Large Language Models

(Part 2)

April 14, 2026

Plan for today

- **Review of last class:** introduction to large language models (LLMs)
- **Our goals today:**
  - What is fine-tuning in LLMs, and what can it be used for?
  - What are popular approaches to fine-tune a pre-trained LLM model?
  - How do we fine-tune an LLM model in python?

Review: Introduction to LLMs and ChatGPT

Core elements of LLMs/ChatGPT Development

- **Training data:** quality of the training data matters!

Core elements of LLMs/ChatGPT Development

- **Training data:** quality of the training data matters!
- **Pre-training:** initial phase of training a language model (“general-purpose”)

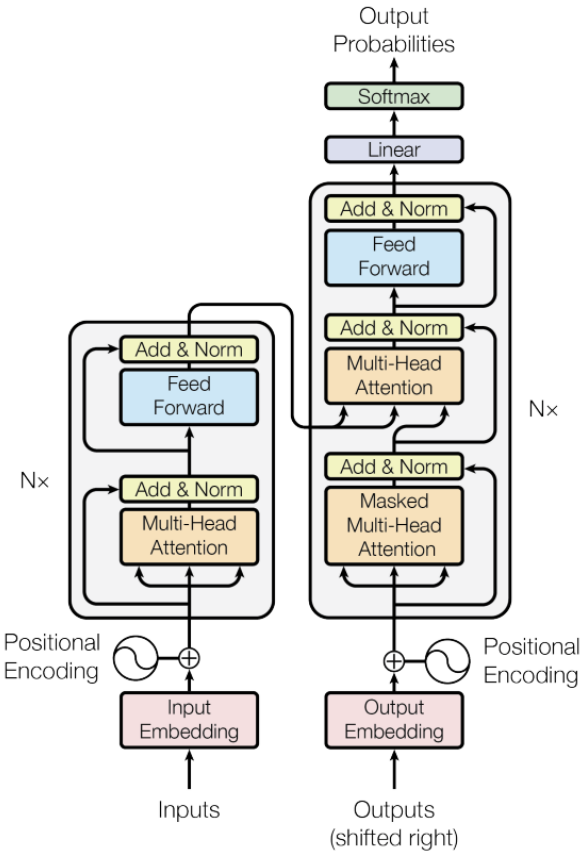
Core elements of LLMs/ChatGPT Development

- **Training data:** quality of the training data matters!
- **Pre-training:** initial phase of training a language model (“general-purpose”)
  - At their core, LLMs (e.g., ChatGPT) are trained to **predict the most likely next word**
    - ↳ Leads to the possibility of false information, hallucinations, and creation of new words

Core elements of LLMs/ChatGPT Development

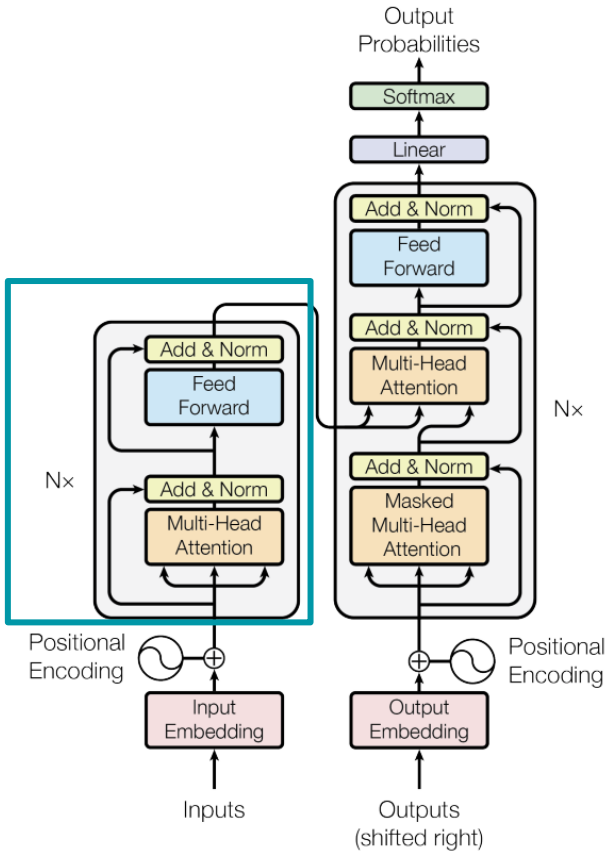
- **Training data:** quality of the training data matters!
- **Pre-training:** initial phase of training a language model (“general-purpose”)
  - At their core, LLMs (e.g., ChatGPT) are trained to **predict the most likely next word**
    - ↳ Leads to the possibility of false information, hallucinations, and creation of new words
  - Architecture: **transformers** (encoder + decoder, with attention)

Transformer Architecture



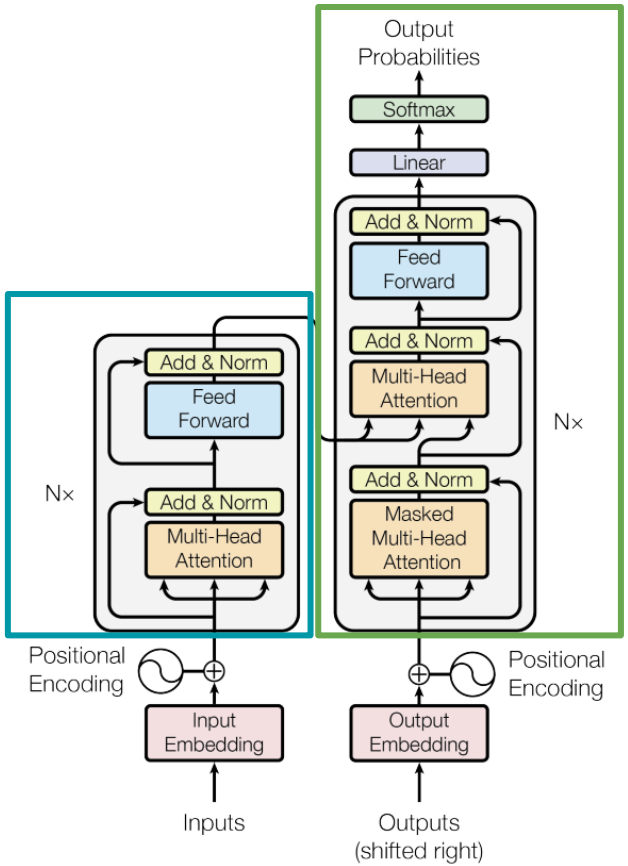
Transformer Architecture

Encoder
[Learnt Representation or Embedding]



Transformer Architecture

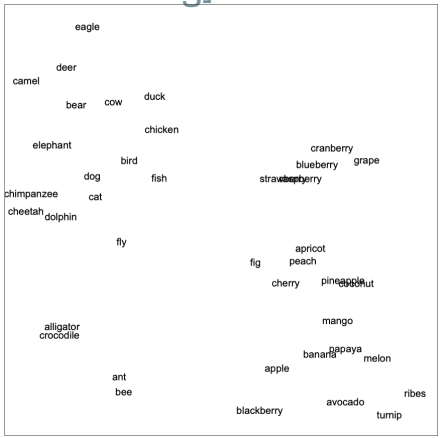
Encoder
[Learnt Representation or Embedding]



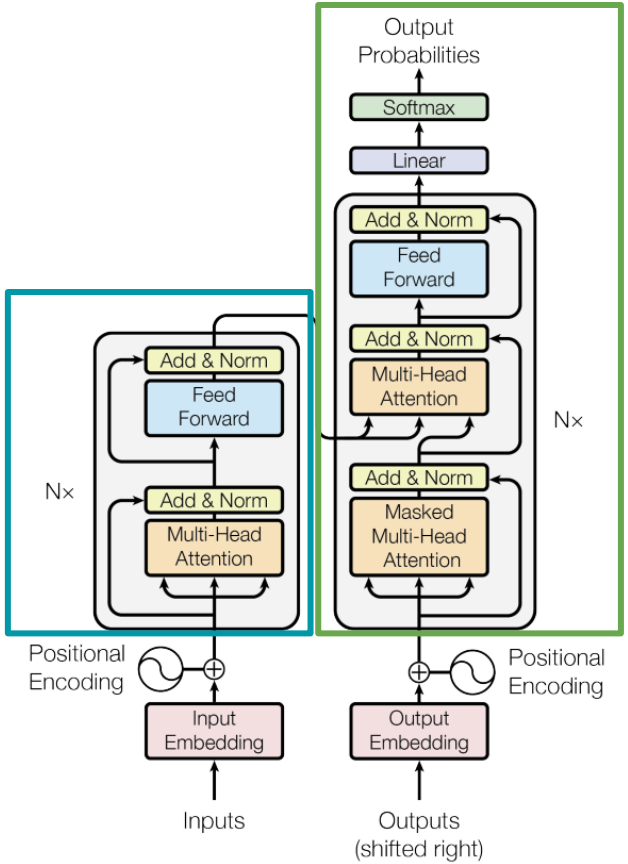
Decoder
[Generation]

Transformer Architecture

Encoder
[Learnt Representation or Embedding]



Example of learnt embeddings from encoder

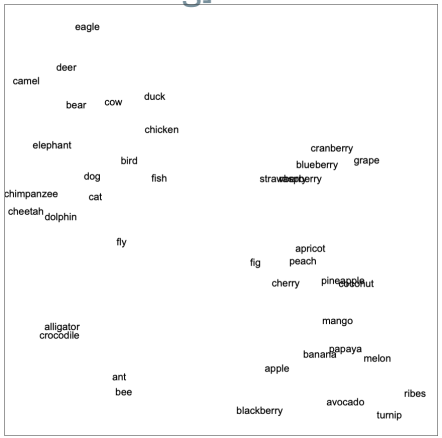


Decoder
[Generation]

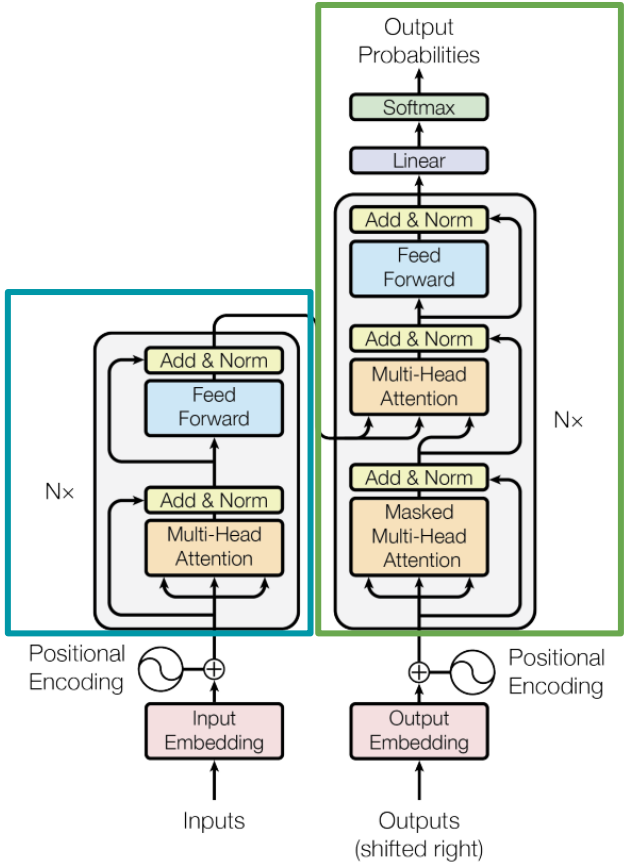
Transformer Architecture

Encoder

[Learnt Representation or Embedding]



Example of learnt embeddings from encoder



Decoder

[Generation]

The best thing about AI is its ability to

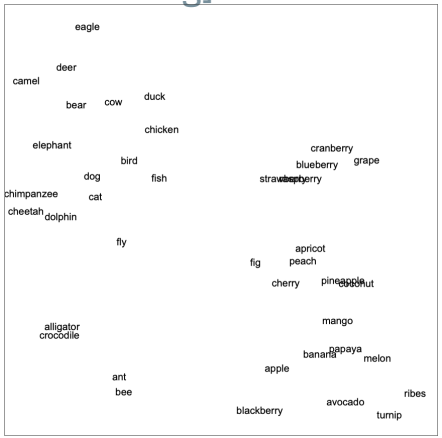
learn	4.5%
predict	3.5%
make	3.2%
understand	3.1%
do	2.9%

Example output of decoder

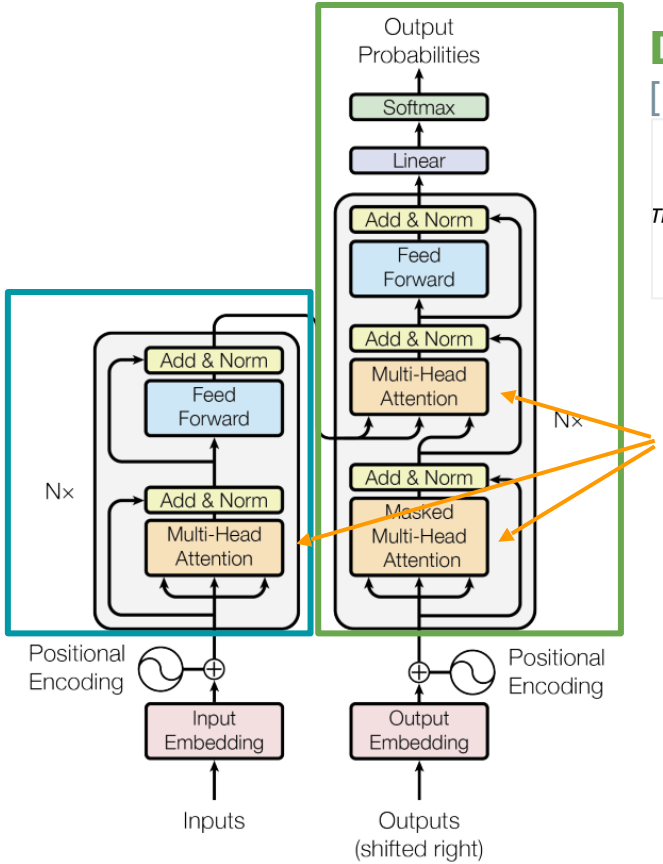
Transformer Architecture

Encoder

[Learnt Representation or Embedding]



Example of learnt embeddings from encoder



Decoder

[Generation]

The best thing about AI is its ability to	learn	4.5%
	predict	3.5%
	make	3.2%
	understand	3.1%
	do	2.9%

Example output of decoder

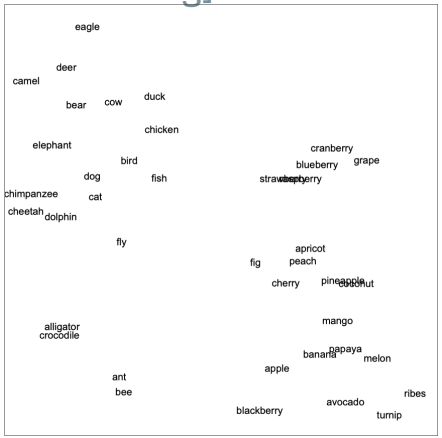
Attention

[to capture long-range context]

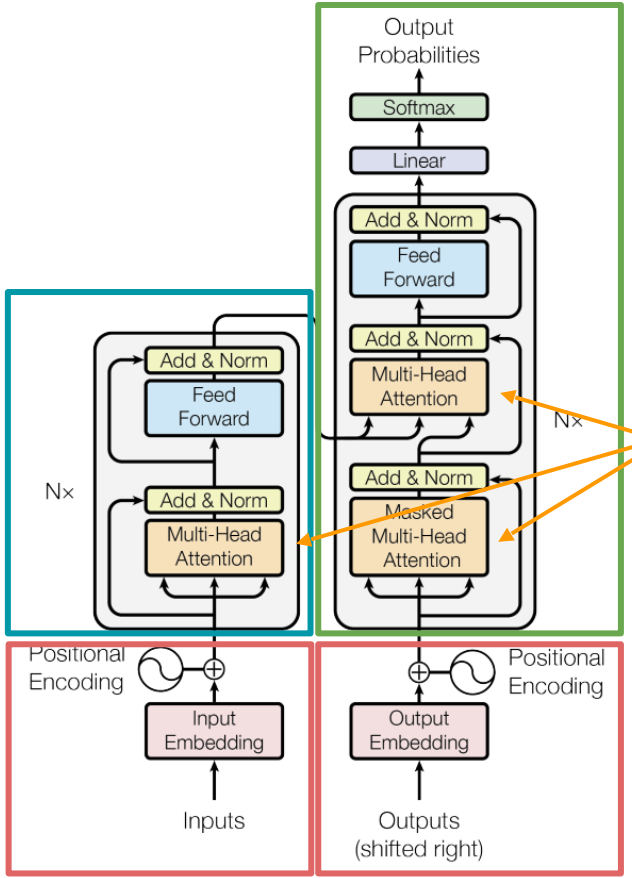
Transformer Architecture

Encoder

[Learnt Representation or Embedding]



Example of learnt embeddings from encoder



Decoder

[Generation]

The best thing about AI is its ability to	learn	4.5%
	predict	3.5%
	make	3.2%
	understand	3.1%
	do	2.9%

Example output of decoder

Attention

[to capture long-range context]

Text-to-numeric processing

[Token-value + Token-position embeddings]

Core elements of LLMs/ChatGPT Development

- **Training data:** quality of the training data matters!
- **Pre-training:** initial phase of training a language model (“general-purpose”)
  - At their core, LLMs (e.g., ChatGPT) are trained to **predict the most likely next word**
    - ↳ Leads to the possibility of false information, hallucinations, and creation of new words
  - Architecture: **transformers** (encoder + decoder, with attention)

Core elements of LLMs/ChatGPT Development

- **Training data:** quality of the training data matters!
- **Pre-training:** initial phase of training a language model (“general-purpose”)
  - At their core, LLMs (e.g., ChatGPT) are trained to **predict the most likely next word**
    - ↳ Leads to the possibility of false information, hallucinations, and creation of new words
  - Architecture: **transformers** (encoder + decoder, with attention)
- **Fine-tuning:** process of further training a language model for a specific task or domain (“specialized”) using additional (generally high-quality) data

Fine-Tuning

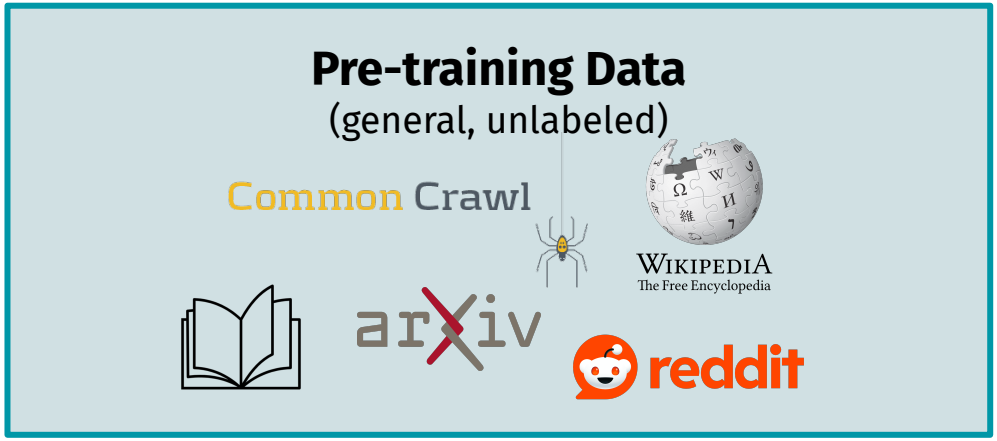
What is fine-tuning?

What is fine-tuning?

- **Fine-tuning:** process of further training a language model for a specific task or domain (“specialized”) using additional (generally high-quality) data
  - GPT3/4 = general-purpose (GPT = generative pre-trained transformer)
  - ChatGPT3/4 = specialized for conversations

What is fine-tuning?

- **Fine-tuning:** process of further training a language model for a specific task or domain ("specialized") using additional (generally high-quality) data
  - GPT3/4 = general-purpose (GPT = generative pre-trained transformer)
  - ChatGPT3/4 = specialized for conversations



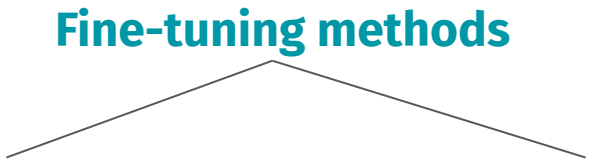
Fine Tuning Data
(specialized, high-quality)

What is fine-tuning?

- **Fine-tuning:** process of further training a language model for a specific task or domain ("specialized") using additional (generally high-quality) data
  - GPT3/4 = general-purpose (GPT = generative pre-trained transformer)
  - ChatGPT3/4 = specialized for conversations
- **Example use cases for fine-tuning:**
  - *Application areas:* legal documents, medical reports, sports analytics, company data, …
  - *Types of tasks:* custom chatbots, translation, summarization, sentiment analysis, various classification/regression tasks, …

Main approaches for fine-tuning

Main approaches for fine-tuning



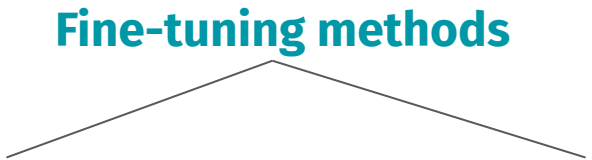
Supervised fine-tuning (SFT)

use labeled data (e.g., prompt-response pairs) from a specific task/domain to adjust the learned weights in the model

Reinforcement learning with human feedback (RLHF)

use human feedback to adjust the weights in the model

Main approaches for fine-tuning



Supervised fine-tuning (SFT)

use labeled data (e.g., prompt-response pairs) from a specific task/domain to adjust the learned weights in the model

- What data to use?
- How do we adjust the weights?

Reinforcement learning with human feedback (RLHF)

use human feedback to adjust the weights in the model

Supervised fine-tuning (SFT)

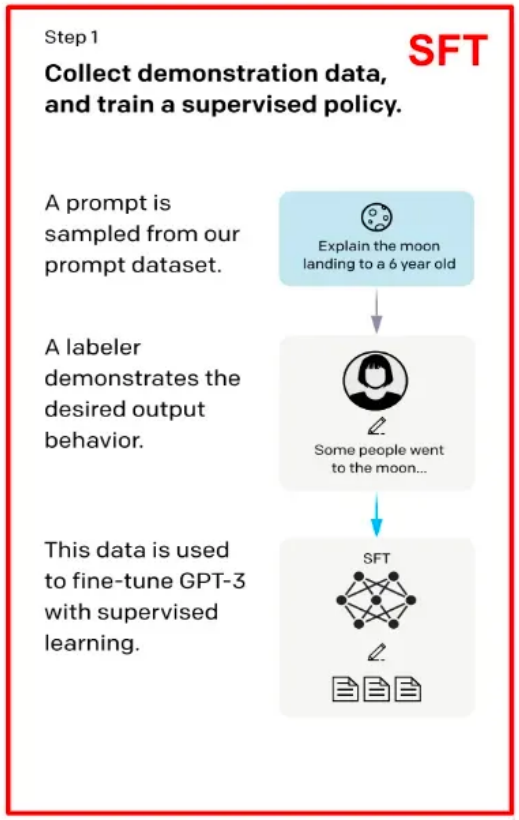


Figure source: Ouyang et al. (2022)

Supervised fine-tuning (SFT)

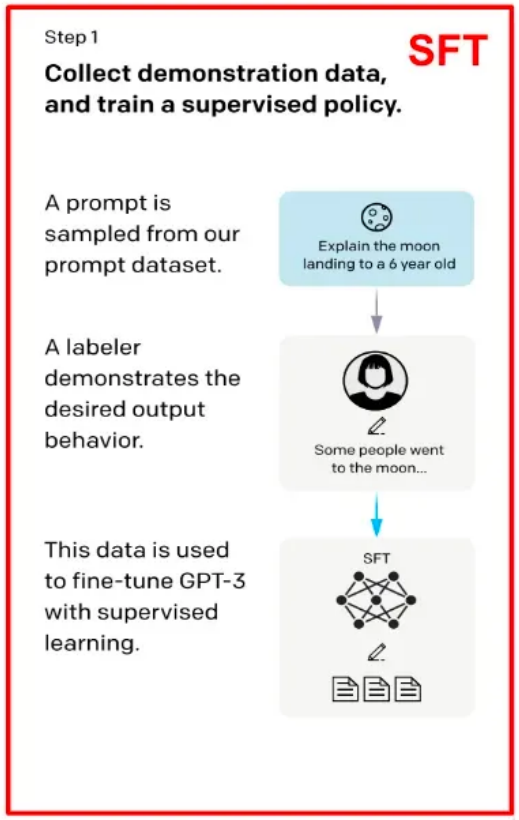


Figure source: Ouyang et al. (2022)

Examples of "supervised" data:

- Prompt-response pairs (e.g., online customer service chats)
- Question-answer pairs (e.g., stack exchange, exam questions)
- Translations (e.g., translated books, religious texts)
- Text/sequence classification (e.g., Yelp, movie reviews)

Supervised fine-tuning (SFT)

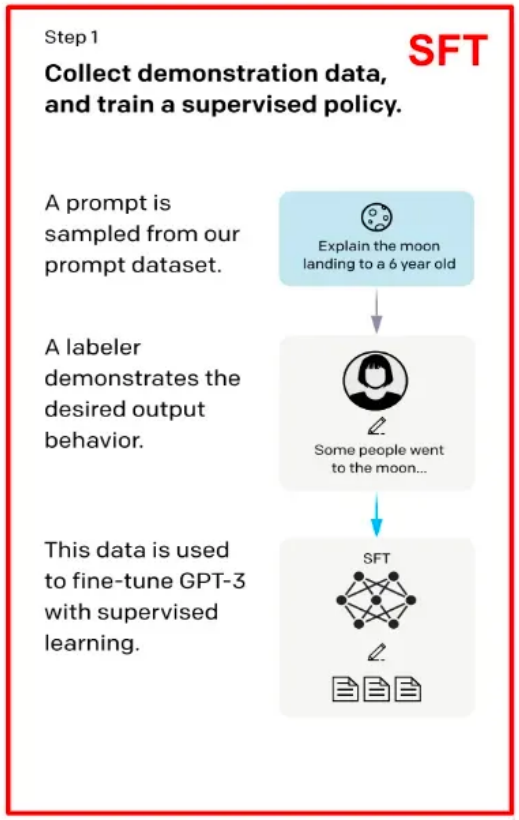


Figure source: Ouyang et al. (2022)

Examples of "supervised" data:

- Prompt-response pairs (e.g., online customer service chats)
- Question-answer pairs (e.g., stack exchange, exam questions)
- Translations (e.g., translated books, religious texts)
- Text/sequence classification (e.g., Yelp, movie reviews)

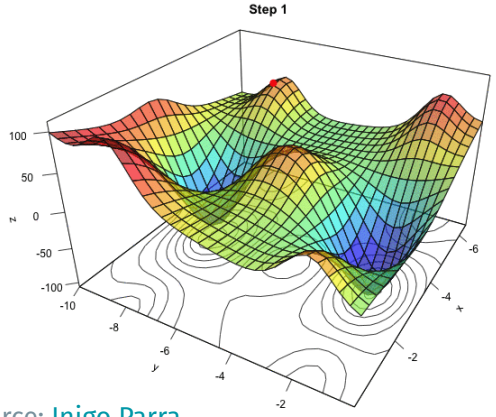


Figure source: Inigo Parra

Supervised fine-tuning (SFT)

How do we update the model weights when fine-tuning?

Supervised fine-tuning (SFT)

How do we update the model weights when fine-tuning?

Full fine-tuning

(Update all model parameters)

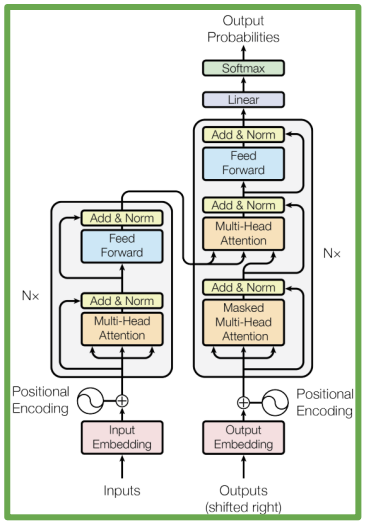
Supervised fine-tuning (SFT)

How do we update the model weights when fine-tuning?

Full fine-tuning

(Update all model parameters)

update everything



Supervised fine-tuning (SFT)

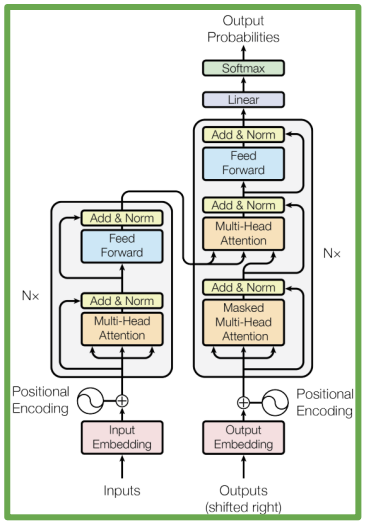
How do we update the model weights when fine-tuning?

Full fine-tuning

(Update all model parameters)

Downsides: "catastrophic forgetting" (may forget how to do general tasks), requires huge amounts of memory

update everything



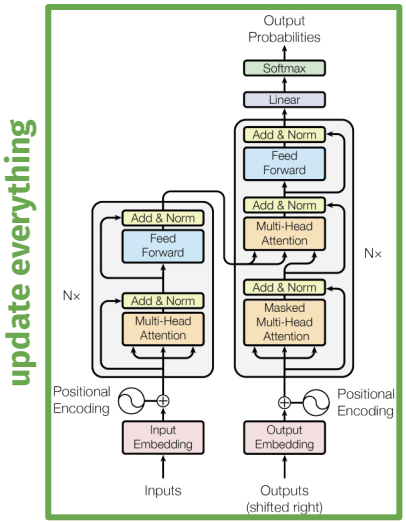
Supervised fine-tuning (SFT)

How do we update the model weights when fine-tuning?

Full fine-tuning

(Update all model parameters)

Downsides: "catastrophic forgetting" (may forget how to do general tasks), requires huge amounts of memory



Parameter-efficient fine-tuning (PEFT)

(Partially update model parameters)

Supervised fine-tuning (SFT)

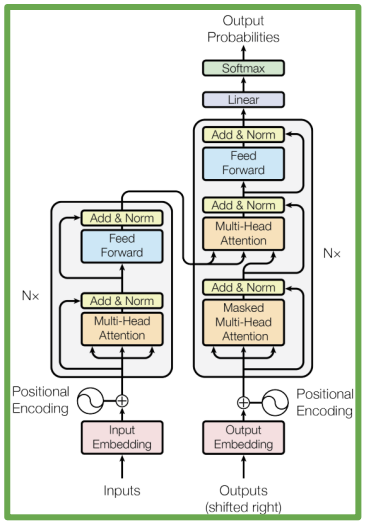
How do we update the model weights when fine-tuning?

Full fine-tuning

(Update all model parameters)

Downsides: "catastrophic forgetting" (may forget how to do general tasks), requires huge amounts of memory

update everything



Parameter-efficient fine-tuning (PEFT)

(Partially update model parameters)

Selective

update subset of original LLM parameters; freeze all others

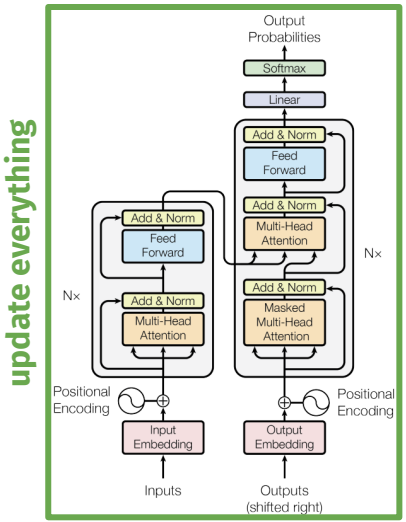
Supervised fine-tuning (SFT)

How do we update the model weights when fine-tuning?

Full fine-tuning

(Update all model parameters)

Downsides: "catastrophic forgetting" (may forget how to do general tasks), requires huge amounts of memory

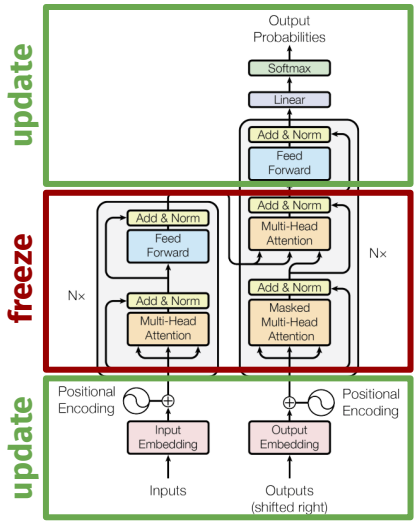


Parameter-efficient fine-tuning (PEFT)

(Partially update model parameters)

Selective

update subset of original LLM parameters; freeze all others



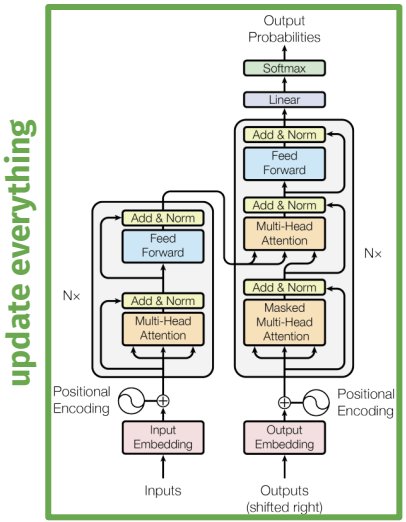
Supervised fine-tuning (SFT)

How do we update the model weights when fine-tuning?

Full fine-tuning

(Update all model parameters)

Downsides: "catastrophic forgetting" (may forget how to do general tasks), requires huge amounts of memory

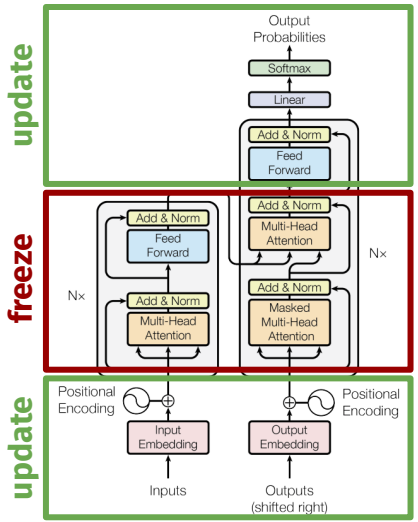


Parameter-efficient fine-tuning (PEFT)

(Partially update model parameters)

Selective

update subset of original LLM parameters; freeze all others



Additive/adapters

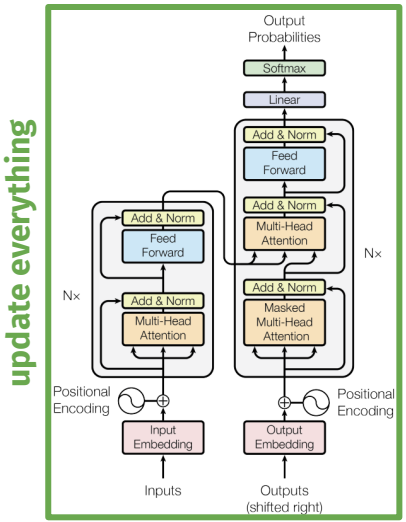
add new trainable components on top of base pre-trained LLM

Supervised fine-tuning (SFT)

How do we update the model weights when fine-tuning?

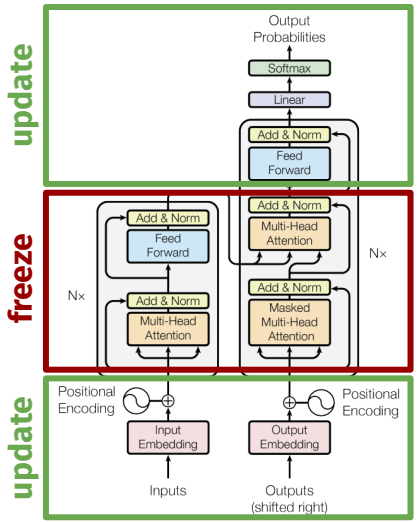
Full fine-tuning
(Update all model parameters)

Downsides: "catastrophic forgetting" (may forget how to do general tasks), requires huge amounts of memory

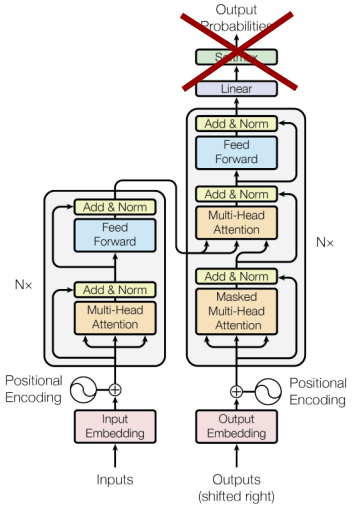


Parameter-efficient fine-tuning (PEFT)
(Partially update model parameters)

Selective
update subset of original LLM parameters; freeze all others



Additive/adapters
add new trainable components on top of base pre-trained LLM

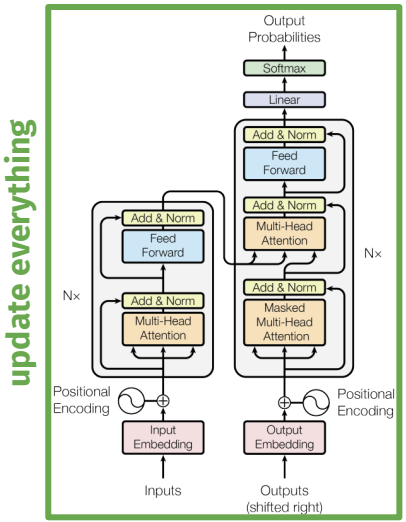


Supervised fine-tuning (SFT)

How do we update the model weights when fine-tuning?

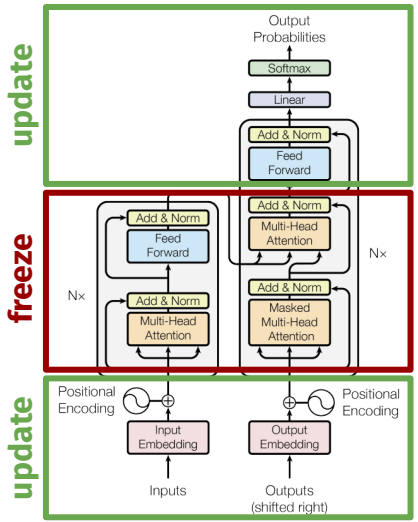
Full fine-tuning
(Update all model parameters)

Downsides: "catastrophic forgetting" (may forget how to do general tasks), requires huge amounts of memory

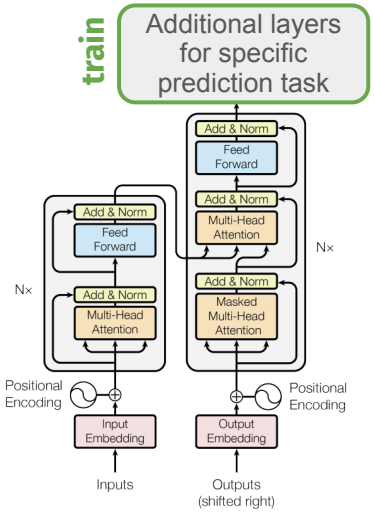


Parameter-efficient fine-tuning (PEFT)
(Partially update model parameters)

Selective
update subset of original LLM parameters; freeze all others



Additive/adapters
add new trainable components on top of base pre-trained LLM

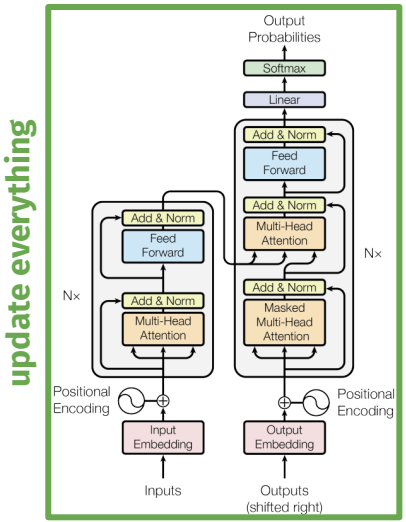


Supervised fine-tuning (SFT)

How do we update the model weights when fine-tuning?

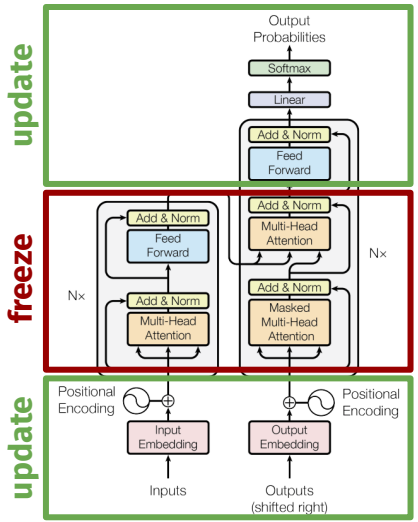
Full fine-tuning
(Update all model parameters)

Downsides: "catastrophic forgetting" (may forget how to do general tasks), requires huge amounts of memory

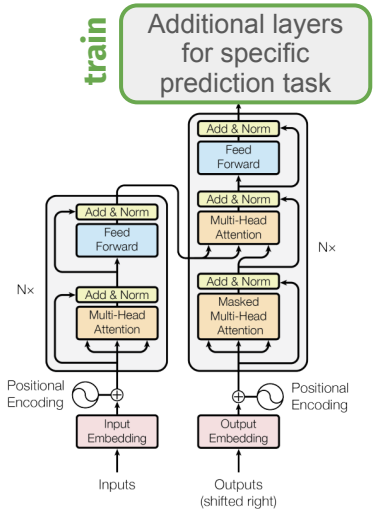


Parameter-efficient fine-tuning (PEFT)
(Partially update model parameters)

Selective
update subset of original LLM parameters; freeze all others



Additive/adapters
add new trainable components on top of base pre-trained LLM



Reparameterization
reparameterize weights to reduce number of parameters that need to be learned/updated

Reparameterized PEFT

Low-Rank Adaptation (LoRA): a popular way to reparameterize weights in SFT

- Reduces number of parameters to train by only allowing for **low-rank** updates to the original network weights (often applied to weights in attention or feed-forward layers)

Figure source: Dasha Herrmannova (2024)

Reparameterized PEFT

Low-Rank Adaptation (LoRA): a popular way to reparameterize weights in SFT

- Reduces number of parameters to train by only allowing for **low-rank** updates to the original network weights (often applied to weights in attention or feed-forward layers)

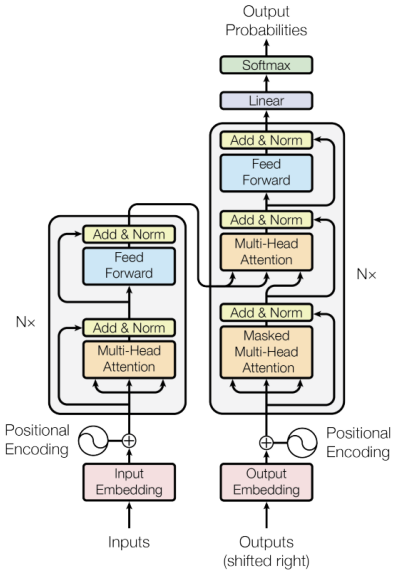


Figure source: Dasha Herrmannova (2024)

Reparameterized PEFT

Low-Rank Adaptation (LoRA): a popular way to reparameterize weights in SFT

- Reduces number of parameters to train by only allowing for **low-rank** updates to the original network weights (often applied to weights in attention or feed-forward layers)

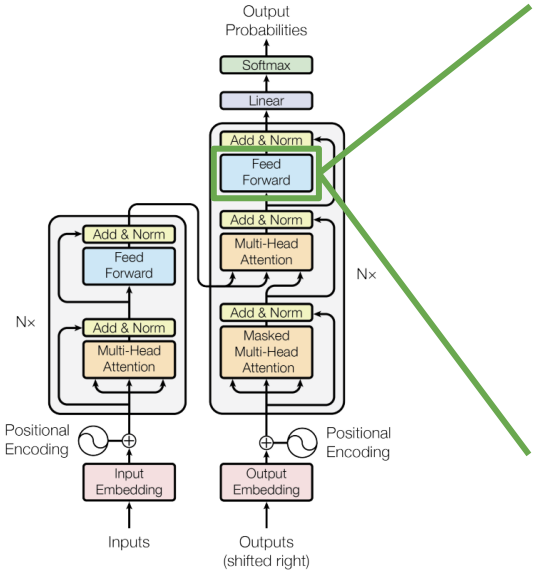
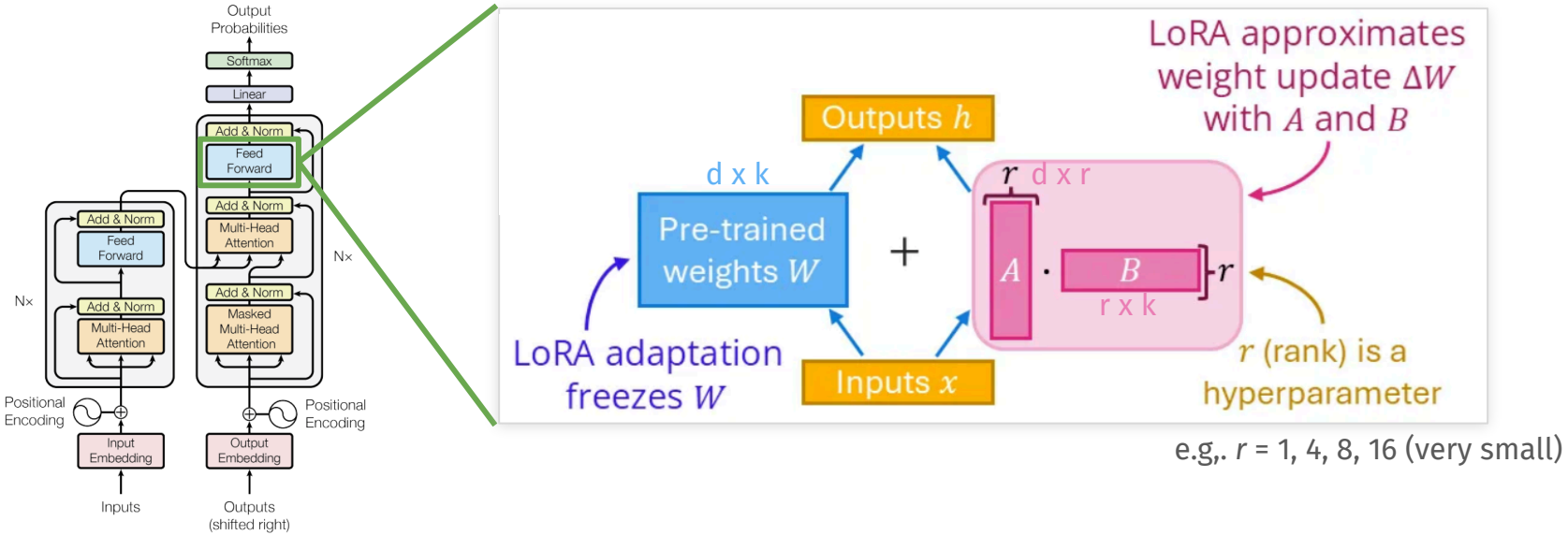


Figure source: Dasha Herrmannova (2024)

Reparameterized PEFT

Low-Rank Adaptation (LoRA): a popular way to reparameterize weights in SFT

- Reduces number of parameters to train by only allowing for **low-rank** updates to the original network weights (often applied to weights in attention or feed-forward layers)



e.g,. r = 1, 4, 8, 16 (very small)

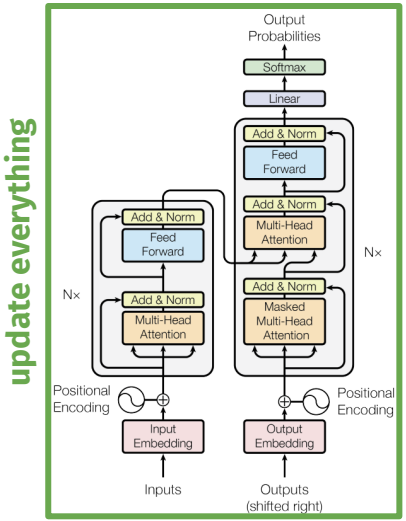
Figure source: Dasha Herrmannova (2024)

Supervised fine-tuning (SFT)

How do we update the model weights when fine-tuning?

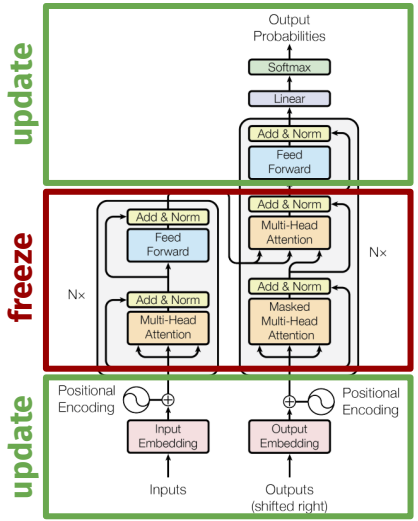
Full fine-tuning
(Update all model parameters)

Downsides: "catastrophic forgetting" (may forget how to do general tasks), requires huge amounts of memory

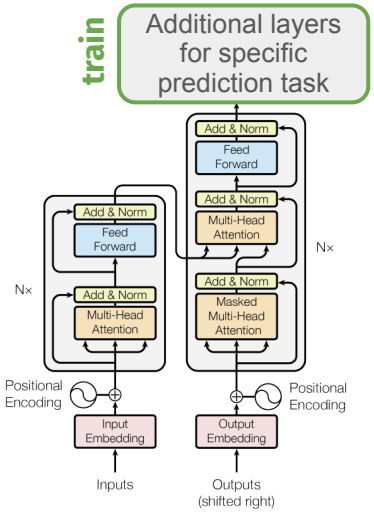


Parameter-efficient fine-tuning (PEFT)
(Partially update model parameters)

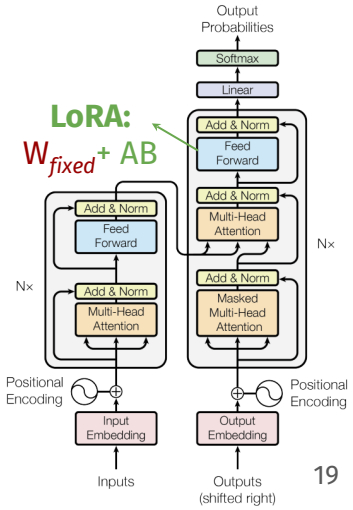
Selective
update subset of original LLM parameters; freeze all others



Additive/adapters
add new trainable components on top of base pre-trained LLM



Reparameterization
reparameterize weights



Supervised fine-tuning in action using **huggingface**

Huggingface: open-source library with pre-trained models

- **Transformers Library**
  - Pre-trained models for NLP tasks (e.g., BERT, GPT, T5, RoBERTa) [https://huggingface.co/models]
  - Easy-to-use APIs for model loading, fine-tuning, and prediction (inference)
- **Datasets Library**
  - Provides easy access to thousands of public datasets
  - Built-in support for data loading, preprocessing, and transformation
- **Tokenizers Library**
  - A fast and efficient library for text tokenization, optimized for performance
  - Supports parallel tokenization
- **Tutorial notebooks:** https://huggingface.co/docs/transformers/main/en/notebooks

Supervised fine-tuning in action using **huggingface**

Huggingface: open-source library with pre-trained models

- **Transformers Library**
  - Pre-trained models for NLP tasks (e.g., BERT, GPT, T5, RoBERTa) [https://huggingface.co/models]
  - Easy-to-use APIs for model loading, fine-tuning, and prediction (inference)
- **Datasets Library**
  - Provides easy access to thousands of public datasets
  - Built-in support for data loading, preprocessing, and transformation
- **Tokenizers Library**
  - A fast and efficient library for text tokenization, optimized for performance
  - Supports parallel tokenization
- **Tutorial notebooks:** https://huggingface.co/docs/transformers/main/en/notebooks

Go to: https://colab.research.google.com/github/tiffanymtang/dsip-s26/blob/main/course_materials/llms/llm_fine_tuning.ipynb

Main approaches for fine-tuning

Supervised fine-tuning (SFT)

use labeled data (e.g., prompt-response pairs) from a specific task/domain to adjust the learned weights in the model

Reinforcement learning with human feedback (RLHF)

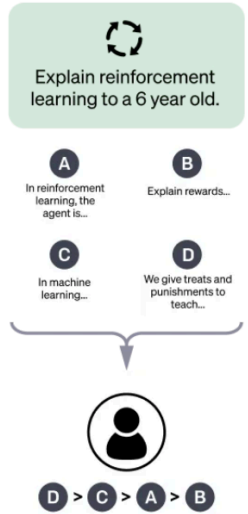
use human feedback to adjust the weights in the model

Reinforcement learning with human feedback

Given pre-trained model:

1. Collect human feedback

A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.

Figure source: Ouyang et al. (2022)

Reinforcement learning with human feedback

Given pre-trained model:

1. Collect human feedback
2. Train reward model to mimic human preferences

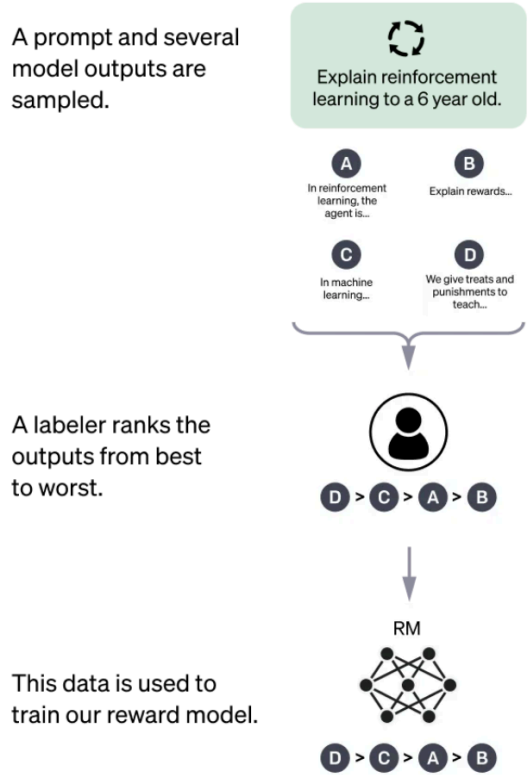


Figure source: Ouyang et al. (2022)

Reinforcement learning with human feedback

Given pre-trained model:

1. Collect human feedback
2. Train reward model to mimic human preferences
3. Fine-tune model using reinforcement learning
    a. Feed training samples through model

A new prompt is sampled from the dataset.

Figure source: Ouyang et al. (2022)

Reinforcement learning with human feedback

Given pre-trained model:

1. Collect human feedback
2. Train reward model to mimic human preferences
3. Fine-tune model using reinforcement learning
    a. Feed training samples through model
    b. Model generates outputs

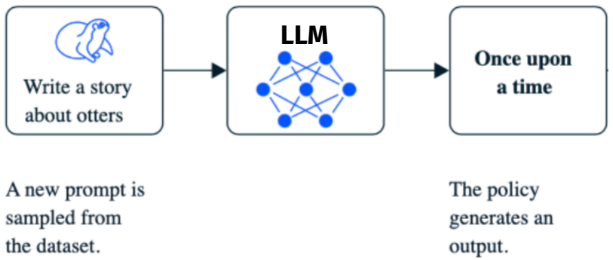


Figure source: Ouyang et al. (2022)

Reinforcement learning with human feedback

Given pre-trained model:

1. Collect human feedback
2. Train reward model to mimic human preferences
3. Fine-tune model using reinforcement learning
    a. Feed training samples through model
    b. Model generates outputs
    c. Outputs get scored by reward model

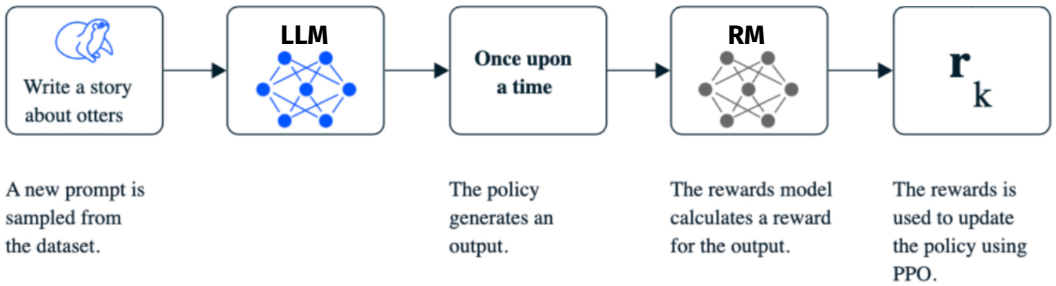


Figure source: Ouyang et al. (2022)

Reinforcement learning with human feedback

Given pre-trained model:

1. Collect human feedback
2. Train reward model to mimic human preferences
3. Fine-tune model using reinforcement learning
    a. Feed training samples through model
    b. Model generates outputs
    c. Outputs get scored by reward model
    d. Update model to maximize reward

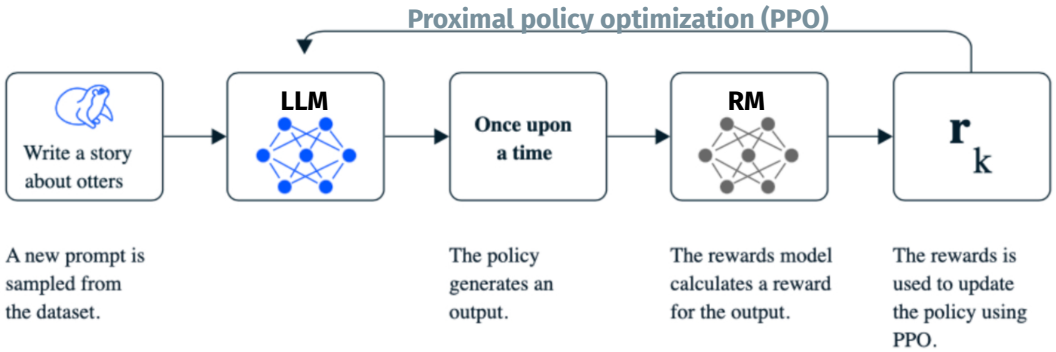


Figure source: Ouyang et al. (2022)

Summary

Summary and Next Time

- **Fine-tuning:** process of further training a language model for a specific task or domain ("specialized") using additional (generally high-quality) data
- **Main approaches for fine-tuning:**
  - *Supervised fine-tuning (SFT):* uses labeled (prompt-response) data
    - Can update weights completely, selectively, additively, or using low-rank adaptations
  - *Reinforcement learning with human feedback (RLHF):* uses human feedback
- Other approaches to improve LLMs: *prompt engineering*, *in-context learning*, and *RAG*