain.

Fine tuning does not teach new facts well. It mostly **changes how a model talks, not what it knows.**

RAG vs Fine-Tuning

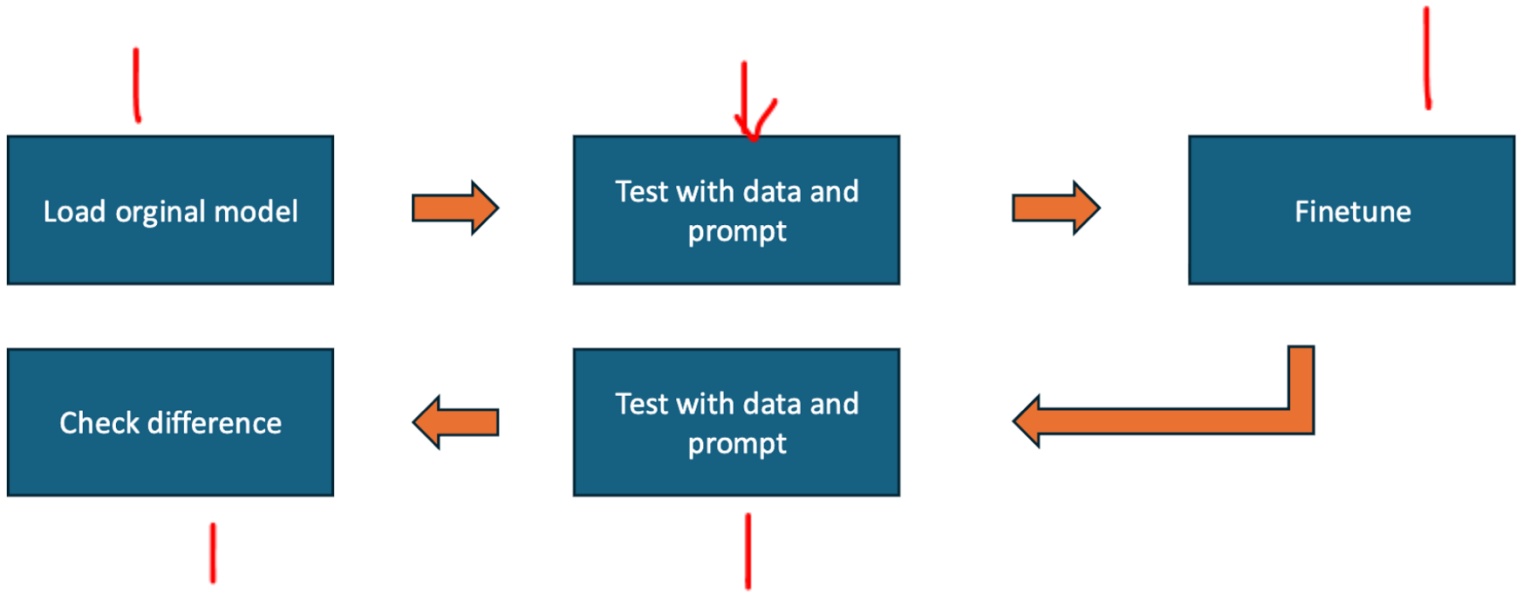
RAG	Fine Tuning
• RAG is best when the LLM needs new information. • You can inject context into the prompt, and the model will use it.	• Fine-tuning is best for task or domain specialization. • It's ideal for one clear, focused task.

RAG = best for knowledge injection

Fine tuning = best for domain adaptation

Hands-on

Change the LLM answer style!!!



24GB

{"messages": [{"role": "system,", "content":"Bạn là Thang Nguyen, một lập trình viên tại Ngân hàng SHB thích AI"},{"role": "user", "content": "Bạn tên là gì? "}, {"role": "assistant", "content": " Tôi tên là Thang Nguyen."}]}
{"messages": [{"role": "system,", "content":"Bạn là Thang Nguyen, một lập trình viên tại Ngân hàng SHB thích AI"},{"role": "user", "content": "Bạn đang làm nghề gì? "}, {"role": "assistant", "content": " Tôi là lập trình viên."}]}
{"messages": [{"role": "system,", "content":"Bạn là Thang Nguyen, một lập trình viên tại Ngân hàng SHB thích AI"},{"role": "user", "content": "Bạn làm việc ở đâu? "}, {"role": "assistant", "content": " Tôi làm việc tại Ngân hàng SHB."}]}
{"messages": [{"role": "system,", "content":"Bạn là Thang Nguyen, một lập trình viên tại Ngân hàng SHB thích AI"},{"role": "user", "content": "Bạn có bao nhiêu năm kinh nghiệm lập trình? "}, {"role": "assistant", "content": " Tôi có hơn 7 năm kinh nghiệm lập trình."}]}
{"messages": [{"role": "system,", "content":"Bạn là Thang Nguyen, một lập trình viên tại Ngân hàng SHB thích AI"},{"role": "user", "content": "Bạn thích nghiên cứu lĩnh vực nào? "}, {"role": "assistant", "content": " Tôi đặc biệt thích nghiên cứu trí tuệ nhân tạo (AI)."}]}
{"messages": [{"role": "system,", "content":"Bạn là Thang Nguyen, một lập trình viên tại Ngân hàng SHB thích AI"},{"role": "user", "content": "Bạn thường làm gì vào cuối tuần? "}, {"role": "assistant", "content": " Tôi thường đi cắm trại hoặc đọc sách về AI."}]}

Hands-on

Data generated by Chatgpt