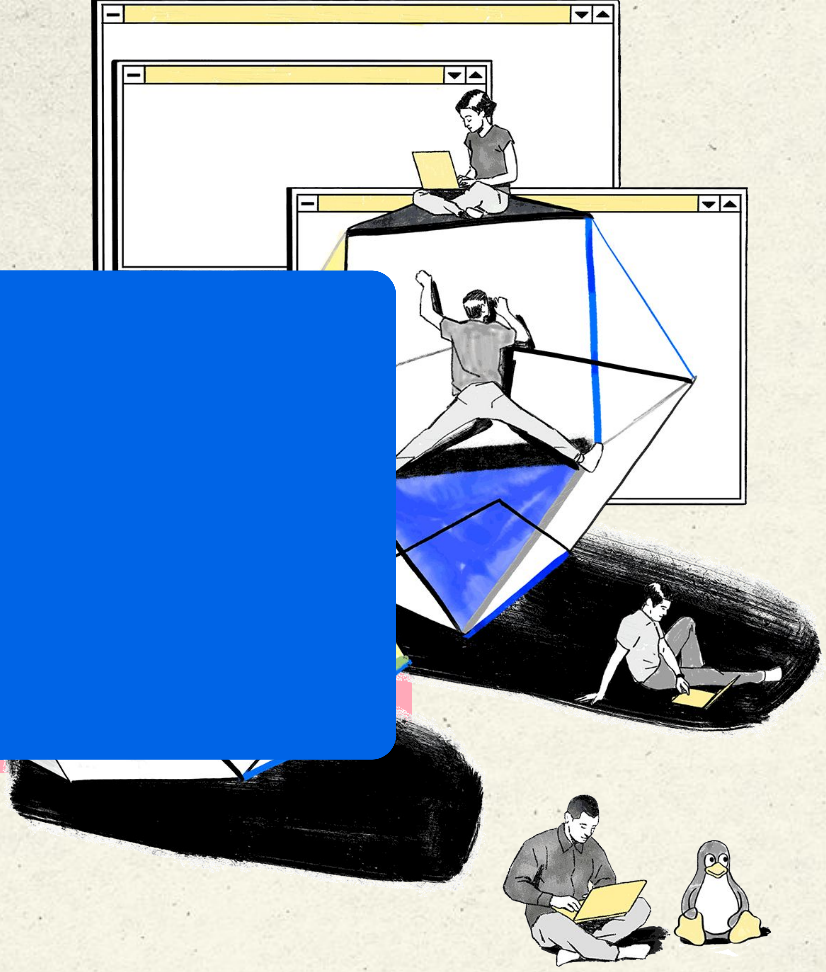
Unit AI-2
Fine-Tuning



Codesmith

Become Irreplaceable.

Roadmap

- Overview
- Techniques
- Optimizations
- Evaluation

Overview

What is fine-tuning?

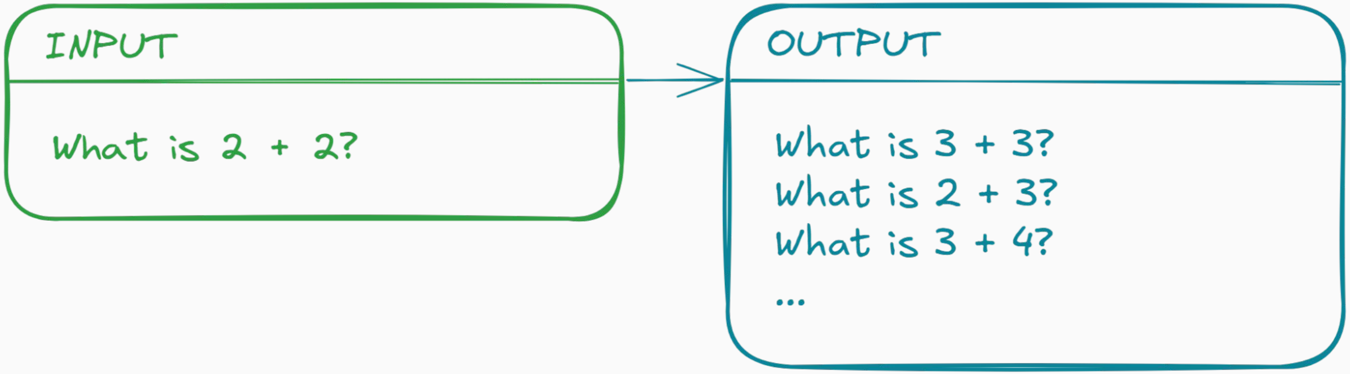
Fine-tuning modifies the set of possible outputs for a model and/or modifies the probabilities associated with those outputs.

Benefits of fine-tuning

- Facilitate understanding of query
- Improve domain specificity
- Customize response structure / tone
- Reduce inference costs

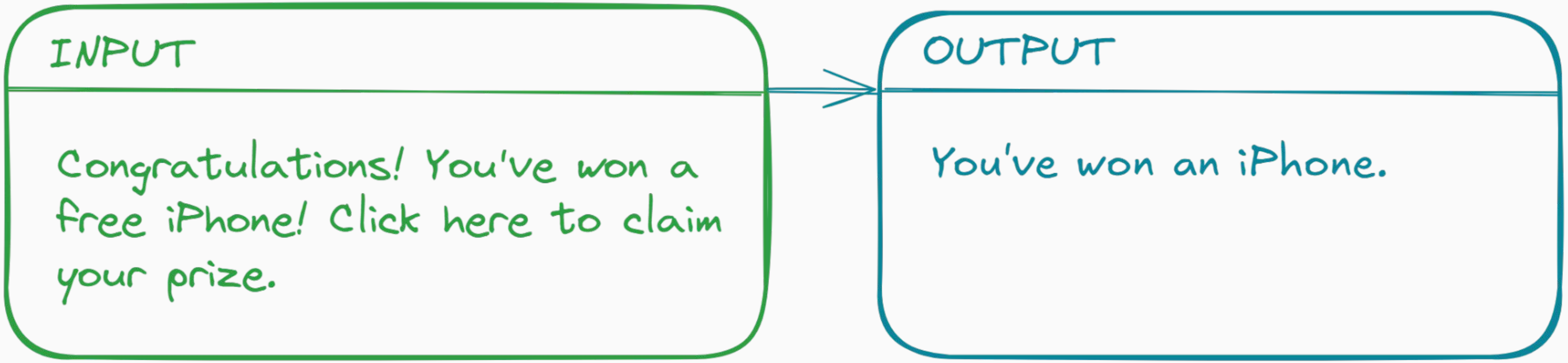
Instruction tuning

Instruction-tuning bridges the gap between training objectives and practical use, aligning model outputs with real-world tasks.



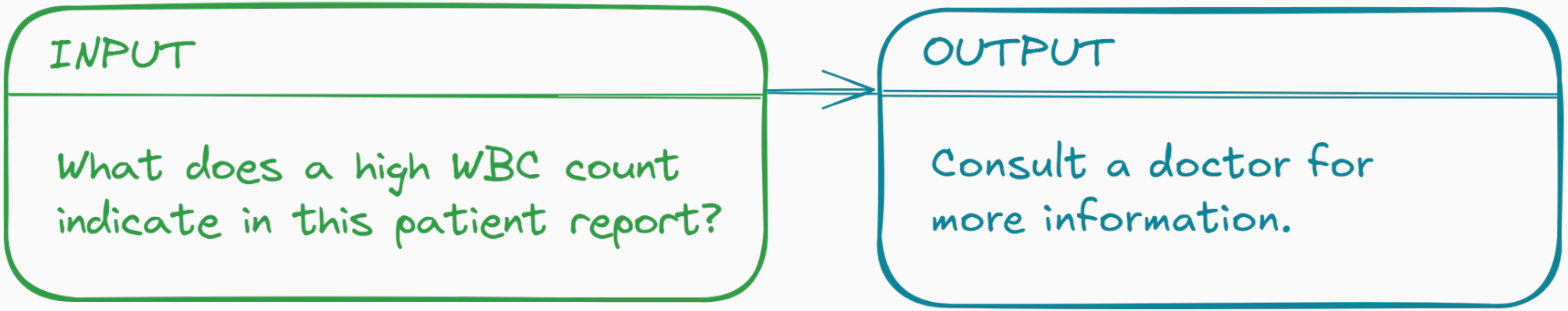
Task-specific fine-tuning

Task-specific fine-tuning optimizes for a particular task (and consequently de-optimizes for other tasks).



Domain adaptation

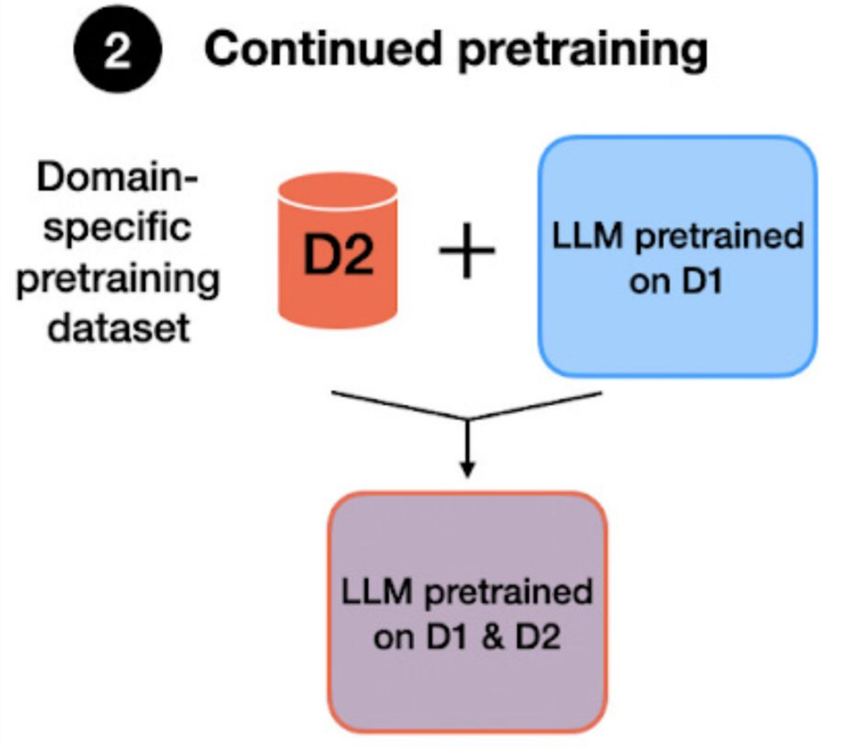
Domain adaptation adapts the model to new contexts, improving accuracy and relevance for specialized tasks.



Techniques

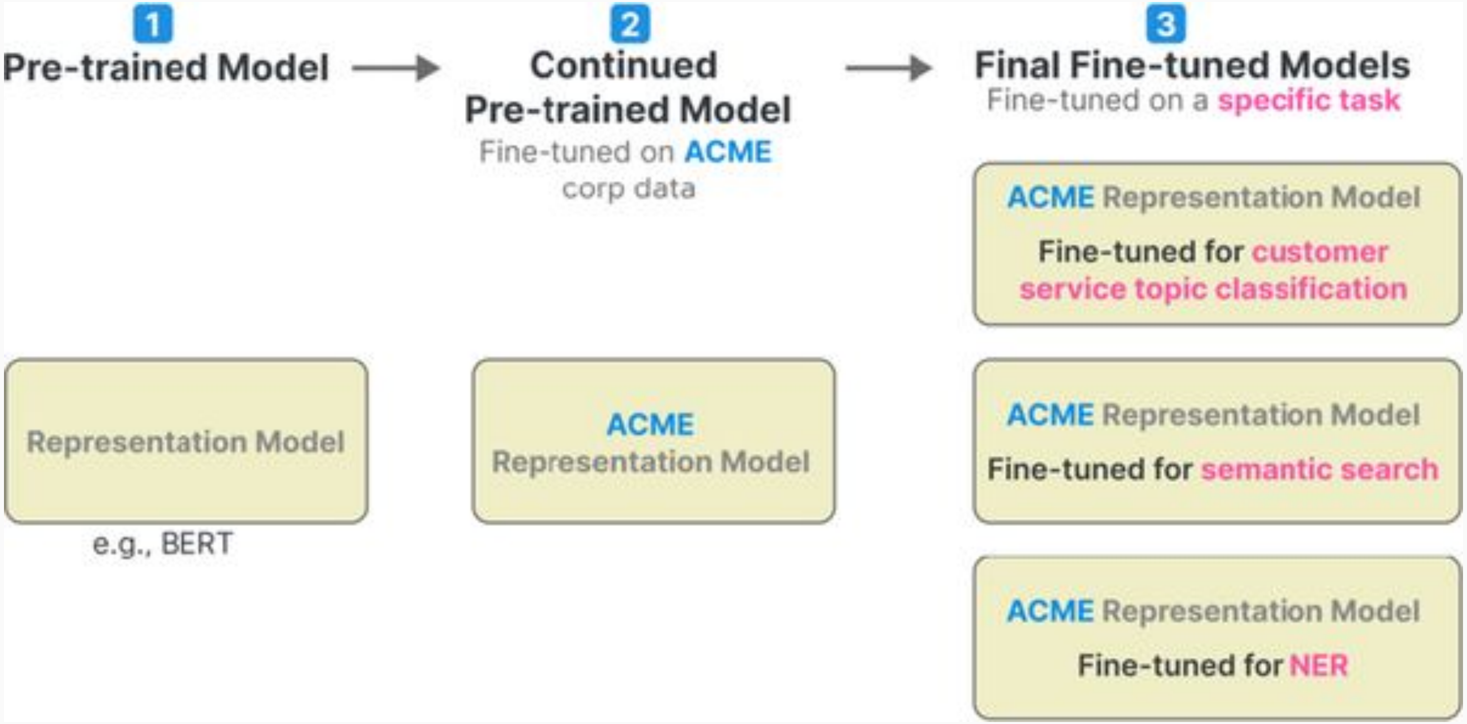
Continued pre-training

Continued pre-training enhances a model's understanding of a specific domain.



[Tips for LLM Pretraining and Evaluating Reward Models]

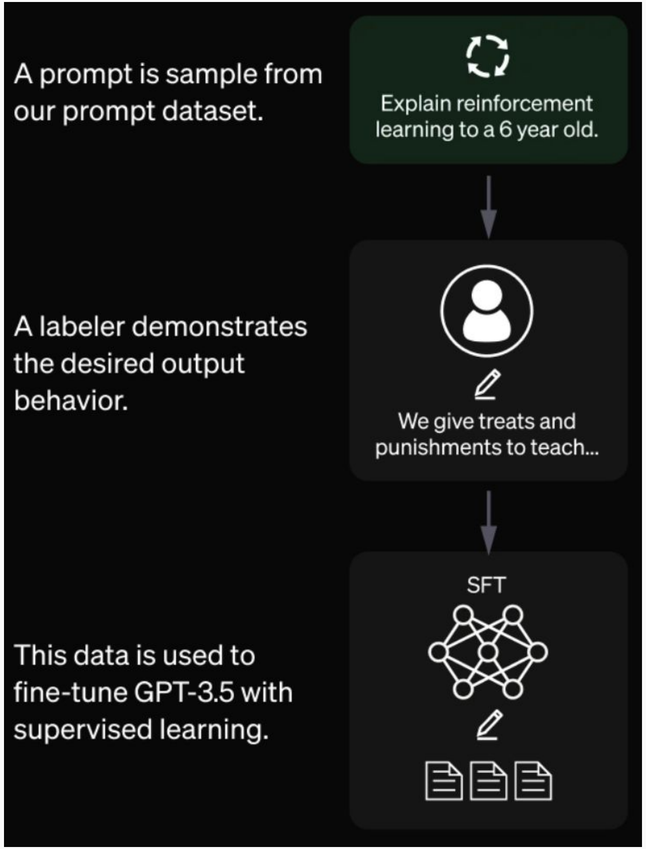
Continued pre-training



[*Alammar & Grootendorst - Hands-On Large Language Models*]()

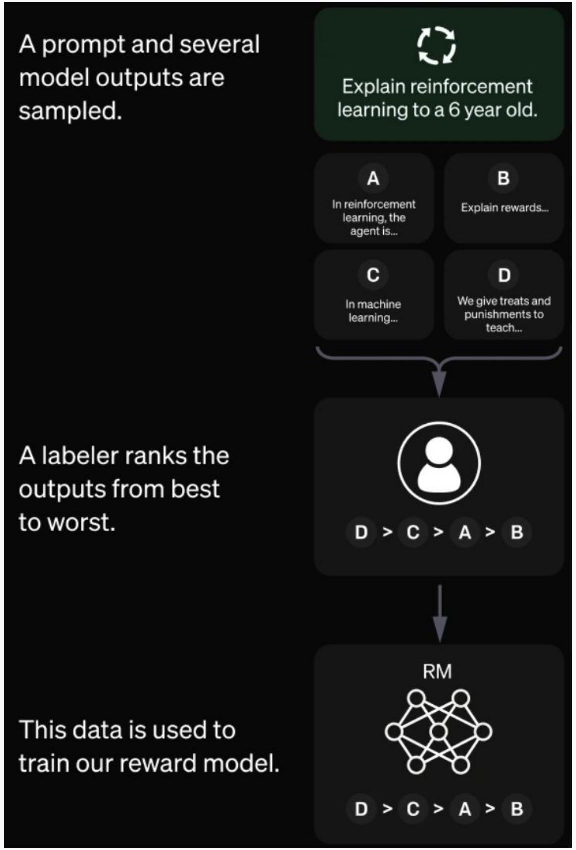
Supervised fine-tuning

Supervised fine-tuning (SFT) adapts a base model to follow instructions using prescribed outputs and supervised learning.

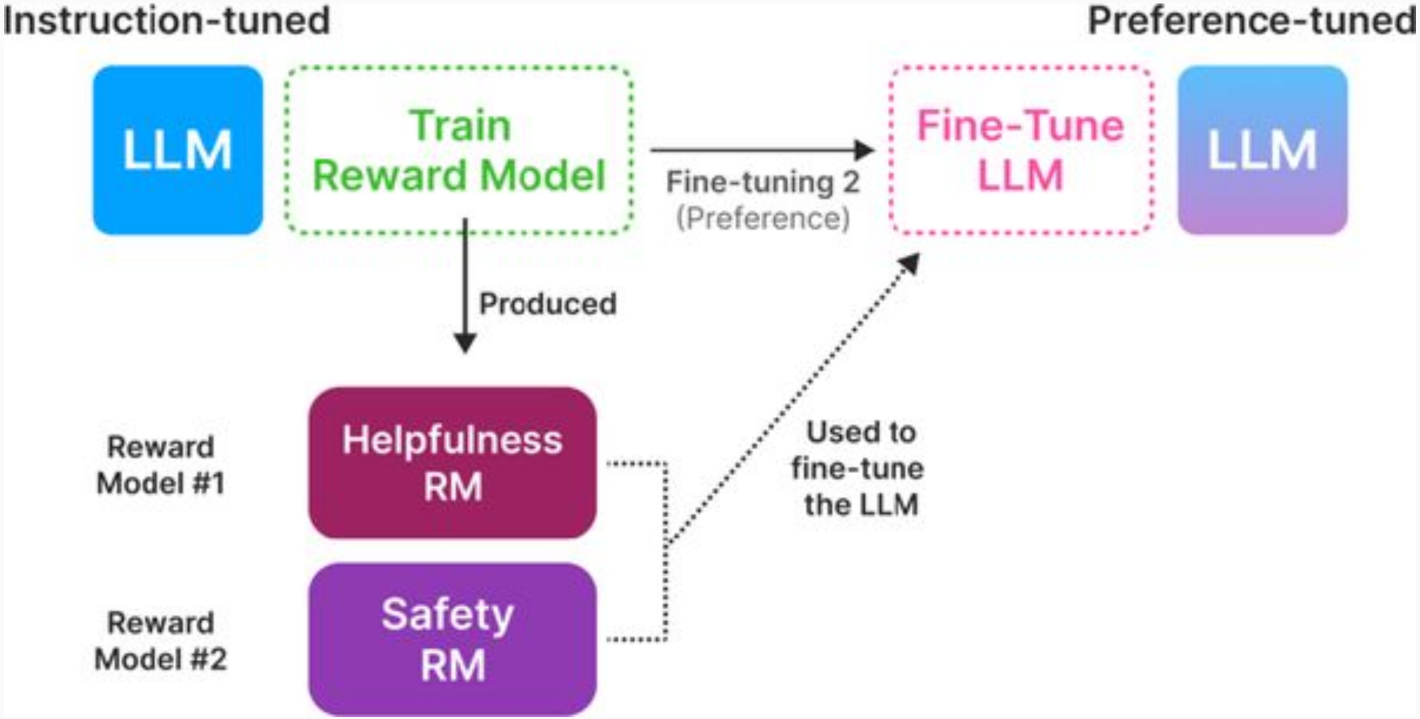


Reinforcement Learning from Human Feedback

Reinforcement Learning from Human Feedback (RLHF) fine-tunes LLMs using labeled outputs and reinforcement learning.



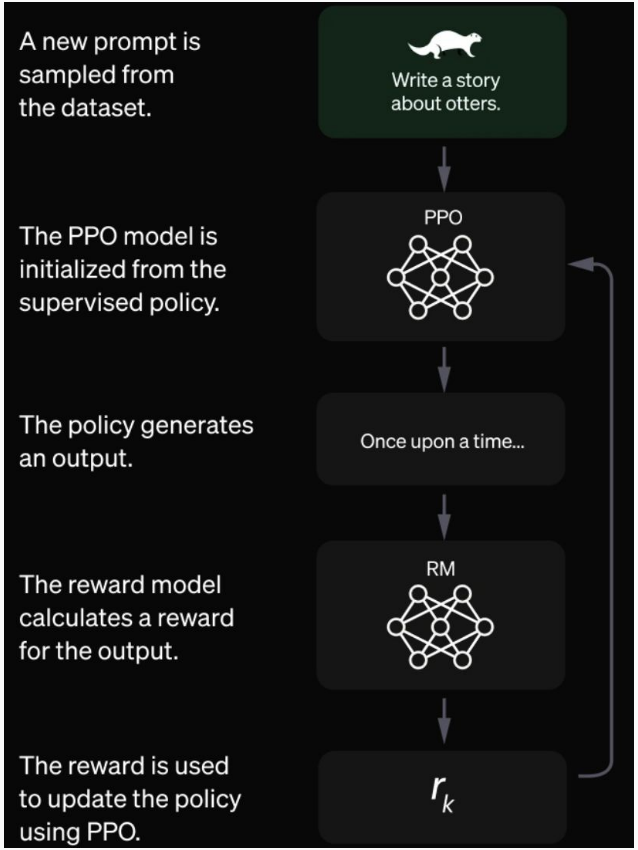
RLHF in action



[Alammar & Grootendorst - Hands-On Large Language Models]()

Proximal Policy Optimization

Proximal Policy Optimization (PPO) makes small, controlled updates to align outputs with human preferences, maintaining stability and generalization consistent with the original model.

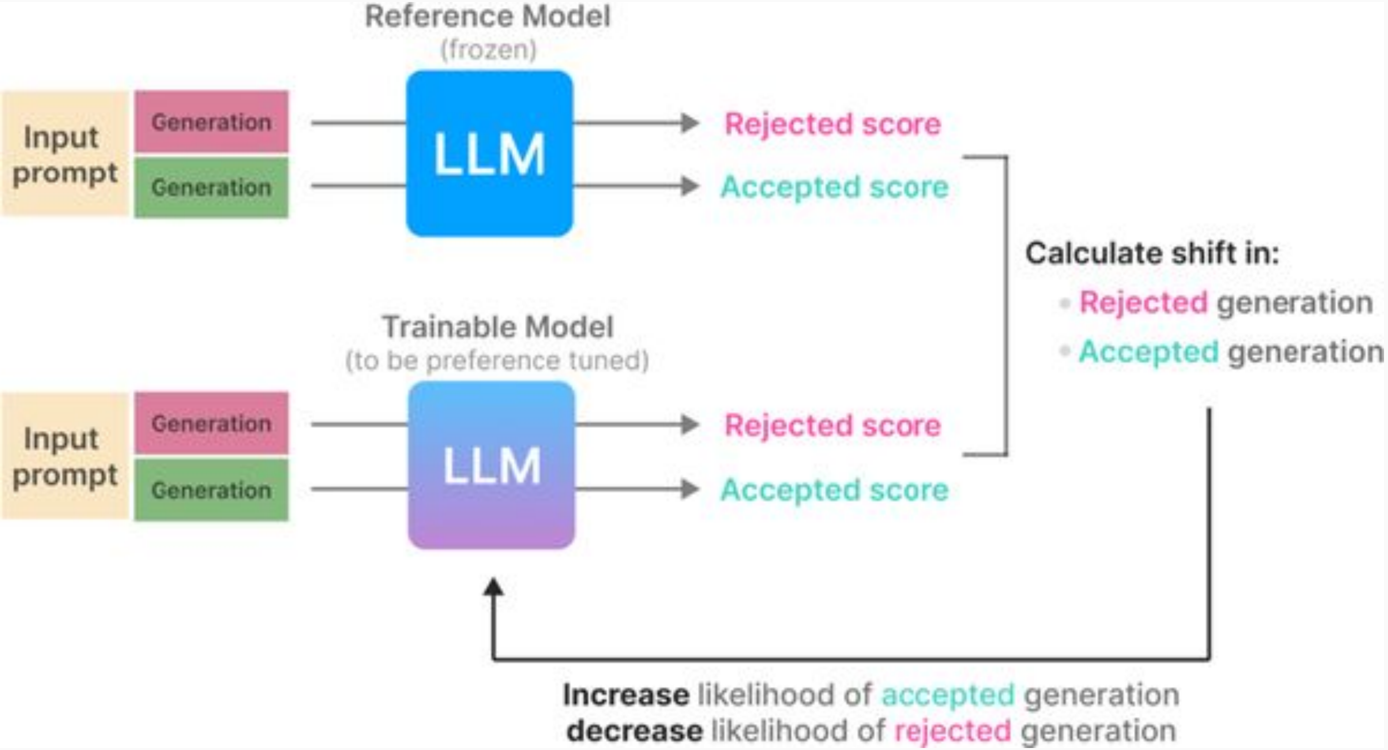


Optimizations

Direct Preference Optimization

Direct Preference Optimization (DPO) is more efficient than PPO, using binary cross-entropy loss instead of reward functions and reinforcement learning.

DPO in action



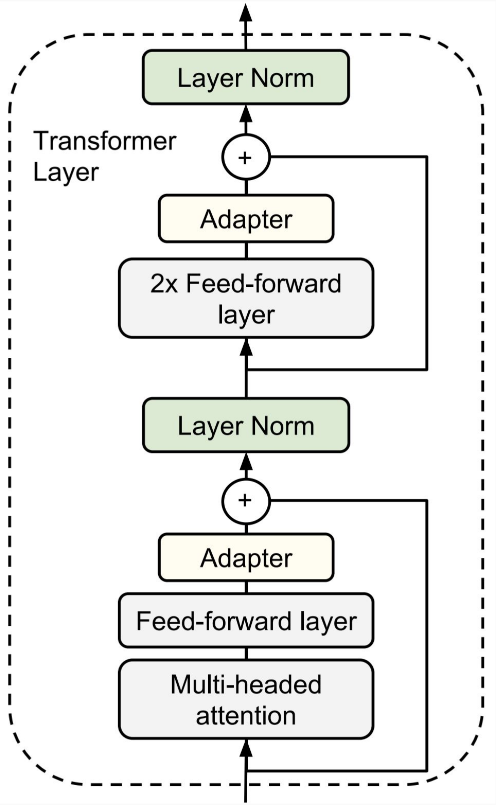
[Alammar & Grootendorst - Hands-On Large Language Models]()

Parameter-Efficient Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) freezes most of the model's original parameters and only trains a small set of parameters or additional layers. This reduces costs and training time without sacrificing much performance.

PEFT: Adapters

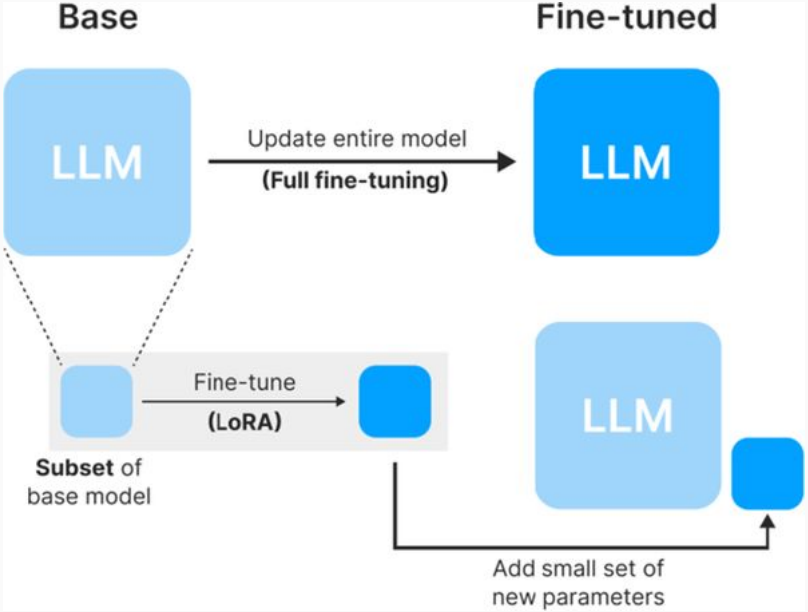
Adapters are modular components added to Transformers that allow for task-specific fine-tuning while leaving the rest of the model frozen. This approach involves 0.5–8% as many parameters as full fine-tuning.



[Parameter-Efficient Transfer Learning for NLP]()

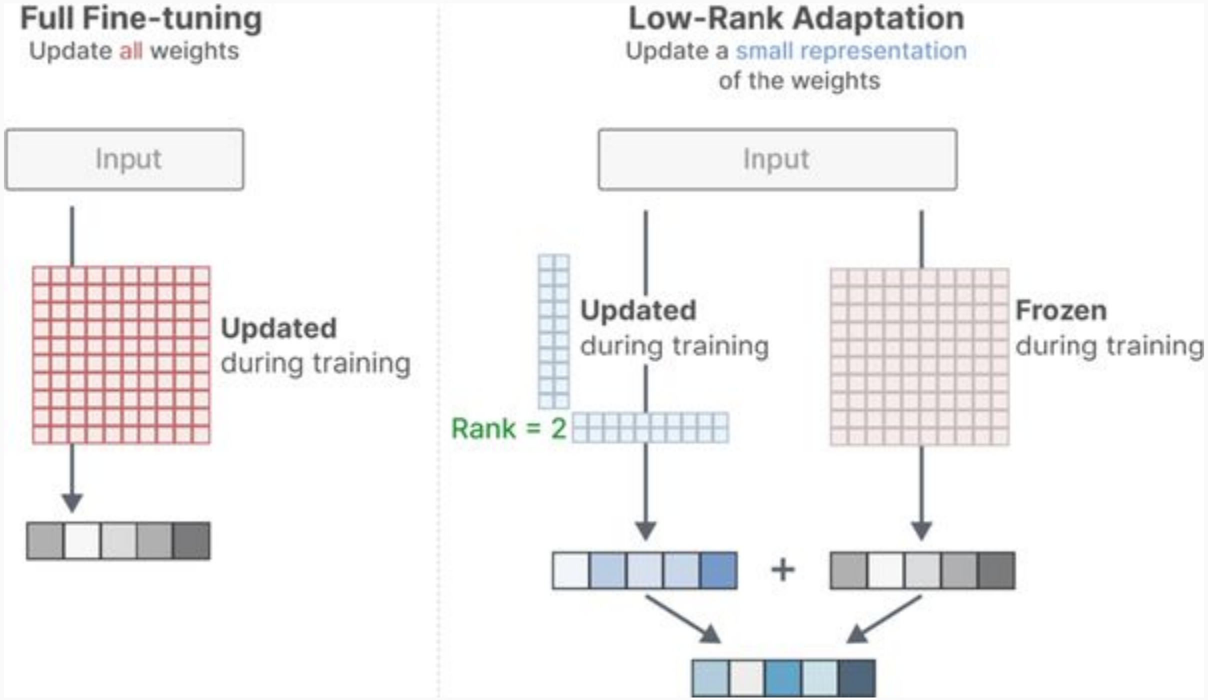
PEFT: Low-Rank Adaptation

Low-Rank Adaptation (LoRA) fine-tunes models by introducing low-rank matrices, significantly reducing the number of trainable parameters.



[*Alammar & Grootendorst - Hands-On Large Language Models*]

LoRA in action

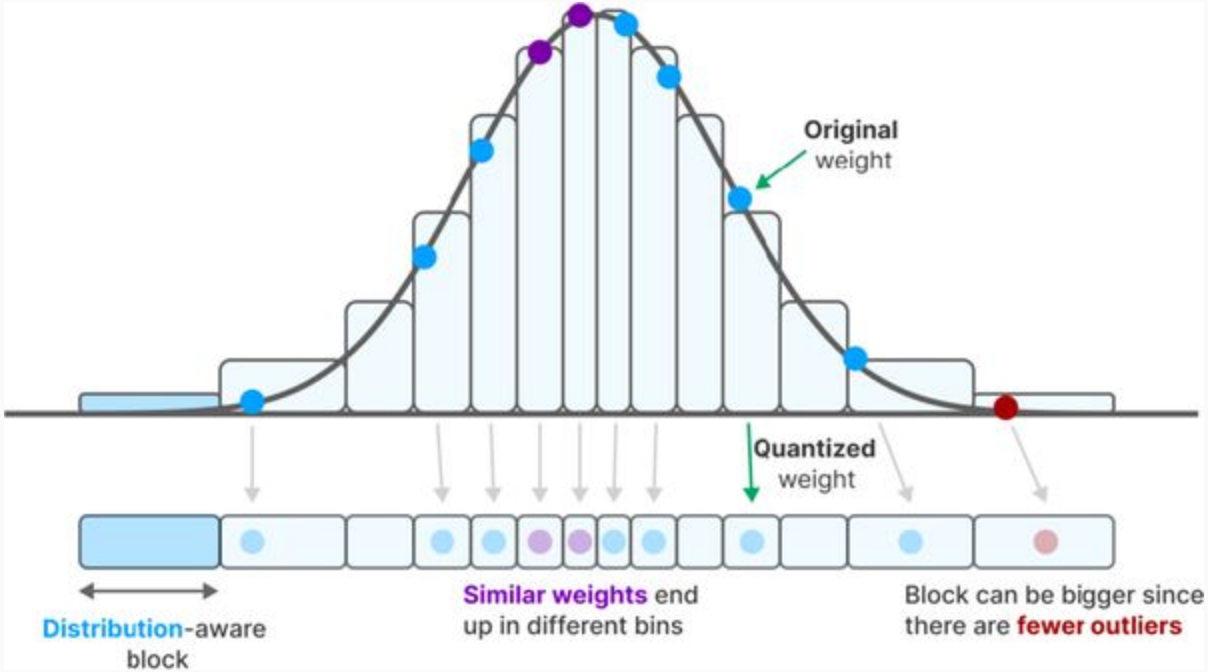


[Alammar & Grootendorst - Hands-On Large Language Models]()

Quantization: QLoRA

Quantization techniques like QLoRA reduce memory and computational requirements by mapping higher-precision weights to lower-precision weights.

QLoRA: block-wise and distribution-aware



[Alammar & Grootendorst - Hands-On Large Language Models](#)

Learning rate schedules

Starting with a small learning rate allows the model to adjust gradually, preventing harmful divergence. Later, lowering the rate helps avoid overshooting during convergence.

Hyperparameter tuning

Optimizing these settings requires experimentation:

- Batch size (number of samples per update)
- Learning rate (magnitude of parameter change per update)
- Number of training epochs (rounds of training)

Evaluation

Datasets

Datasets can be used as-is or augmented for fine-tuning and/or evaluation:

- Instruction: C4, samsum
- Task-specific: MS MARCO, SQuAD
- Domain: MIMIC-III, Law2Vec, AG News
- Alignment: RealToxicityPrompts
- Hugging Face Datasets, Kaggle, Zenodo, Google Dataset Search

Metrics

Word-level metrics provide granular information and are easy to calculate:

- ROUGE (summarization)
- BLEU (translation)
- BERTScore (summarization / translation)
- Perplexity (model confidence)

Benchmarks

Benchmarks can be a great starting point for evaluating model performance:

- MMLU
- GLUE
- SuperGLUE
- BIG-Bench
- TruthfulQA
- GSM8k
- HellaSwag
- HELM
- HumanEval

Automated evaluation

LLM-as-a-Judge uses an LLM to evaluate the quality of one or two others.

Reinforcement Learning from AI Feedback (RLAIF) uses an LLM to train a reward model, which in turn fine-tunes the target model.

Evaluation challenges

Word-level metrics miss many important aspects of model performance.

Benchmarks are insufficiently specific and can be overfitted on.

Fine-tuning for speech recognition

A.L.S. Stole His Voice. A.I. Retrieved It.

In an experiment that surpassed expectations, implants in a patient's brain were able to recognize words he tried to speak, and A.I. helped produce sounds that came close to matching his true voice.

[*A.L.S. Stole His Voice. A.I. Retrieved It.*]

Next steps

Try fine-tuning a model (GPT / Claude / Gemini) using a fine-tuning API!

For a deeper dive, check out libraries like `transformers`, `accelerate`, `peft`, `trl`, and `bitsandbytes` as well as services like AWS Bedrock.