

<https://www.deeplearning.ai/short-courses/post-training-of-langs/>

Post Training of LLMs	2
Introduction	2
Introduction to Post-Training	13
What is Post-training	14
Methods Used During LLM training	17
Post-Training of LLMs	24
SFT : limitating Example Responses	26
Best Use Cases for SFT	29
Principles of SFT Data Curation	31
Full Fine-Tuning vs Parameter Efficient Fine-Tuning (PEFT)	33
Post-Training of LLMs	36
SFT : Limitating example Response	37
L3 Supervised Fine-Tuning (SFT)	41
Setting up Helper Functions	41
load_model_and_tokenizer	45
display_dataset	49
Load base Model & Test on Simple Question	51
Base Model (After SFT) Output	57
Doing SFT on Small Model	61
load_dataset	66
SFTConfig	71
SFTTrainier	82
Testing incomplete SFT Training results	86
Basics of DPO	91
DPO : Contrastive Learning from Positive and Negative Samples	94
DPO minizmizes	100
Best Use Case for DPO	103
Principles of DPO Data Curation	105
DPO in Practice	107
DPO: Contrastive Learning from Positive and Negative Samples	111
L5 : Direct Preference Optimization	114
Load Instruct Model & Test on Simple Questions	118
Results of the DPO-trained Model	125
Load the small model for training without GPUs	131
DPO training	146
Basic of Online RL	163
Reinforcement Learning for LLMs: Online vs Offline	165
Online RL : let Model Explore Better Response by itselv	170
Reward Function in Online RL	171
Option 1 : Trained Reward Modle	171
Option 2: Verifiable Reward	174
Policy Training in Online RL	176
PPO	176
GRPO	176
JPPO (@)	178
GRPO vs PPO	184
Online RL in Practice	186
Online RL : Let the model explore batter response by itself	188
L7 online RL : import librarires	191
PRepare for evaluation dataset for Math : GSM8K	193

Load the Evaluation Dataset	202
Load the model and evaluate	211
Load the training dataset	226
GRPO Training	230
Results of the fully trained Qwen model	241
Conclusion	256
Common methods in post-training	258
Why online RL degrades performance less compared with SFT	261

Post Training of LLMs

Introduction

12:40



音量



5G



51



earn.deeplearning.ai



13



DeepLearning.AI



Post-training of LLMs



University of
Washington



NexusFlow AI

Banghua Zhu

Banghua has trained and post-trained
in many models

12:41 香港

5G 50



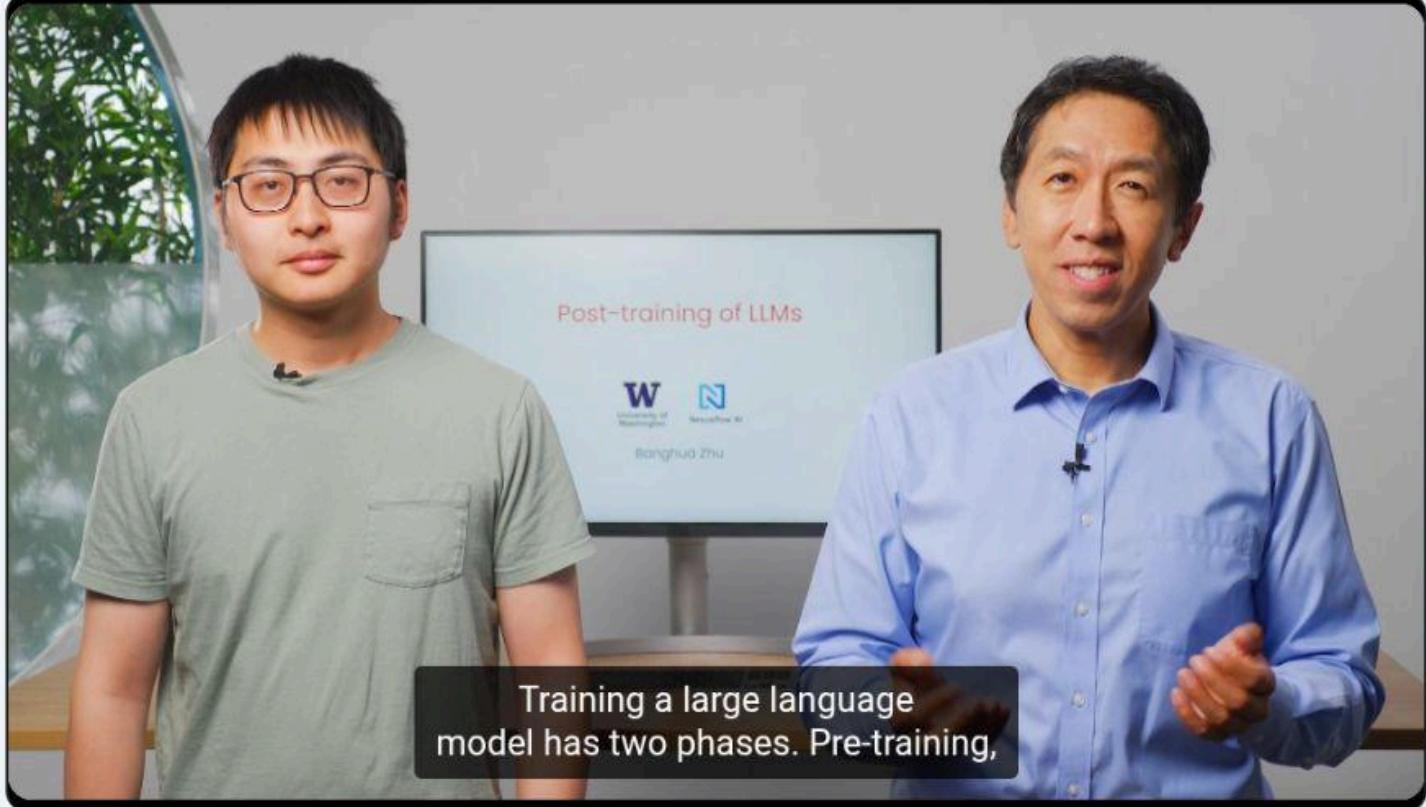
earn.deeplearning.ai



13



DeepLearning.AI



12:41 小红书 香港

5G 50



earn.deeplearning.ai



13



DeepLearning.AI



12:42



香港

5G



5G



50



earn.deeplearning.ai

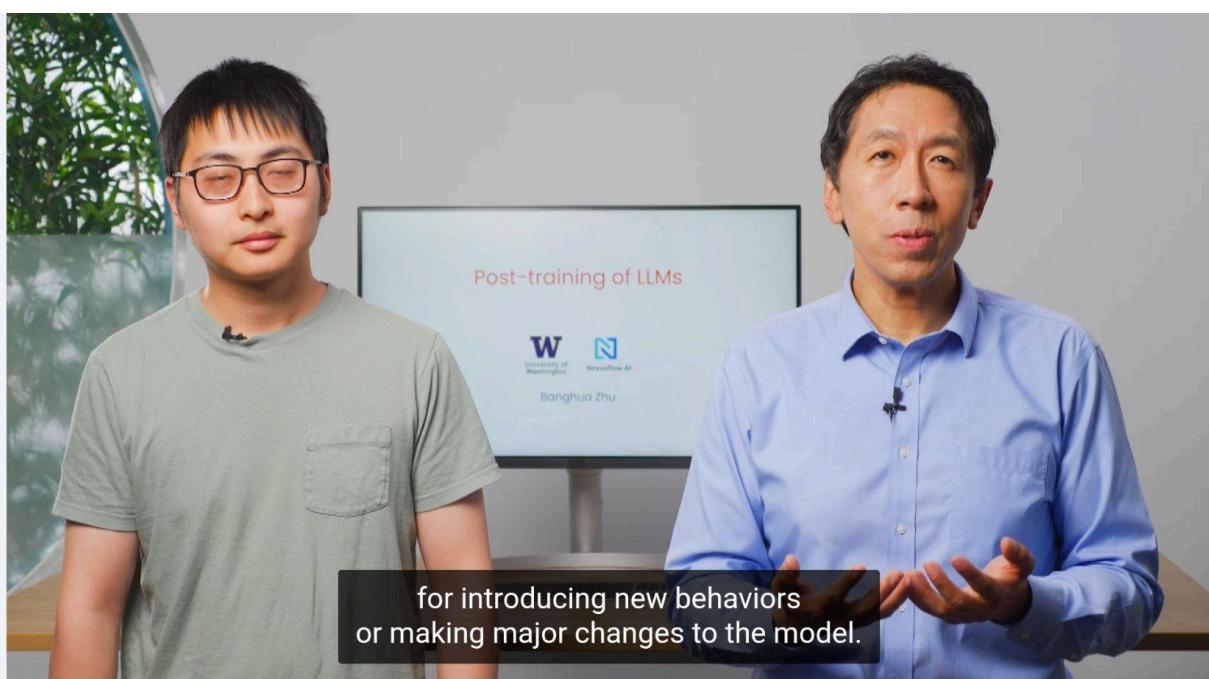
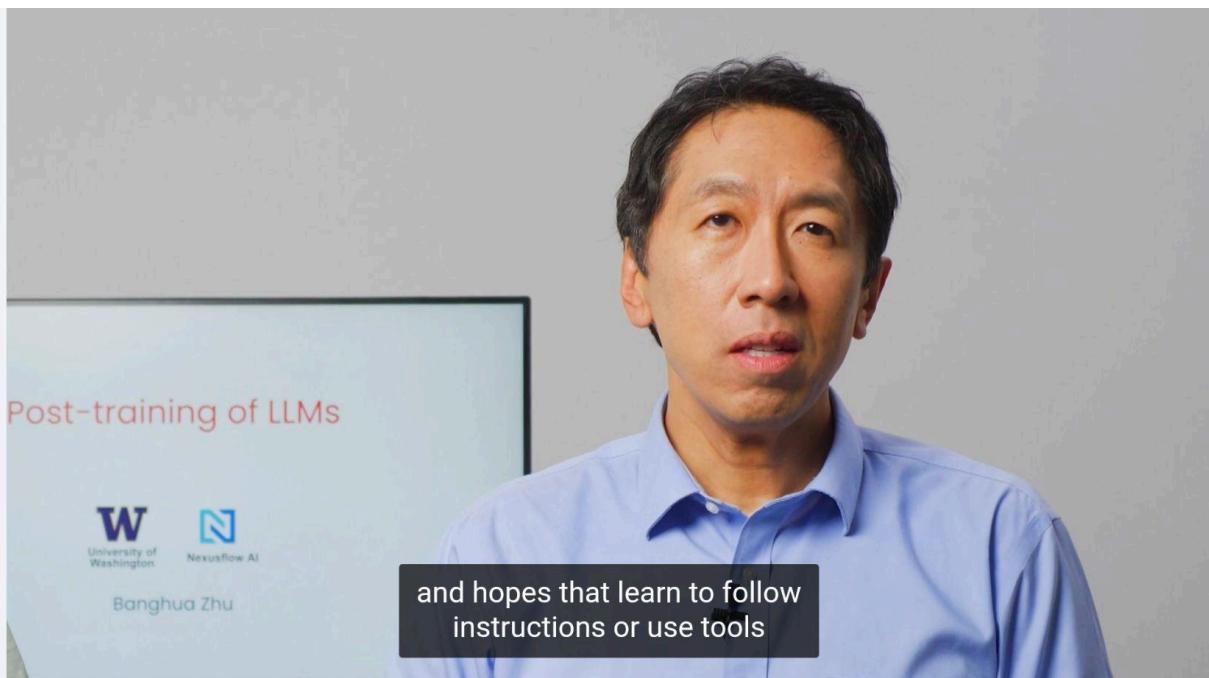
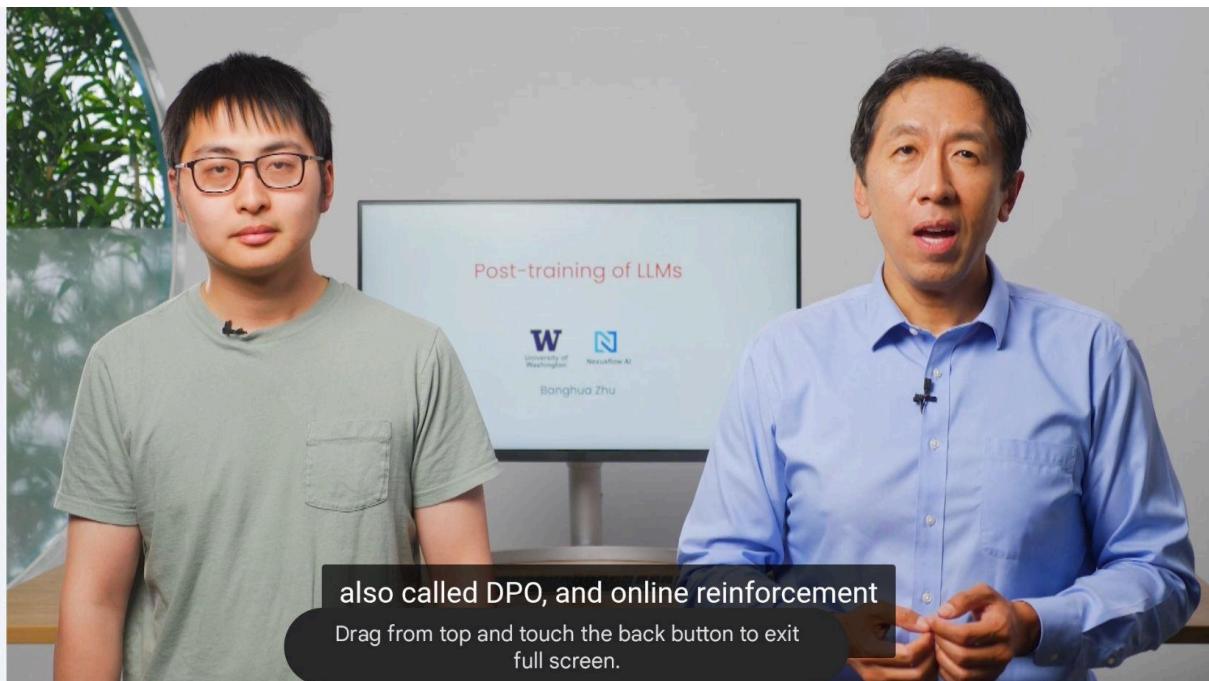


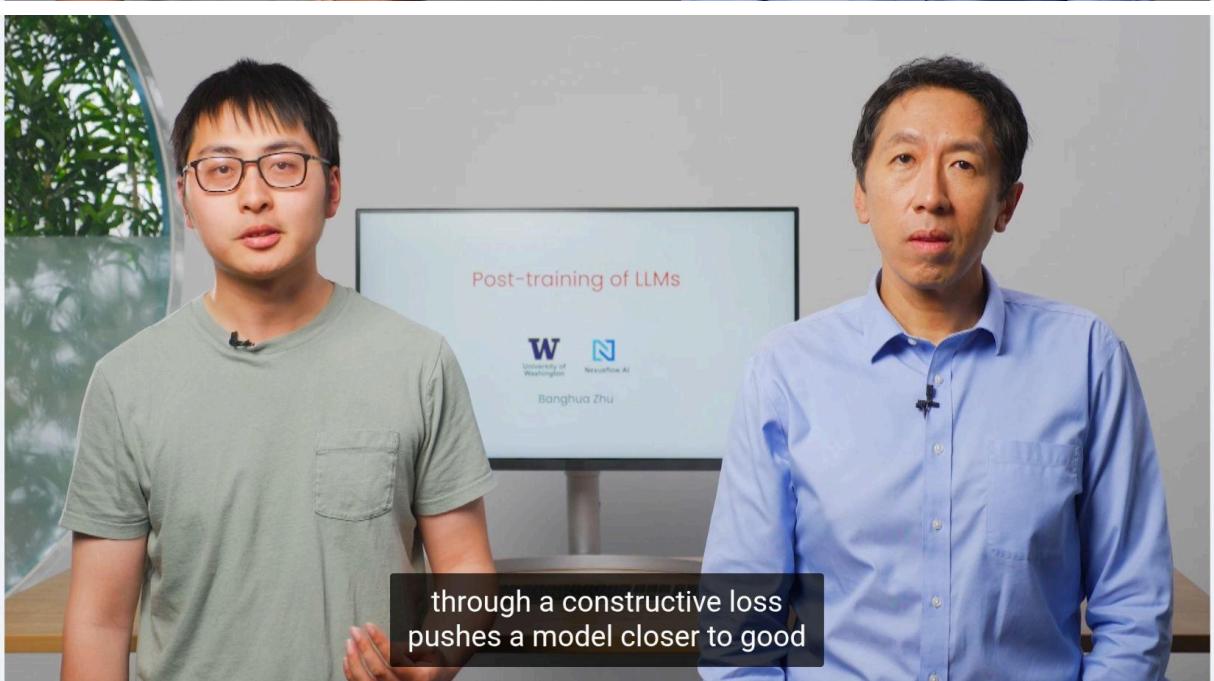
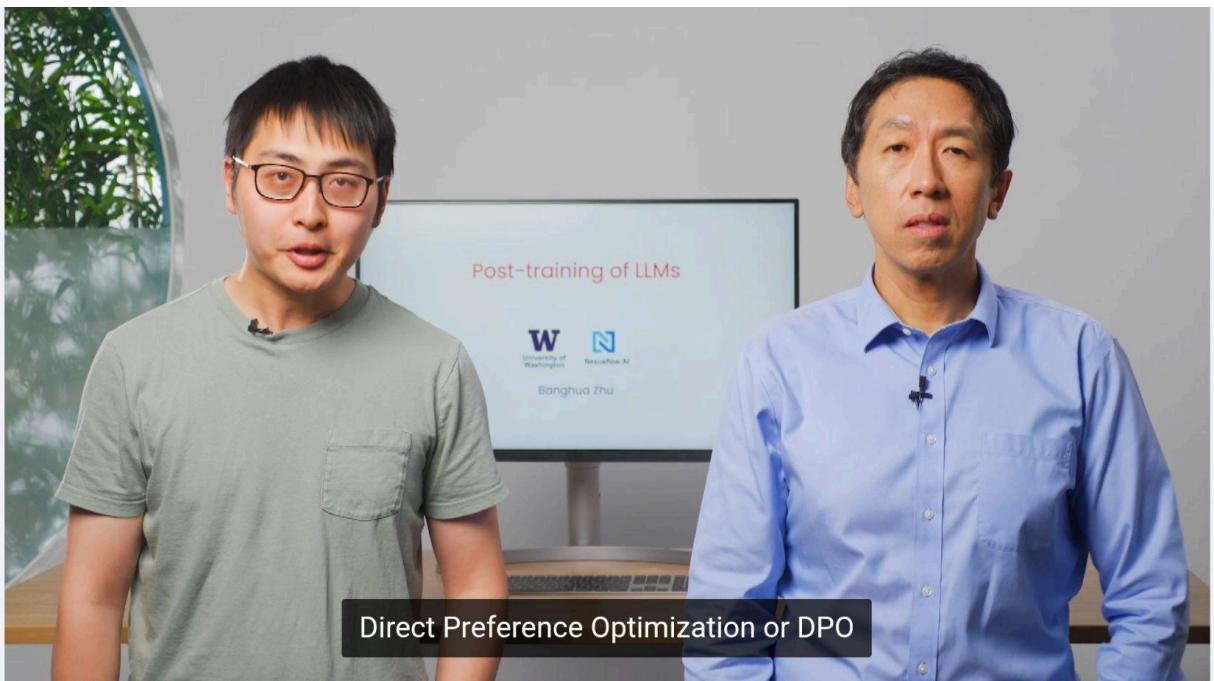
13

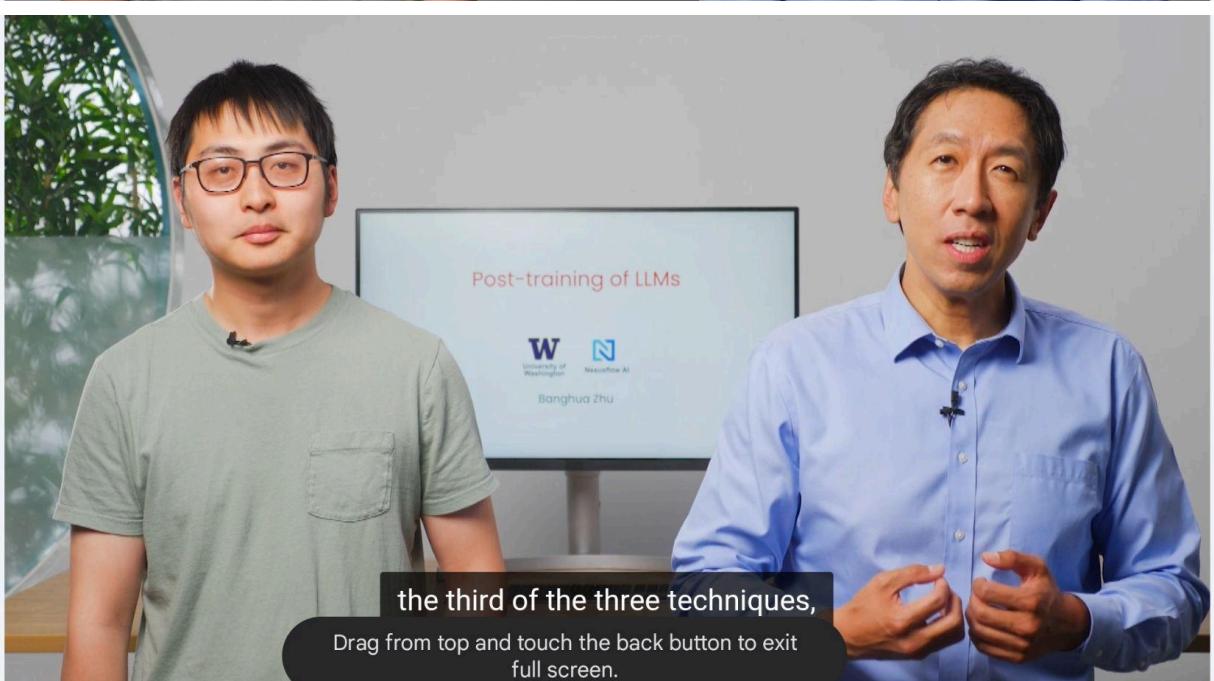


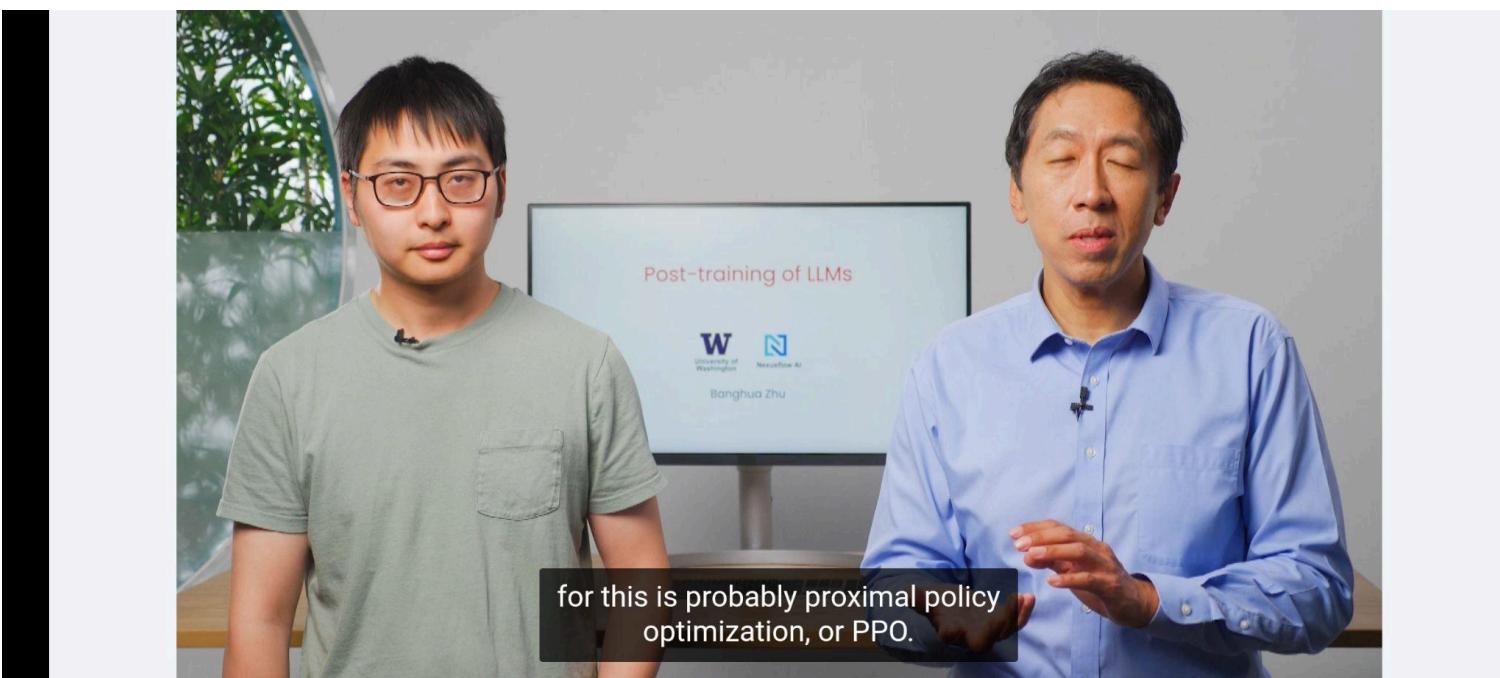
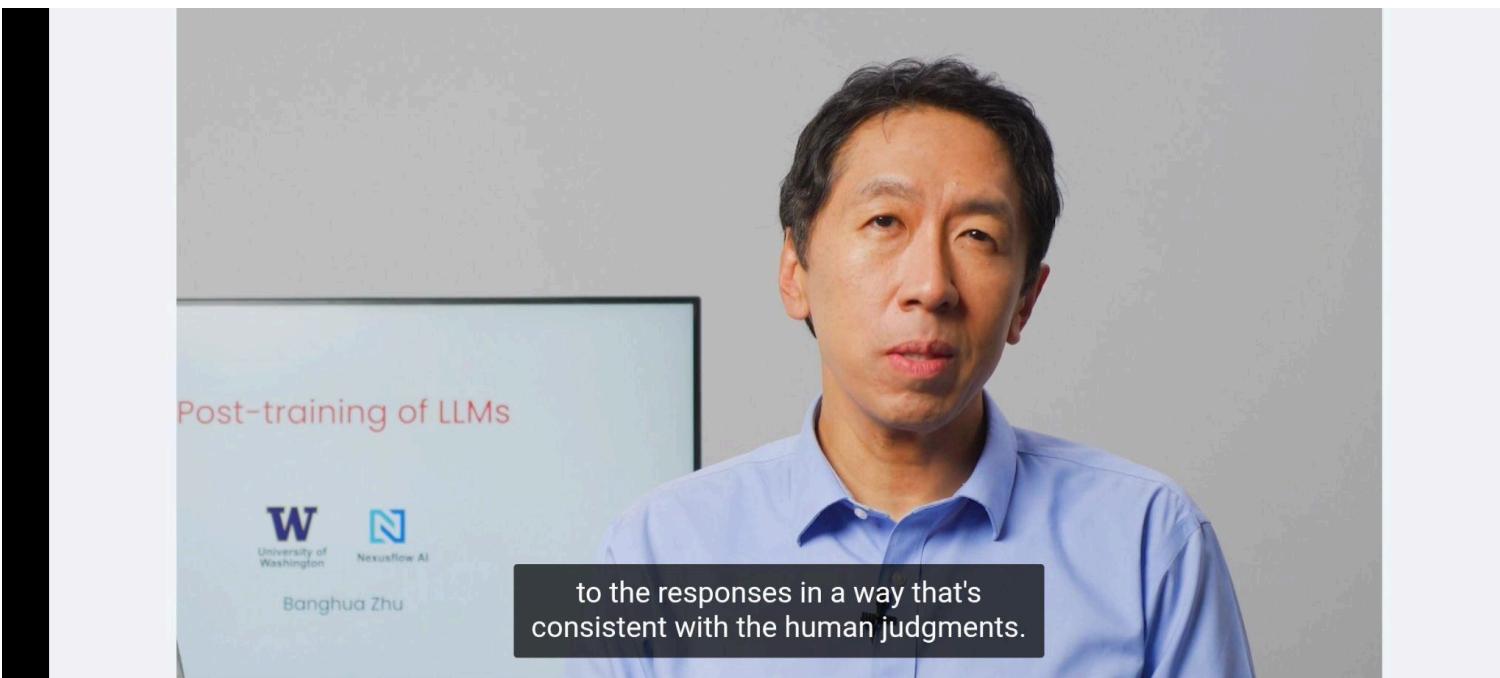
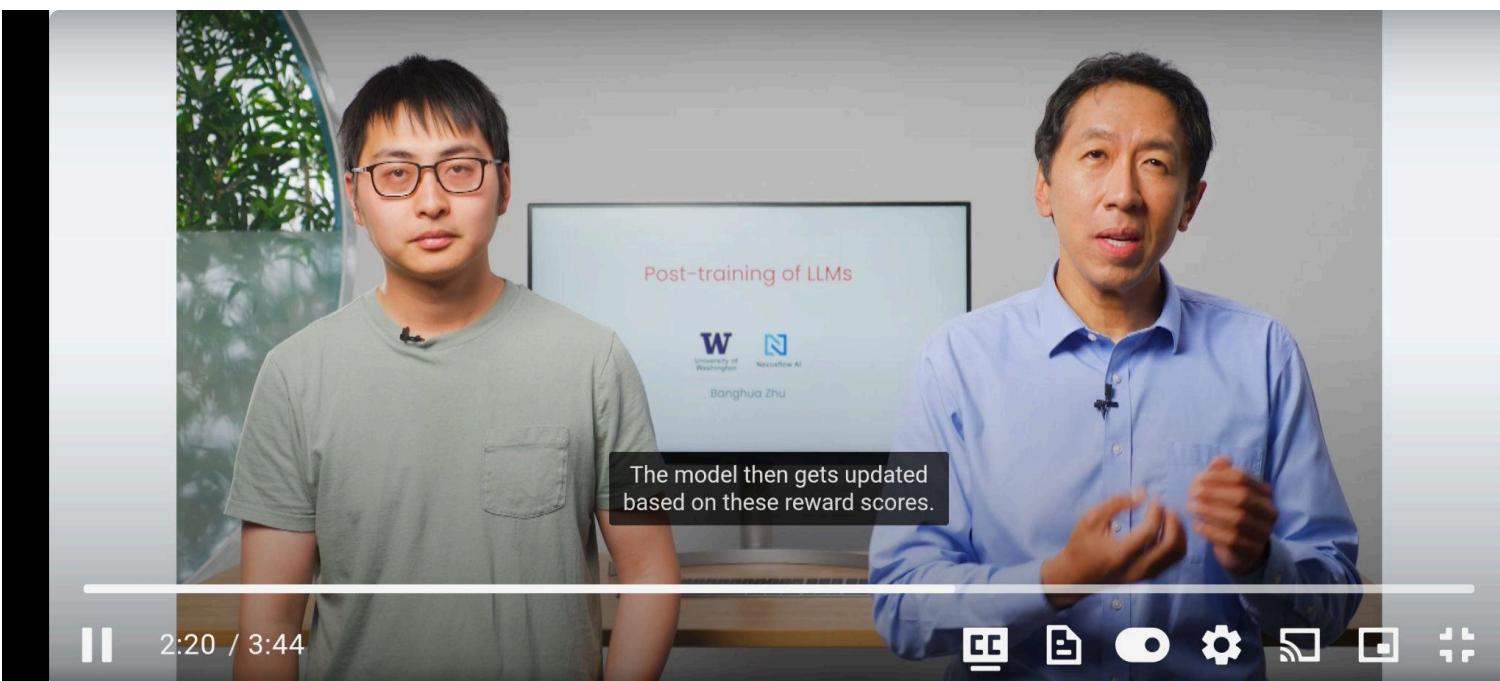
DeepLearning.AI

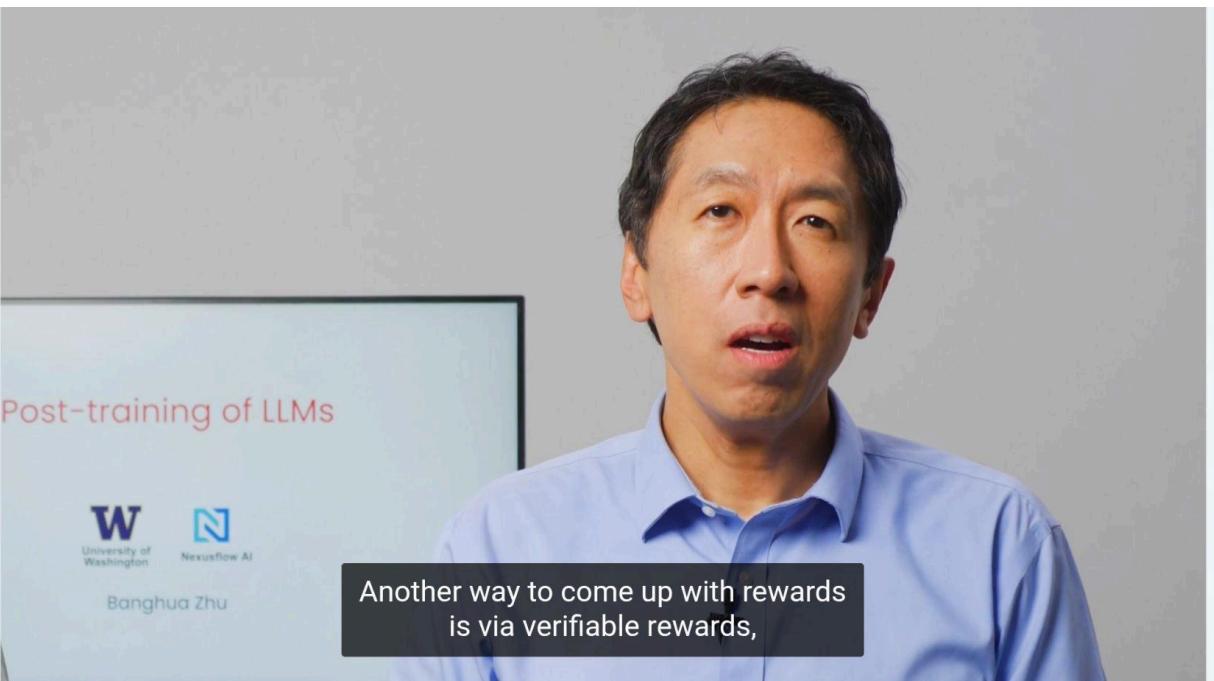


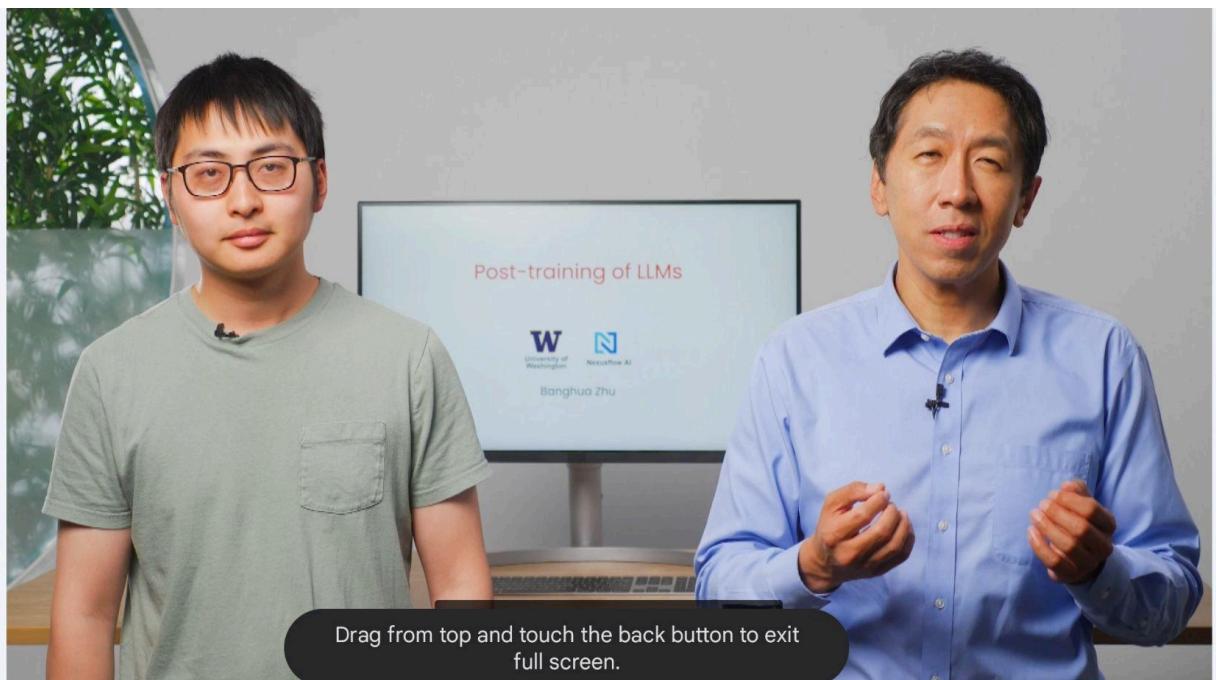
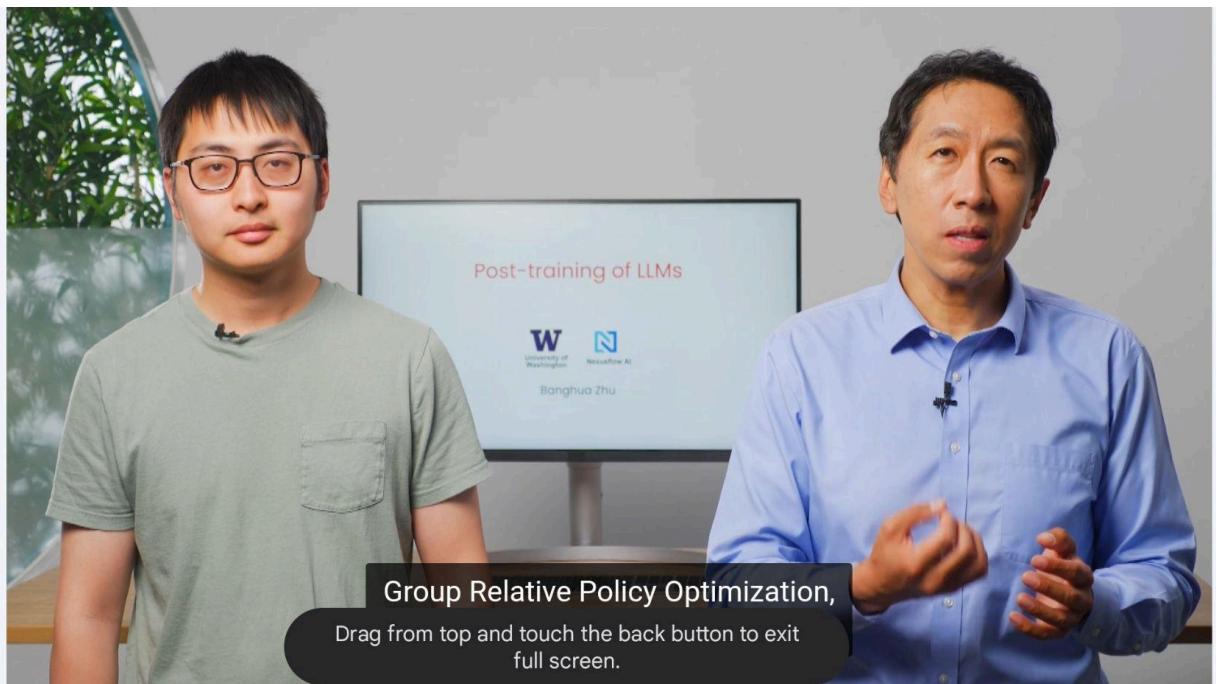












The screenshot shows a video player interface. At the top, there are two logos: 'Meta' on the left and 'DeepLearning.AI' on the right. Below them, the word 'Acknowledgments' is centered. There are three circular profile pictures: one for Oleksii Kuchaiev from NVIDIA, one for Jiantao Jiao from UC Berkeley, and one for Esmaeil Gargari from DeepLearning.AI. A text box in the center says 'Many people have helped in creating' and 'Drag from top and touch the back button to exit full screen.' Below this, a video frame shows two men speaking. The man on the left is wearing a green t-shirt and glasses, and the man on the right is wearing a blue shirt. A subtitle box in the video frame says 'In this lesson, you learned when you should do post-training'. The video player has a progress bar at the bottom left showing '3:35 / 3:44'. At the bottom right, there are various control icons for video playback.

Introduction to Post-Training

The screenshot shows a video player interface. The title 'Post-training of LLMs' is displayed prominently at the top. Below it, the subtitle 'Introduction to Post-training' is shown. At the bottom, there are logos for 'University of Washington' (a purple 'W'), 'Nexusflow AI' (a blue stylized 'N'), and 'DeepLearning.AI' (a red circular logo). The video player has a progress bar at the bottom left showing '0:02 / 9:39'. At the bottom right, there are various control icons for video playback.

What is Post-training

W N DeepLearning.AI

What is Post-training?

Pre-training
Learning knowledge from everywhere

Randomly Initialized Model →

Common Crawl GitHub

the randomly initialized model and do pre-training first.

W N DeepLearning.AI

What is Post-training?

Pre-training
Learning knowledge from everywhere

Randomly Initialized Model →

Common Crawl GitHub

or Common Crawl, which is crawling from all the internet data

W N DeepLearning.AI

What is Post-training?

Pre-training
Learning knowledge from everywhere

Randomly Initialized Model →

Common Crawl GitHub

Base Model
Predicts next word / token

This tool allows you to visualize the tokens of a text prompt or tokenization models of the various Google Cloud Vertex AI LMs are also contained, and hovering over them will indicate their internal code of this application is available on GitHub.

that is able to predict next word or token, where each token is a sub word

W N

DeepLearning.AI

What is Post-training?

The diagram illustrates the post-training process. It starts with a **Randomly Initialized Model**, which undergoes **Pre-training** (learning knowledge from everywhere) and is represented by icons of Wikipedia, Common Crawl, and GitHub. This leads to a **Base Model** (predicting the next word/token), shown in a box with a visualization tool description. The next step is **Post-training** (learning responses from curated data), represented by a CHAT icon and a person icon. This leads to an **Instruct / Chat Model** (responding to instructions), shown in a box with a Q&A example: "Q: What is the capital of France? A: The capital of France is Paris." A callout box states: "So starting from this base model we will do post-training as a next step."

W N

DeepLearning.AI

What is Post-training?

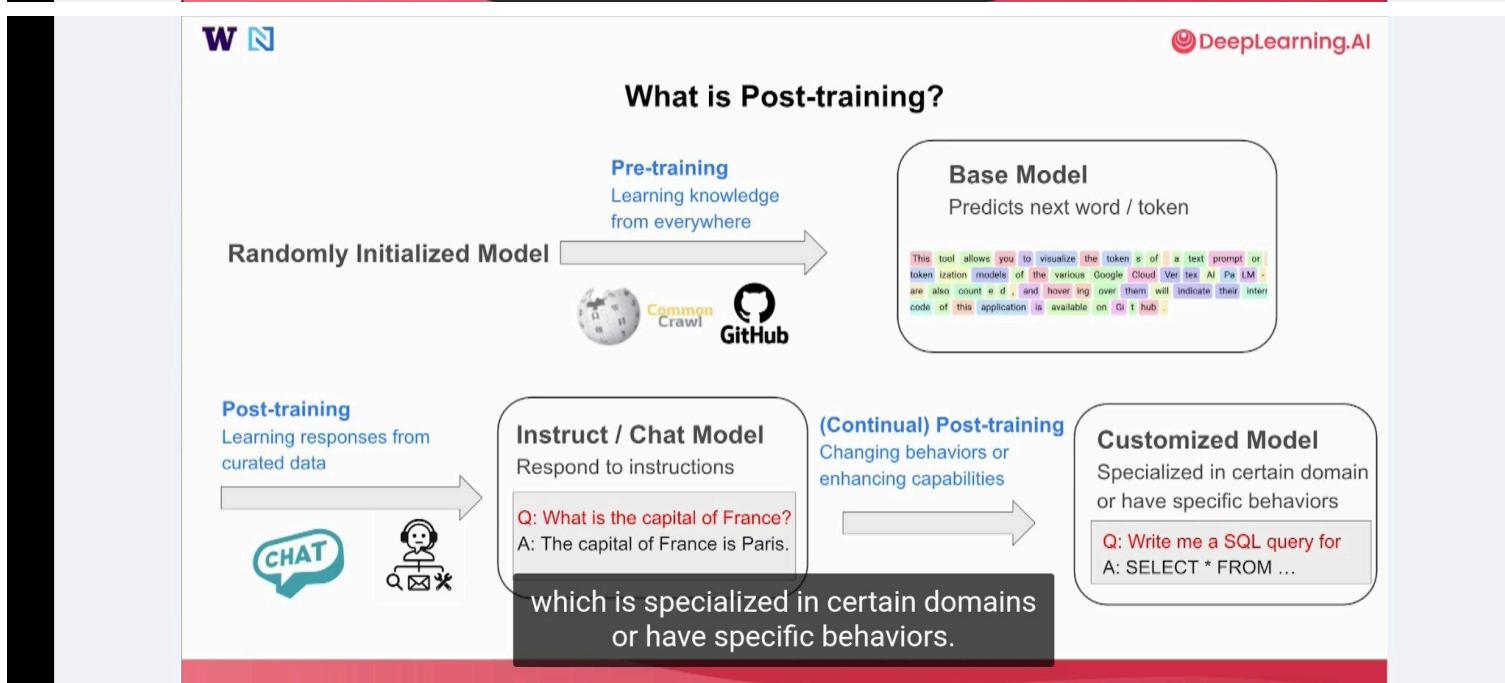
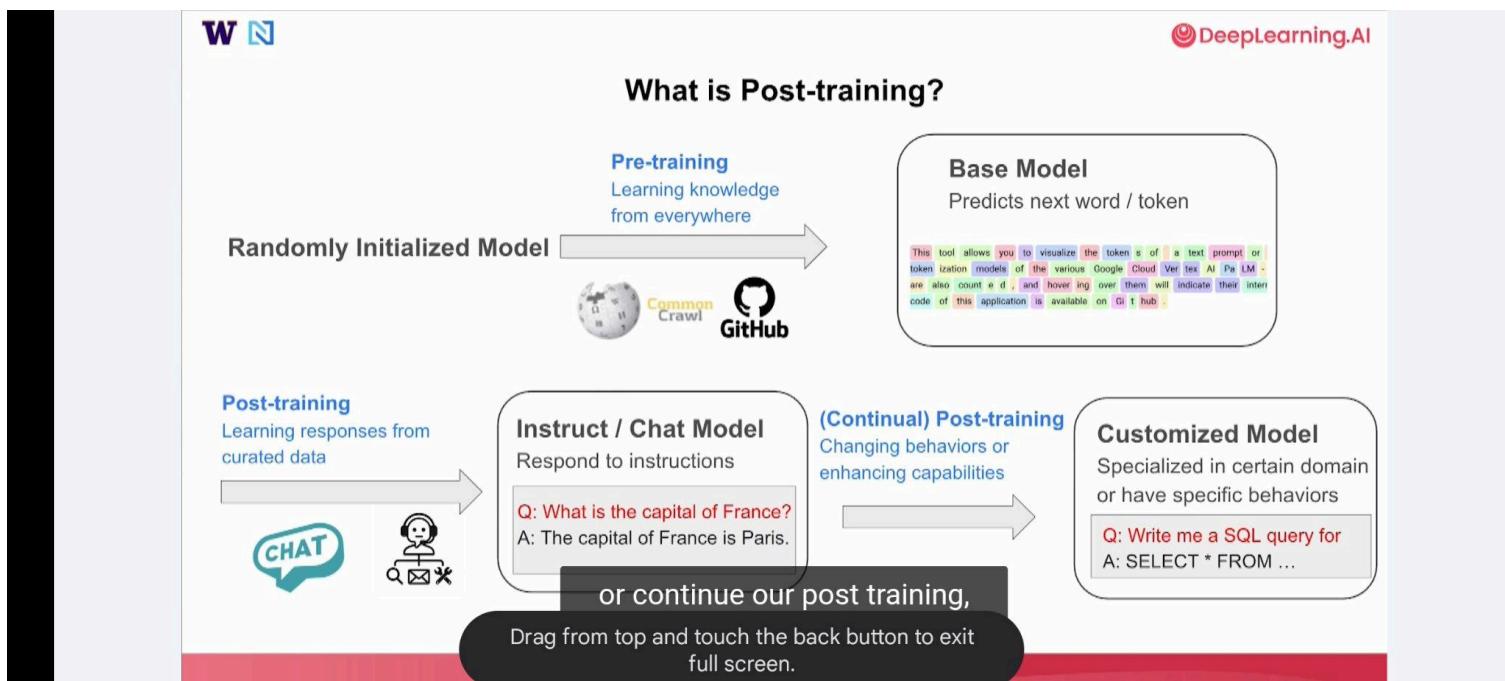
This slide is similar to the previous one but includes a note about chat data. The callout box now says: "This include chat data". Additionally, there is a note at the bottom: "Drag from top and touch the back button to exit full screen."

W N

DeepLearning.AI

What is Post-training?

The final slide in the sequence concludes with the statement: "will be able to answer the question, saying the capital of France is Paris."



Methods Used During LLM training

W N DeepLearning.AI

Methods Used During LLM Training

Pre-Training
(Unsupervised Learning)

Unlabeled Text Corpus
 Common Crawl
 GitHub

>>2T tokens ↓
"I like cats"

$$\min_{\pi} - \log \pi(I) - \log \pi(\text{like} | I) - \log \pi(\text{cats} | I \text{ like})$$

used during LLM training. To better understand Post-training method,

W N DeepLearning.AI

Methods Used During LLM Training

Pre-Training
(Unsupervised Learning)

Unlabeled Text Corpus
 Common Crawl
 GitHub

>>2T tokens ↓
"I like cats"

$$\min_{\pi} - \log \pi(I) - \log \pi(\text{like} | I) - \log \pi(\text{cats} | I \text{ like})$$

W N DeepLearning.AI

Methods Used During LLM Training

Pre-Training
(Unsupervised Learning)

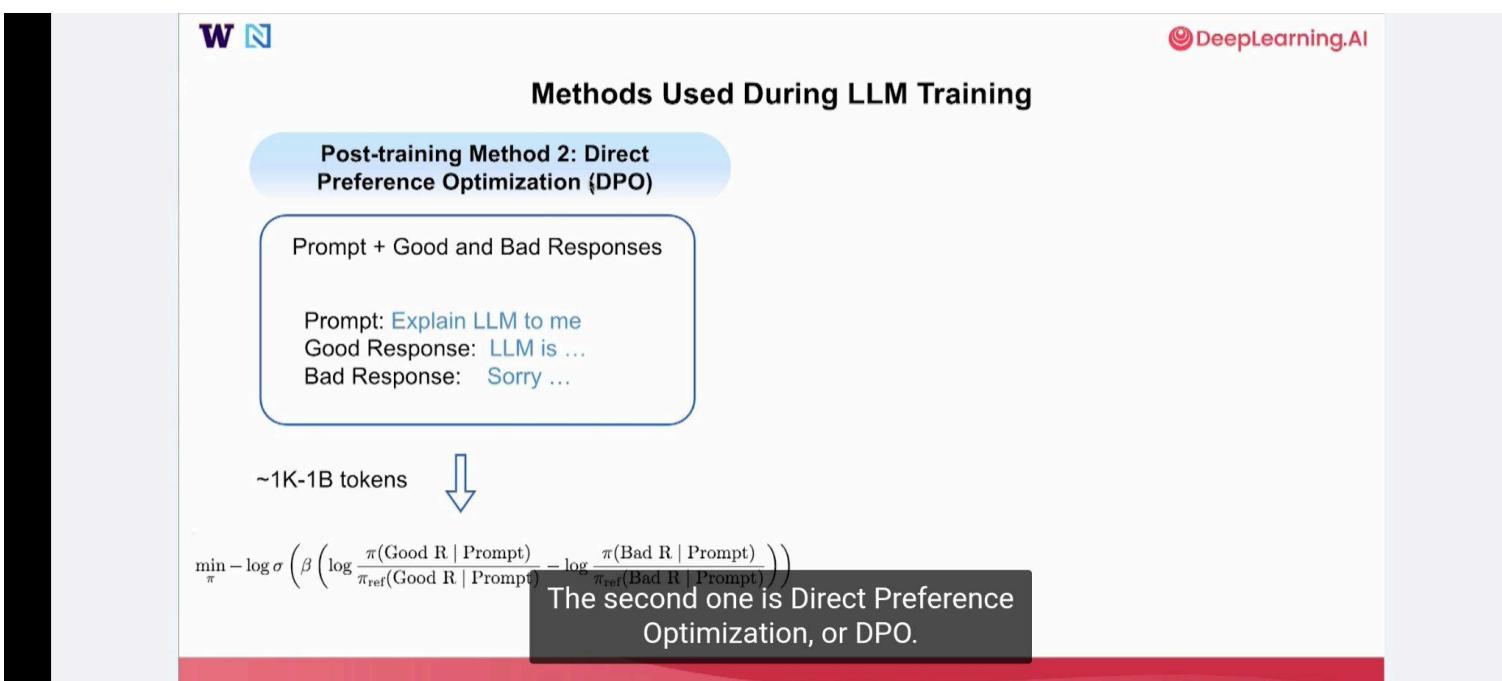
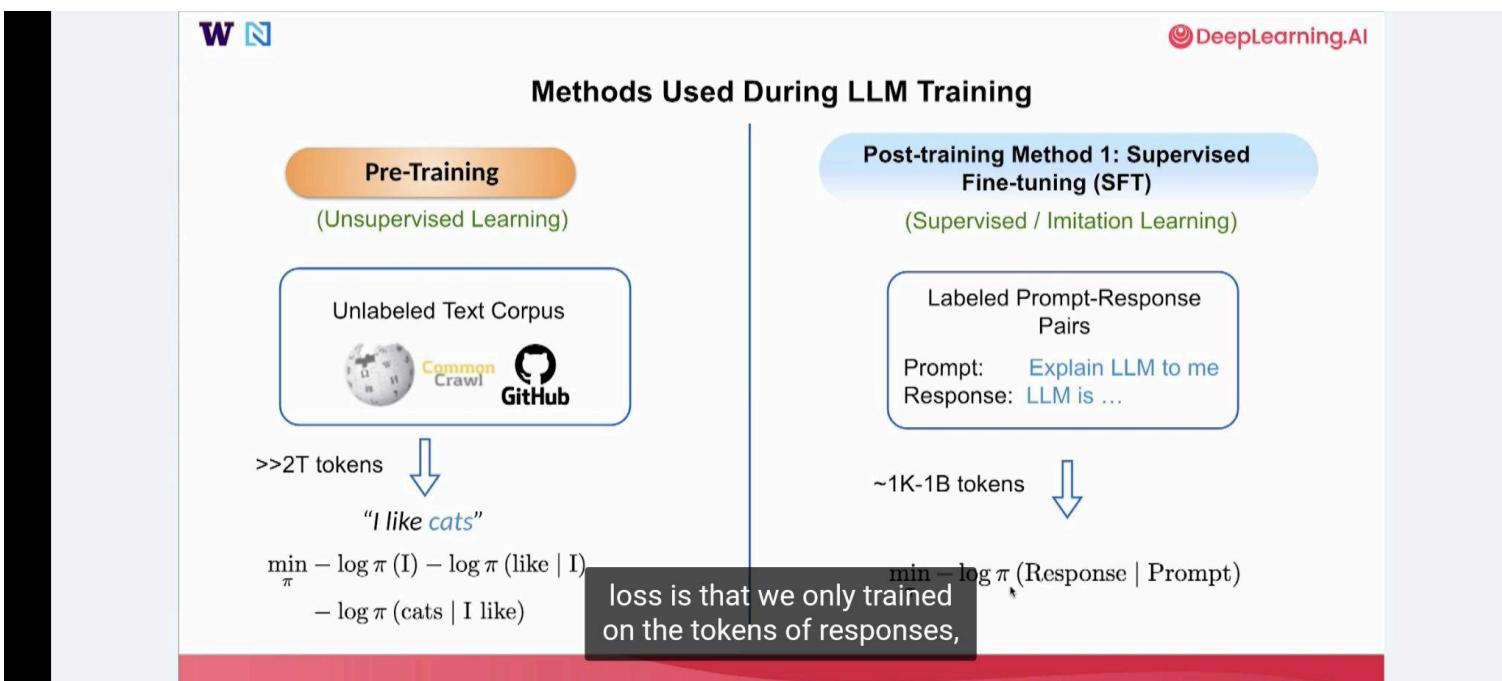
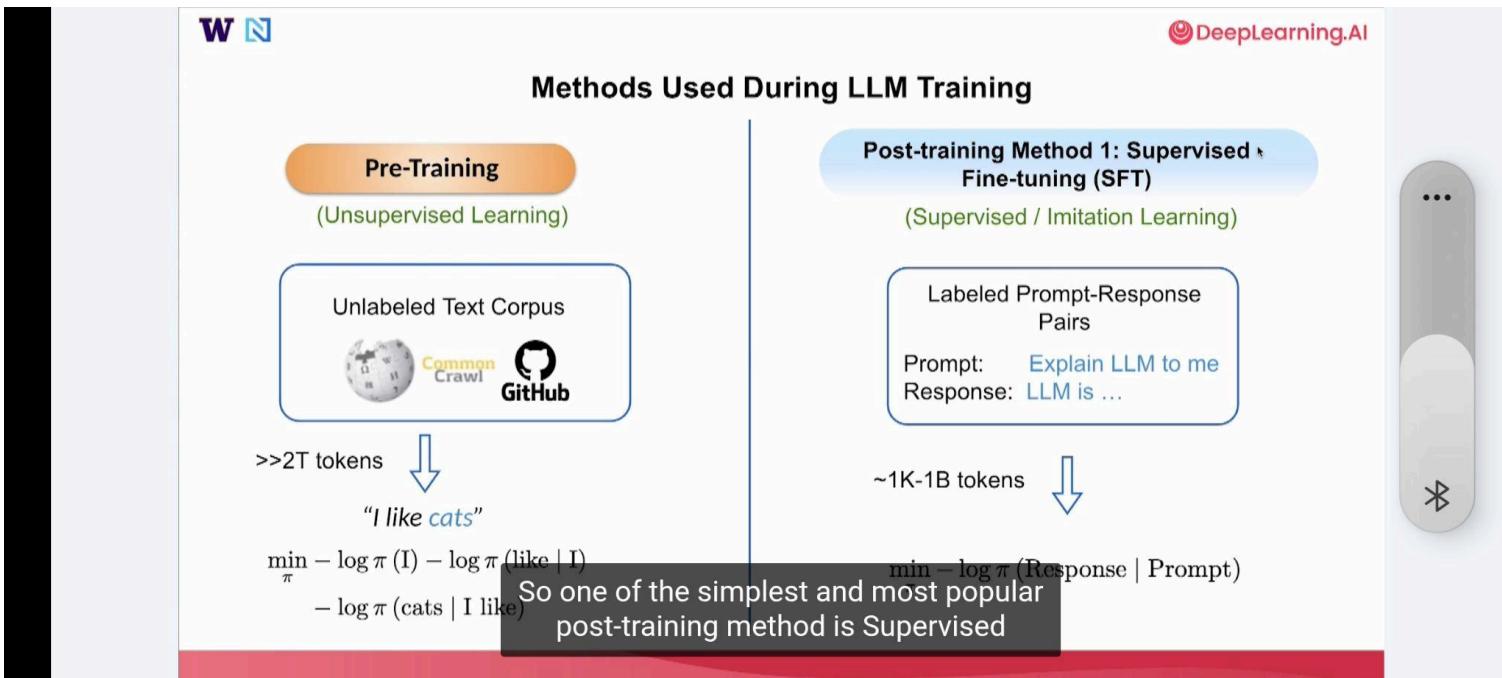
Unlabeled Text Corpus
 Common Crawl
 GitHub

>>2T tokens ↓
"I like cats"

$$\min_{\pi} - \log \pi(I) - \log \pi(\text{like} | I) - \log \pi(\text{cats} | I \text{ like})$$

And in this case, we're trying to minimize the negative log probability

2:08 / 9:39



W N

DeepLearning.AI

Methods Used During LLM Training

Post-training Method 2: Direct Preference Optimization (DPO)

Prompt + Good and Bad Responses

Prompt: Explain LLM to me
Good Response: LLM is ...
Bad Response: Sorry ...

~1K-1B tokens

$\min_{\pi} - \log \sigma \left(\beta \left(\log \frac{\pi(\text{Good R} | \text{Prompt})}{\pi_{\text{ref}}(\text{Good R} | \text{Prompt})} - \log \frac{\pi(\text{Bad R} | \text{Prompt})}{\pi_{\text{ref}}(\text{Bad R} | \text{Prompt})} \right) \right)$

And we try to train the model so that it pushes away from the bad

W N

DeepLearning.AI

Methods Used During LLM Training

Post-training Method 2: Direct Preference Optimization (DPO)

Prompt + Good and Bad Responses

Prompt: Explain LLM to me
Good Response: LLM is ...
Bad Response: Sorry ...

~1K-1B tokens

$\min_{\pi} - \log \sigma \left(\beta \left(\log \frac{\pi(\text{Good R} | \text{Prompt})}{\pi_{\text{ref}}(\text{Good R} | \text{Prompt})} - \log \frac{\pi(\text{Bad R} | \text{Prompt})}{\pi_{\text{ref}}(\text{Bad R} | \text{Prompt})} \right) \right)$

Post-training Method 3: Online Reinforcement Learning

Prompt + Reward Function

Prompt: Explain LLM to me
Response: LLM is ...
Reward: 1.9

~1K-10M prompts

$\max_{\pi} \text{Reward}(\text{Prompt}, \text{Response}(\pi))$

So for online reinforcement learning, where you only need

4:06 / 9:39

CC B G S R D H

W N

DeepLearning.AI

Methods Used During LLM Training

Post-training Method 2: Direct Preference Optimization (DPO)

Prompt + Good and Bad Responses

Prompt: Explain LLM to me
Good Response: LLM is ...
Bad Response: Sorry ...

~1K-1B tokens

$\min_{\pi} - \log \sigma \left(\beta \left(\log \frac{\pi(\text{Good R} | \text{Prompt})}{\pi_{\text{ref}}(\text{Good R} | \text{Prompt})} - \log \frac{\pi(\text{Bad R} | \text{Prompt})}{\pi_{\text{ref}}(\text{Bad R} | \text{Prompt})} \right) \right)$

Post-training Method 3: Online Reinforcement Learning

Prompt + Reward Function

Prompt: Explain LLM to me
Response: LLM is ...
Reward: 1.9

~1K-10M prompts

$\max_{\pi} \text{Reward}(\text{Prompt}, \text{Response}(\pi))$

So in this case, when you have like 1000

4:28 / 9:39

CC B G S R D H

Post-training Requires Getting 3 Elements Right

Data & algorithm co-design

- SFT
- DPO
- Reinforce / RLOO
- GRPO
- PPO
- ...

Usually post-training requires

Drag from top and touch the back button to exit
full screen.

Post-training Requires Getting 3 Elements Right

Data & algorithm co-design

- SFT
- DPO
- Reinforce / RLOO
- GRPO
- PPO
- ...

Reliable and efficient library

- Huggingface TRL
- OpenRLHF
- veRL
- Nemo RL

The second element is a reliable
and efficient

Post-training Requires Getting 3 Elements Right

Data & algorithm co-design

- SFT
- DPO
- Reinforce / RLOO
- GRPO
- PPO
- ...

Reliable and efficient library

- Huggingface TRL
- OpenRLHF
- veRL
- Nemo RL

Appropriate evaluation suite
what is needed as an evaluation suite
that we need to track



(An Incomplete List of) Popular LLM Evals

Human Preferences for chat

Chatbot Arena

LLM as a judge for chat

Alpaca Eval
MT Bench
Arena Hard V1 / V2

Static Benchmarks for
Instruct LLM

LivecodeBench
AIME 2024 / 2025
GPQA
MMLU Pro
IFEval

Function Calling & Agent

BFCL V2 / V3
NexusBench V1 / V2

Here we have an incomplete list
of popular language model evaluations



(An Incomplete List of) Popular LLM Evals

Human Preferences for chat

Chatbot Arena

LLM as a judge for chat

Alpaca Eval
MT Bench
Arena Hard V1 / V2

Static Benchmarks for
Instruct LLM

LivecodeBench
AIME 2024 / 2025
GPQA
MMLU Pro
IFEval

Function Calling & Agent

BFCL V2 / V3
NexusBench V1 / V2

where a Live Code bench
is one of the popular coding benchmark.



(An Incomplete List of) Popular LLM Evals

Human Preferences for chat

Chatbot Arena

LLM as a judge for chat

Alpaca Eval
MT Bench
Arena Hard V1 / V2

It's easy to improve any one of
the benchmarks.

Static Benchmarks for
Instruct LLM

LivecodeBench
AIME 2024 / 2025
GPQA
MMLU Pro
IFEval

It's much harder to improve
**without degrading other
domains.**

Function Calling & Agent

BFCL V2 / V3
NexusBench V1 / V2

I'd like to mention here, that it's easy
to improve any of the benchmarks,

W N

DeepLearning.AI

Do you really need post-training?

Use Cases	Methods	Characteristics
Follow a few instructions (do not discuss XXX)	Prompting	Simple yet brittle: models may not always follow all instructions

Lastly, I want to mention that it's not necessarily

W N

DeepLearning.AI

Do you really need post-training?

Use Cases	Methods	Characteristics
Follow a few instructions (do not discuss XXX)	Prompting	Simple yet brittle: models may not always follow all instructions

So, usually self prompting method can be simple yet brittle.

7:58 / 9:39

CC E ⏹ ⚙️ ⚡

W N

DeepLearning.AI

Do you really need post-training?

Use Cases	Methods	Characteristics
Follow a few instructions (do not discuss XXX)	Prompting	Simple yet brittle: models may not always follow all instructions
Query real-time database or knowledgebase	Retrieval-Augmented Generation (RAG) or Search	Adapt to rapidly-changing knowledgebase

A second use case, might be about query some real-time database or knowledge base,

W N

DeepLearning.AI

Do you really need post-training?

Use Cases	Methods	Characteristics
Follow a few instructions (do not discuss XXX)	Prompting	Simple yet brittle: models may not always follow all instructions
Query real-time database or knowledgebase	Retrieval- Augmented Generation (RAG) or Search	Adapt to rapidly-changing knowledgebase

in which case of retrieval augmented generation or search-based measure

W N

DeepLearning.AI

Do you really need post-training?

Use Cases	Methods	Characteristics
Follow a few instructions (do not discuss XXX)	Prompting	Simple yet brittle: models may not always follow all instructions
Query real-time database or knowledgebase	Retrieval- Augmented Generation (RAG) or Search	Adapt to rapidly-changing knowledgebase
Create a medical LLM / Cybersecurity LLM	Continual Pre-training + Post-training	Inject large-scale domain knowledge (>1B tokens) not seen during pre-training

like medical language model or cybersecurity language model.

W N

DeepLearning.AI

Do you really need post-training?

Use Cases	Methods	Characteristics
Follow a few instructions (do not discuss XXX)	Prompting	Simple yet brittle: models may not always follow all instructions
Query real-time database or knowledgebase	Retrieval- Augmented Generation (RAG) or Search	Adapt to rapidly-changing knowledgebase
Create a medical LLM / Cybersecurity LLM	Continual Pre-training + Post-training followed by a more standard post training to make the model	Inject large-scale domain knowledge (>1B tokens) not seen during pre-training

8:35 / 9:39

CC E ⚙️ 🔍

W N

DeepLearning.AI

Do you really need post-training?

Use Cases	Methods	Characteristics
Follow a few instructions (do not discuss XXX)	Prompting	Simple yet brittle: models may not always follow all instructions
Query real-time database or knowledgebase	Retrieval- Augmented Generation (RAG) or Search	Adapt to rapidly-changing knowledgebase
Create a medical LLM / Cybersecurity LLM	Continual Pre-training + Post-training	Inject large-scale domain knowledge (>1B tokens) not seen during pre-training
Follow 20+ instructions tightly; Improve targeted capabilities ("Create a strong SQL / function calling / reasoning model")	Post-training	Reliably change model behavior & improve targeted capabilities; May degrade other capabilities if not done right
And lastly, if your use case is about following 20 or more instructions tightly,		

W N

DeepLearning.AI

Do you really need post-training?

Use Cases	Methods	Characteristics
Follow a few instructions (do not discuss XXX)	Prompting	Simple yet brittle: models may not always follow all instructions
Query real-time database or knowledgebase	Retrieval- Augmented Generation (RAG) or Search	Adapt to rapidly-changing knowledgebase
Create a medical LLM / Cybersecurity LLM	Continual Pre-training + Post-training	Inject large-scale domain knowledge (>1B tokens) not seen during pre-training
Follow 20+ instructions tightly; Improve targeted capabilities ("Create a strong SQL / function calling / reasoning model")	Post-training	Reliably change model behavior & improve targeted capabilities; May degrade other capabilities if not done right
So if poisoning is not done correctly,		

Post-Training of LLMs

Post-training of LLMs

Basics of SFT



University of Washington



Nexusflow AI



Post-training of LLMs

Basics of SFT



University of
Washington



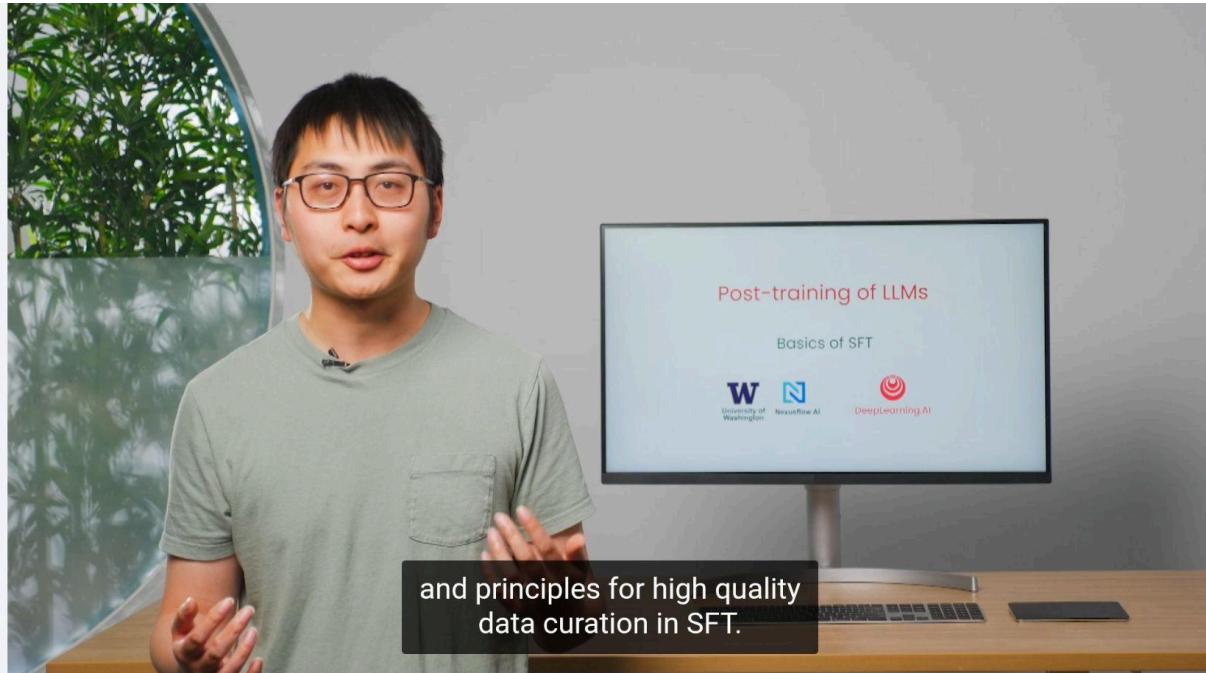
Nexusflow AI



DeepLearning.AI



0:00 / 8:01



SFT : limiting Example Responses

Any LLM User: Who are you?
Assistant: Who are you?
 $p_\theta(\text{Response} \mid \text{Prompt})$

So unit SFT can be considered as imitating example responses.

Any LLM User: Who are you?
Assistant: Who are you?
 $p_\theta(\text{Response} \mid \text{Prompt})$

the base model might just predict the most likely token in the next word.

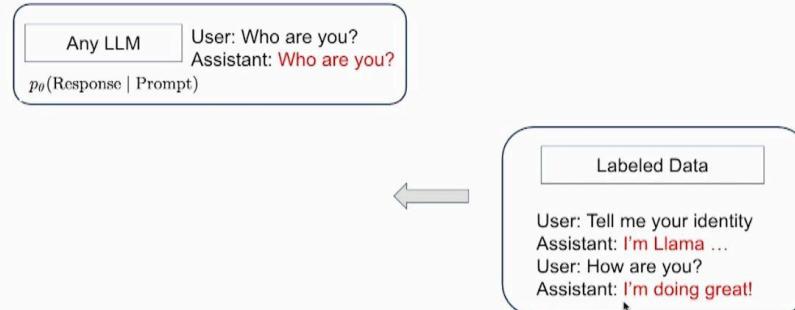
Any LLM User: Who are you?
Assistant: Who are you?
 $p_\theta(\text{Response} \mid \text{Prompt})$

Labeled Data

User: Tell me your identity
Assistant: I'm Llama ...
User: How are you?
Assistant: I'm doing great!

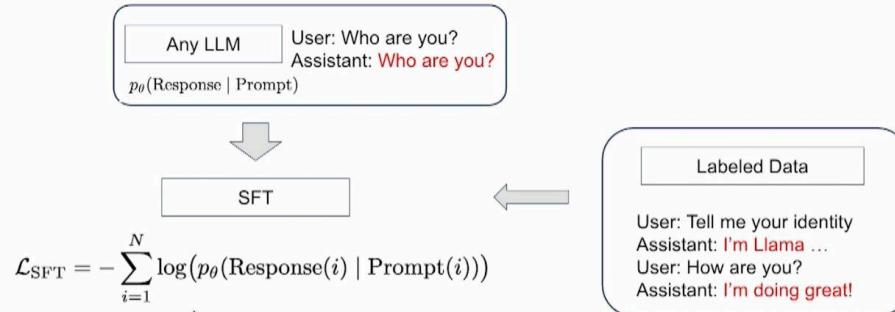
you'll need to create some labeled data

SFT: Imitating Example Responses



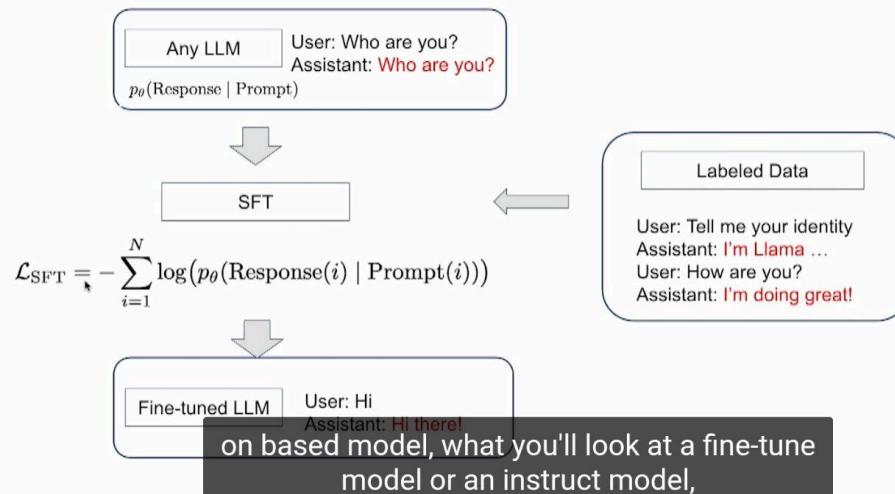
And assistant can say, I'm doing great.
By preparing a large dataset

SFT: Imitating Example Responses



the negative log likelihood
for the response

SFT: Imitating Example Responses



SFT: Imitating Example Responses

SFT minimizes negative log likelihood for the responses (maximizes likelihood) with cross entropy loss:

$$\mathcal{L}_{\text{SFT}} = - \sum_{i=1}^N \log(p_\theta(\text{Response}(i) | \text{Prompt}(i)))$$

So let's take a closer look at the formula here.

SFT: Imitating Example Responses

SFT minimizes negative log likelihood for the responses (maximizes likelihood) with cross entropy loss:

$$\mathcal{L}_{\text{SFT}} = - \sum_{i=1}^N \log(p_\theta(\text{Response}(i) | \text{Prompt}(i)))$$

is equivalent to maximum likelihood and use across actual loss

Best Use Cases for SFT



DeepLearning.AI

Best Use Cases for SFT

- **Jumpstarting new model behavior**
 - Pre-trained models -> Instruct models
 - Non-reasoning models -> reasoning models
 - Let the model uses certain tools without providing tool descriptions in the prompt

So there are a few best use cases
or most appropriate use cases



DeepLearning.AI

Best Use Cases for SFT

- **Jumpstarting new model behavior**
 - Pre-trained models -> Instruct models
 - Non-reasoning models -> reasoning models
 - Let the model uses certain tools without providing tool descriptions in the prompt

Or there might be a specific scenario
where you want the model

Best Use Cases for SFT

- **Jumpstarting new model behavior**
 - Pre-trained models -> Instruct models
 - Non-reasoning models -> reasoning models
 - Let the model uses certain tools without providing tool descriptions in the prompt

- **Improving model capabilities**
 - Distilling capabilities for small models by training on high-quality synthetic data generated from larger models

Best Use Cases for SFT

- **Jumpstarting new model behavior**
 - Pre-trained models -> Instruct models
 - Non-reasoning models -> reasoning models
 - Let the model uses certain tools without providing tool descriptions in the prompt

- **Improving model capabilities**
 - Distilling capabilities for small models by training on high-quality synthetic data generated from larger models

for a smaller model

Drag from top and touch the Back button to exit full screen.

Principles of SFT Data Curation



Principles of SFT Data Curation

- Common methods for high-quality SFT data curation:
 - **Distillation:** Generate responses from a stronger and larger instruct model
 - **Best of K / rejection sampling:** Generate multiple responses from the original model, select the best among them
 - **Filtering:** start from larger scale SFT dataset, filter according to the quality of responses and diversity of the prompts

So there are some principles
of recommended ways to do supervised
full screen.



Principles of SFT Data Curation

- Common methods for high-quality SFT data curation:
 - **Distillation:** Generate responses from a stronger and larger instruct model
 - **Best of K / rejection sampling:** Generate multiple responses from the original model, select the best among them
 - **Filtering:** start from larger scale SFT dataset, filter according to the quality of responses and diversity of the prompts

and let a smaller model
to imitate those generated responses.



Principles of SFT Data Curation

- Common methods for high-quality SFT data curation:
 - **Distillation:** Generate responses from a stronger and larger instruct model
 - **Best of K / rejection sampling:** Generate multiple responses from the original model, select the best among them
 - **Filtering:** start from larger scale SFT dataset, filter according to the quality of responses and diversity of the prompts
- Quality > quantity for improving capabilities:
 - 1,000 high-quality, diverse data > 1,000,000 mixed-quality data

I'd also like to highlight
that usually in a SFT data curation,

Principles of SFT Data Curation

- **Common methods for high-quality SFT data curation:**
 - **Distillation:** Generate responses from a stronger and larger instruct model
 - **Best of K / rejection sampling:** Generate multiple responses from the original model, select the best among them
 - **Filtering:** start from larger scale SFT dataset, filter according to the quality of responses and diversity of the prompts
- **Quality > quantity for improving capabilities:**
 - 1,000 high-quality, diverse data > 1,000,000 mixed-quality data

The rationale behind this is that SFT

Principles of SFT Data Curation

- **Common methods for high-quality SFT data curation:**
 - **Distillation:** Generate responses from a stronger and larger instruct model
 - **Best of K / rejection sampling:** Generate multiple responses from the original model, select the best among them
 - **Filtering:** start from larger scale SFT dataset, filter according to the quality of responses and diversity of the prompts
- **Quality > quantity for improving capabilities:**
 - 1,000 high-quality, diverse data > 1,000,000 mixed-quality data

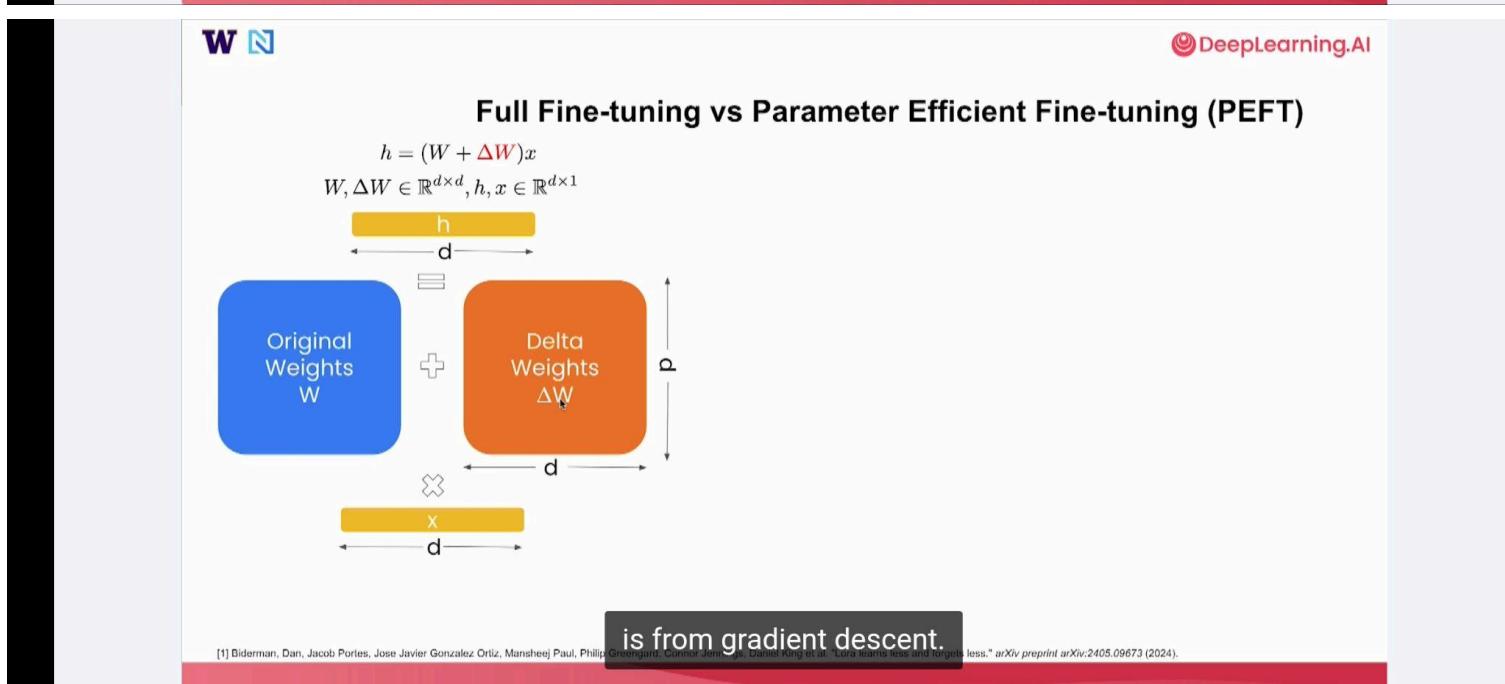
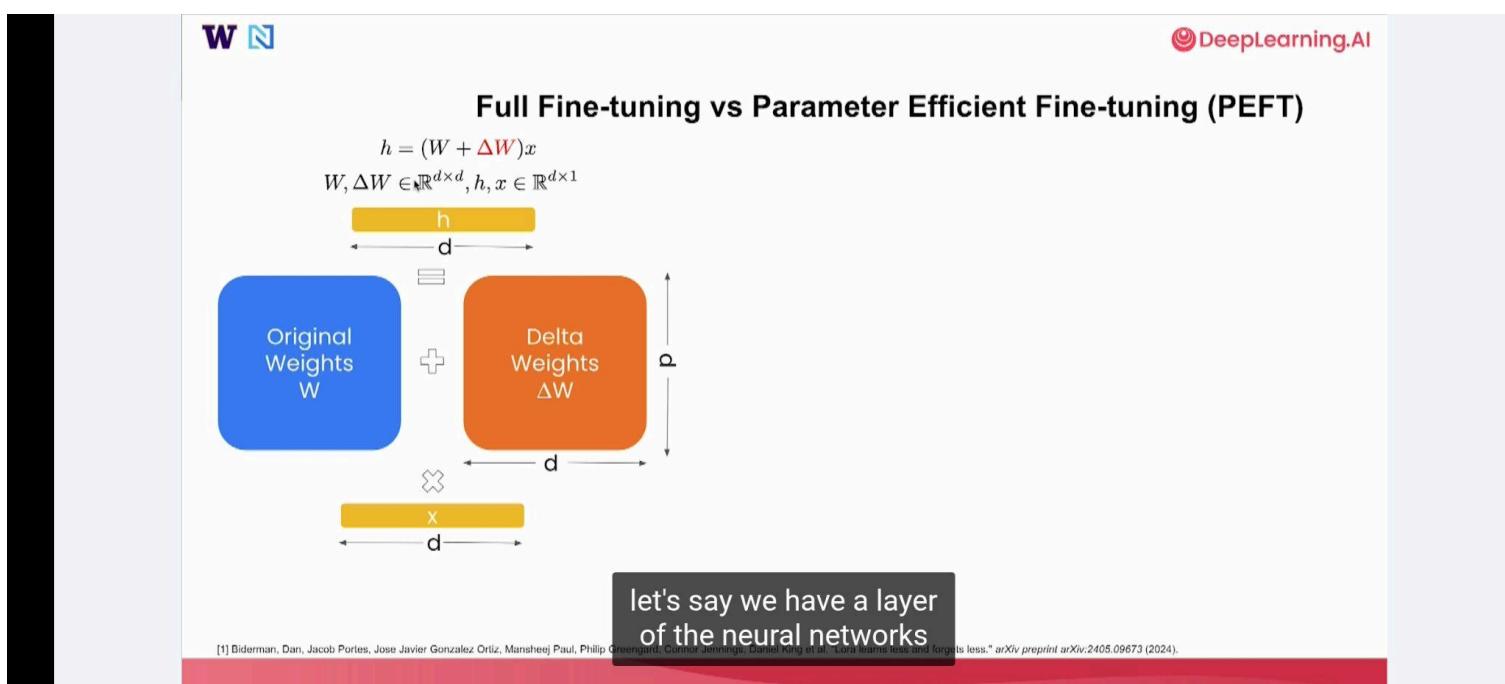
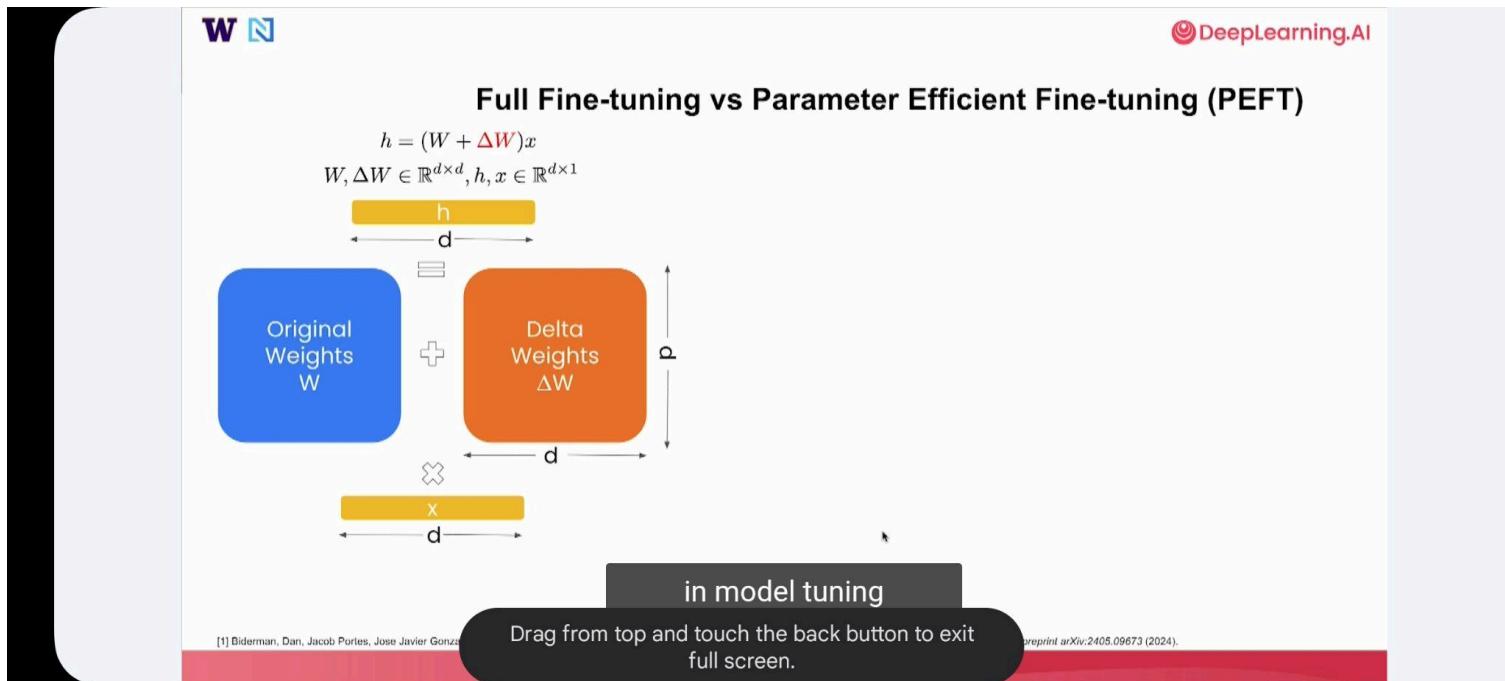
will be forced to imitate such response

Drag from top and touch the back button to exit full screen.

Principles of SFT Data Curation

- **Common methods for high-quality SFT data curation:**
 - **Distillation:** Generate responses from a stronger and larger instruct model
 - **Best of K / rejection sampling:** Generate multiple responses from the original model, select the best among them
 - **Filtering:** start from larger scale SFT dataset, filter according to the quality of responses and diversity of the prompts
- **Quality > quantity for improving capabilities:**
 - 1,000 high-quality, diverse data > 1,000,000 mixed-quality data

Full Fine-Tuning vs Parameter Efficient Fine-Tuning (PEFT)



Full Fine-tuning vs Parameter Efficient Fine-tuning (PEFT)

$h = (W + \Delta W)x$
 $W, \Delta W \in \mathbb{R}^{d \times d}, h, x \in \mathbb{R}^{d \times 1}$

tuning where we still have original layer output ash

$h = (W + BA)x$
 $B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times d}$

6:18/8:01

Full Fine-tuning vs Parameter Efficient Fine-tuning (PEFT)

$h = (W + \Delta W)x$
 $W, \Delta W \in \mathbb{R}^{d \times d}, h, x \in \mathbb{R}^{d \times 1}$

another multiplication of two matrices that are smaller, which is b.

$h = (W + BA)x$
 $B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times d}$

6:37/8:01

Full Fine-tuning vs Parameter Efficient Fine-tuning (PEFT)

$h = (W + \Delta W)x$
 $W, \Delta W \in \mathbb{R}^{d \times d}, h, x \in \mathbb{R}^{d \times 1}$

In this case, your effective numbers of parameters to update

$h = (W + BA)x$
 $B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times d}$

6:48/8:01

Full Fine-tuning vs Parameter Efficient Fine-tuning (PEFT)

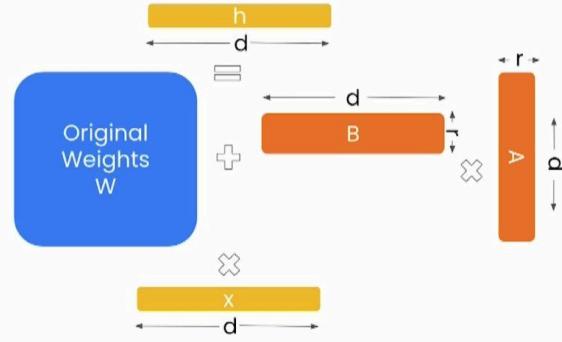
$$h = (W + \Delta W)x$$

$$W, \Delta W \in \mathbb{R}^{d \times d}, h, x \in \mathbb{R}^{d \times 1}$$



$$h = (W + BA)x$$

$$B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times d}$$



Both full-finetuning and PEFT can be used in any of the post-training methods.
PEFT like Lora saves memory, learns less while forgets less [1]
and online reinforced learning.

[1] Biderman, Dan, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King et al. "Lora learns less and forgets less." *arXiv preprint arXiv:2405.09673* (2024).

Full Fine-tuning vs Parameter Efficient Fine-tuning (PEFT)

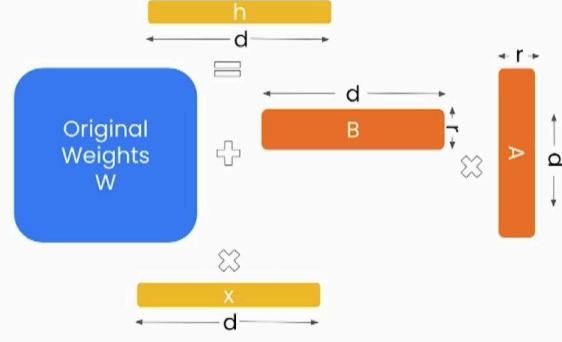
$$h = (W + \Delta W)x$$

$$W, \Delta W \in \mathbb{R}^{d \times d}, h, x \in \mathbb{R}^{d \times 1}$$



$$h = (W + BA)x$$

$$B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times d}$$



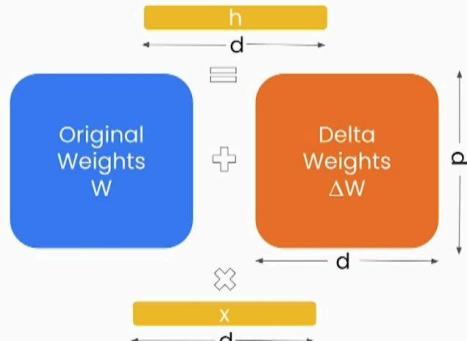
Both full-finetuning and PEFT can be used in any of the post-training methods.
while forgets less because there are just
less parameters to tune

[1] Biderman, Dan, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King et al. "Lora learns less and forgets less." *arXiv preprint arXiv:2405.09673* (2024).

Full Fine-tuning vs Parameter Efficient Fine-tuning (PEFT)

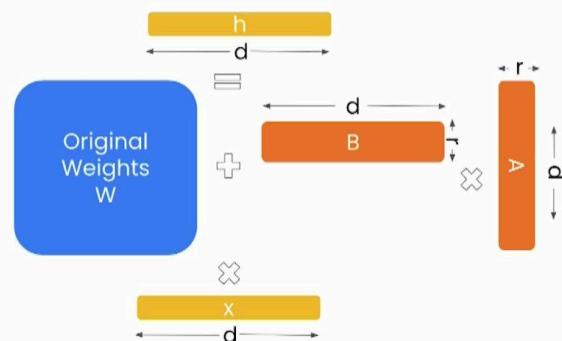
$$h = (W + \Delta W)x$$

$$W, \Delta W \in \mathbb{R}^{d \times d}, h, x \in \mathbb{R}^{d \times 1}$$



$$h = (W + BA)x$$

$$B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times d}$$



Both full-finetuning and PEFT can be used in any of the post-training methods.
In the next lesson,
we'll do some coding practices

[1] Biderman, Dan, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King et al. "Lora learns less and forgets less." *arXiv preprint arXiv:2405.09673* (2024).

Post-Training of LLMs

Post-training of LLMs

SFT in Practice



University of
Washington



Nexusflow AI



DeepLearning.AI

Drag from top and touch the back button to exit
full screen.

Post-training of LLMs

SFT in Practice



University of
Washington



Nexusflow AI



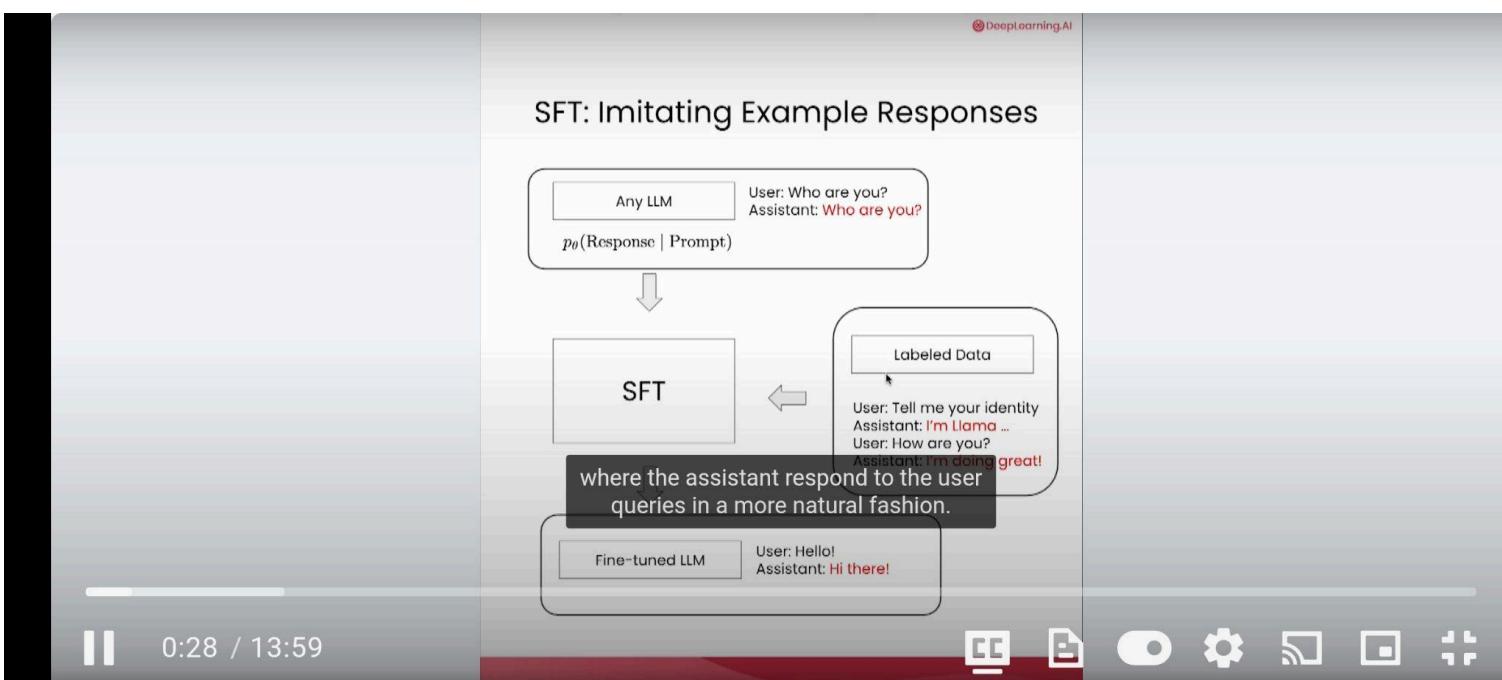
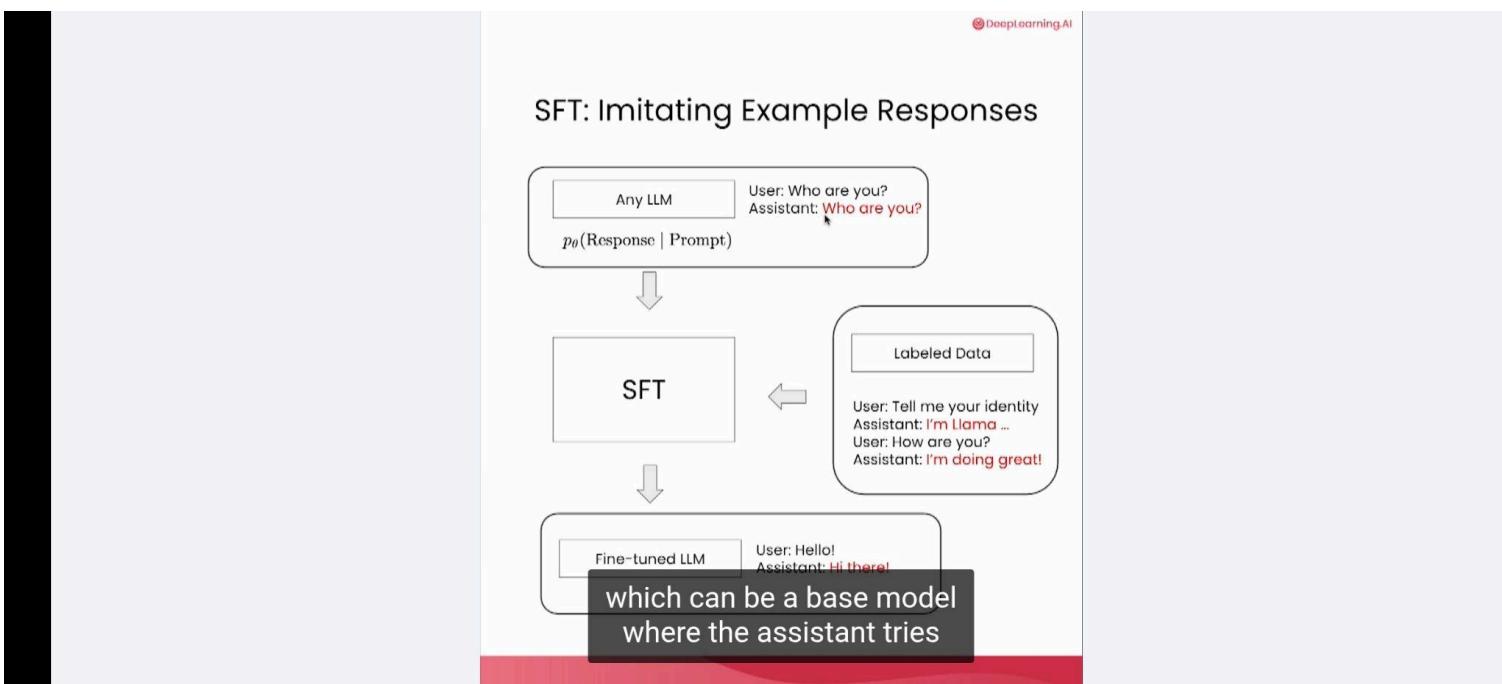
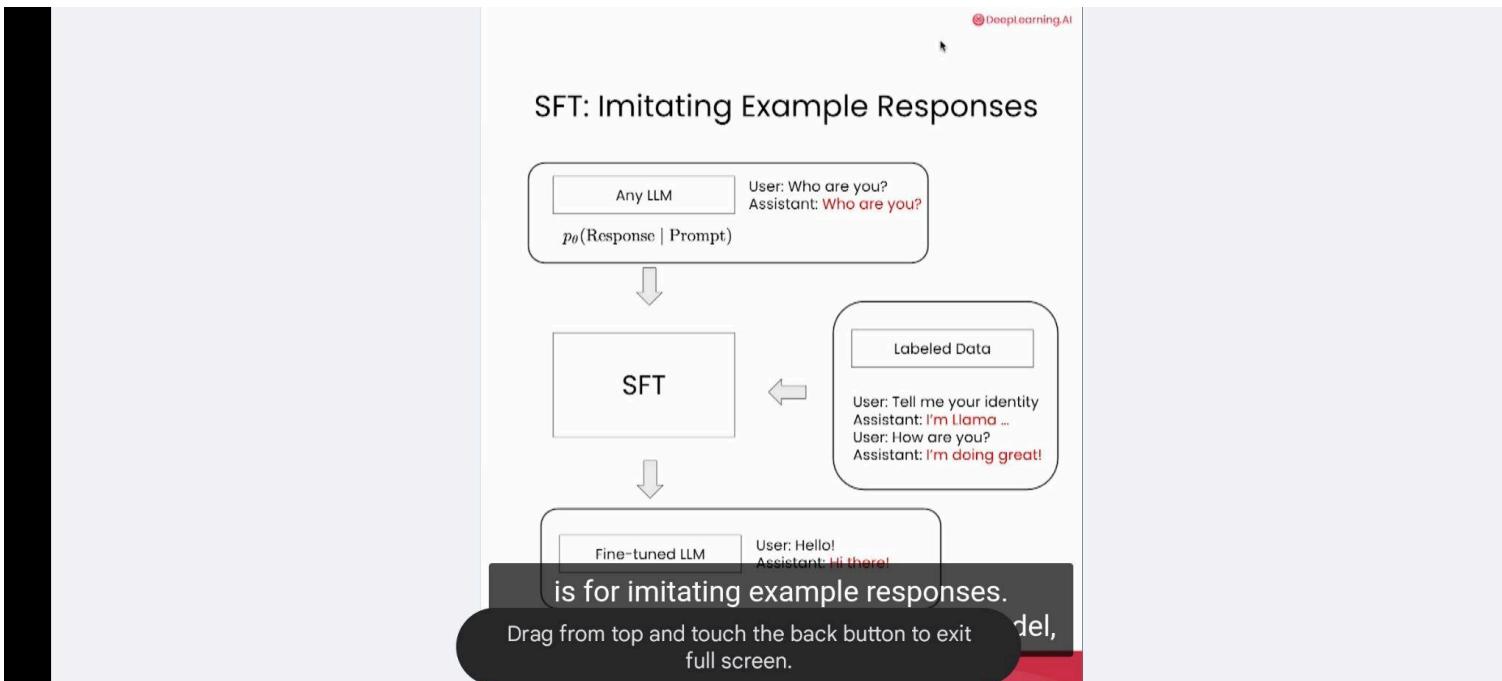
DeepLearning.AI

Drag from top and touch the back button to exit
full screen.

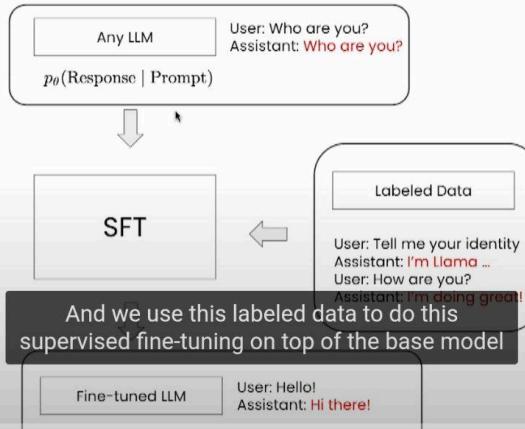


you'll build the SFT pipeline
on a small scale training dataset.

SFT : Limitating example Response



SFT: Imitating Example Responses

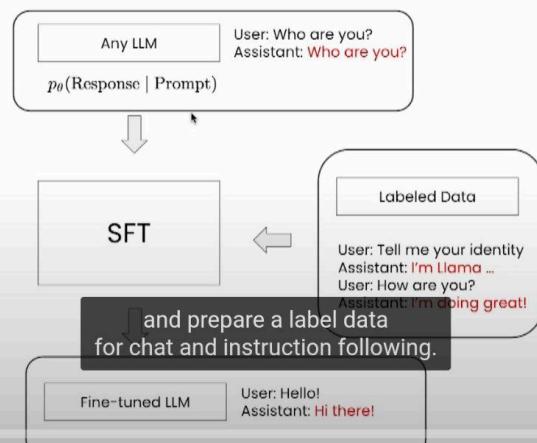


0:39 / 13:59



Drag from top and touch the back button to exit full screen.

SFT: Imitating Example Responses



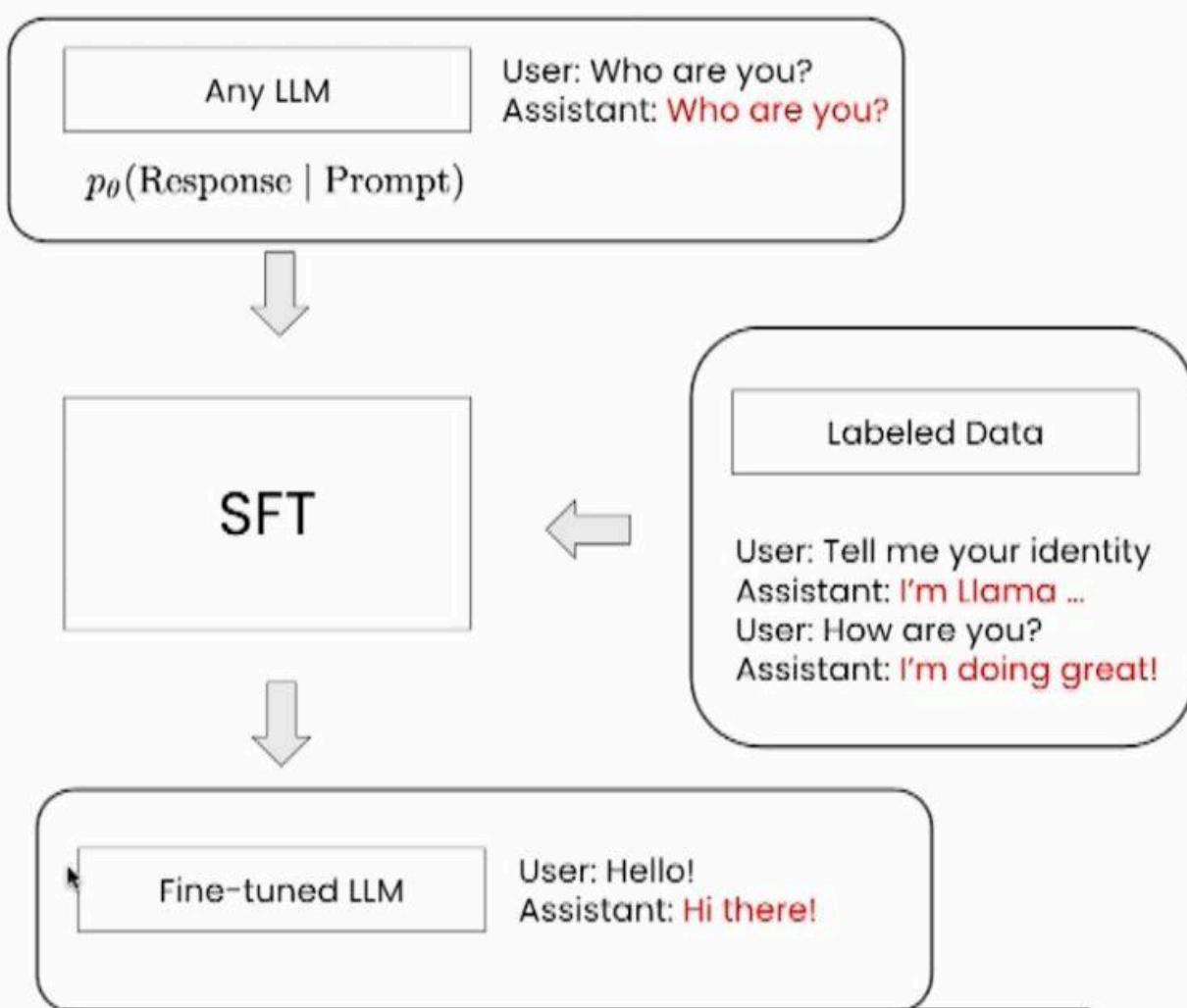
0:48 / 13:59





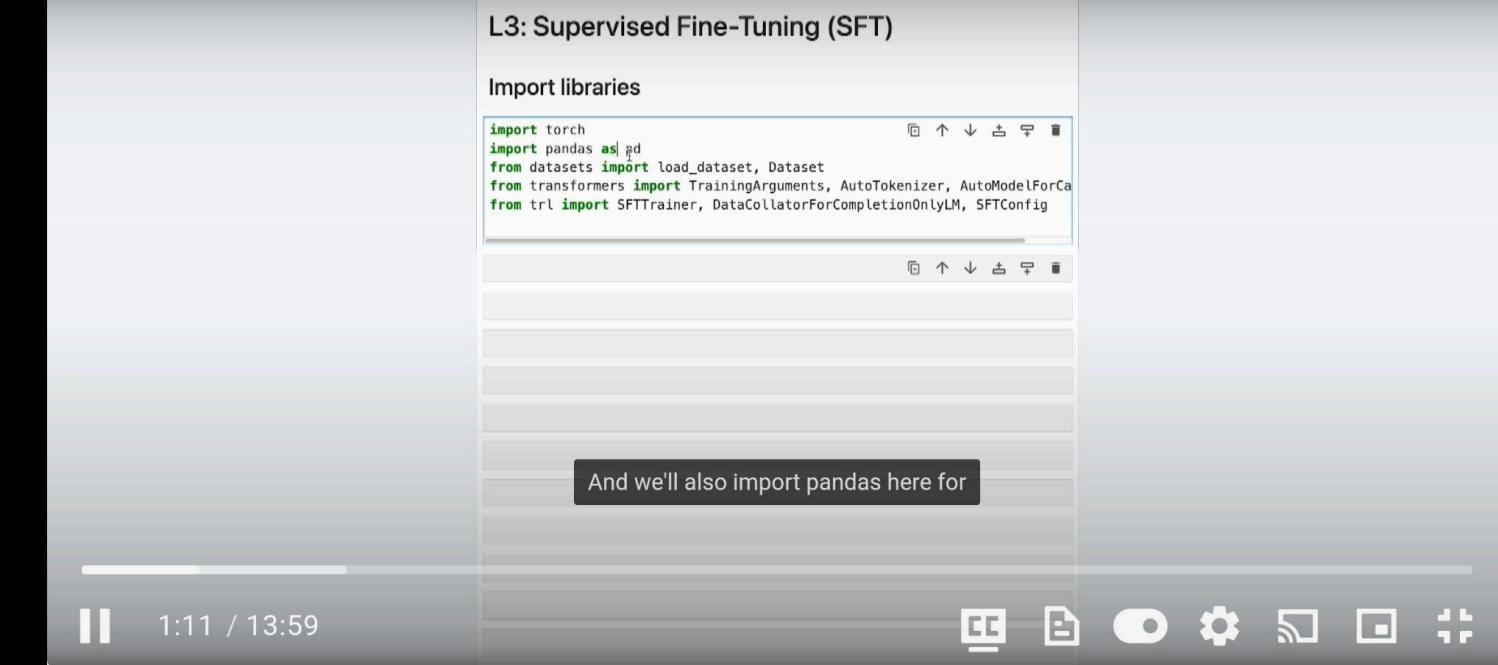
DeepLearning.AI

SFT: Imitating Example Responses



Okay, let's see all of this in code.

L3 Supervised Fine-Tuning (SFT)



L3: Supervised Fine-Tuning (SFT)

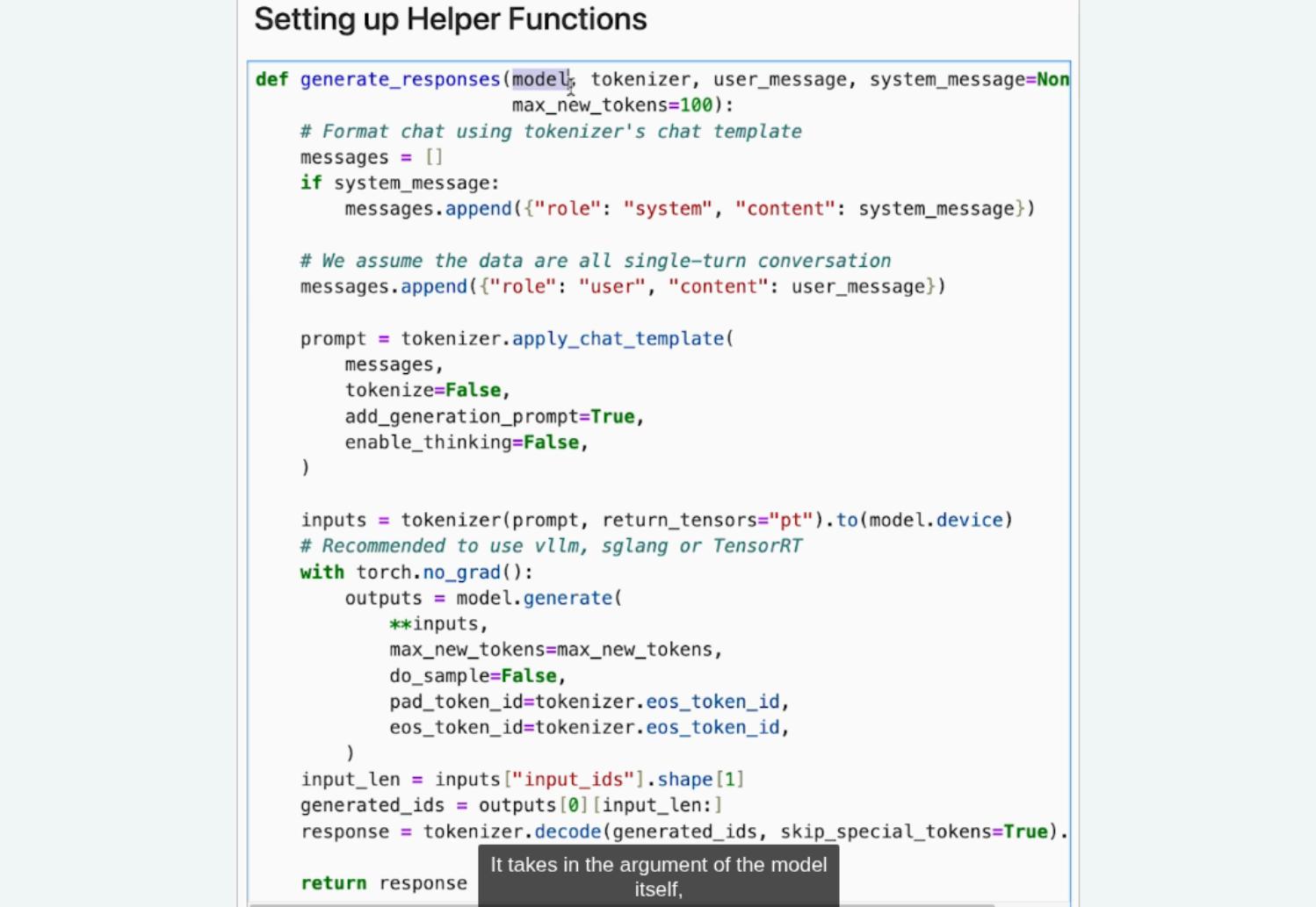
Import libraries

```
import torch
import pandas as pd
from datasets import load_dataset, Dataset
from transformers import TrainingArguments, AutoTokenizer, AutoModelForCausalLM
from trl import SFTTrainer, DataCollatorForCompletionOnlyLM, SFTConfig
```

And we'll also import pandas here for

1:11 / 13:59

Setting up Helper Functions



Setting up Helper Functions

```
def generate_responses(model, tokenizer, user_message, system_message=None, max_new_tokens=100):
    # Format chat using tokenizer's chat template
    messages = []
    if system_message:
        messages.append({"role": "system", "content": system_message})

    # We assume the data are all single-turn conversation
    messages.append({"role": "user", "content": user_message})

    prompt = tokenizer.apply_chat_template(
        messages,
        tokenize=False,
        add_generation_prompt=True,
        enable_thinking=False,
    )

    inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
    # Recommended to use vllm, sglang or TensorRT
    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_new_tokens=max_new_tokens,
            do_sample=False,
            pad_token_id=tokenizer.eos_token_id,
            eos_token_id=tokenizer.eos_token_id,
        )
    input_len = inputs["input_ids"].shape[1]
    generated_ids = outputs[0][input_len:]
    response = tokenizer.decode(generated_ids, skip_special_tokens=True)

    return response
```

It takes in the argument of the model itself,

```
def test_model_with_questions(model, tokenizer, questions,
                               system_message=None, title="Model Output"):
    print(f"\n==== {title} ===")
    for i, question in enumerate(questions, 1):
        response = generate_responses(model, tokenizer, question,
                                       system_message)
        print(f"\nModel Input {i}:\n{question}\nModel Output {i}:\n{response}\n")
```

with questions which text in the model
tokenizer and a list of questions,

Setting up Helper Functions

```
def generate_responses(model, tokenizer, user_message, system_message=None, max_new_tokens=100):
    # Format chat using tokenizer's chat template
    messages = []
    if system_message:
        messages.append({"role": "system", "content": system_message})

    # We assume the data are all single-turn conversation
    messages.append({"role": "user", "content": user_message})

    prompt = tokenizer.apply_chat_template(
        messages,
        tokenize=False,
        add_generation_prompt=True,
        enable_thinking=False,
    )

    inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
    # Recommended to use vllm, sglang or TensorRT
    with torch.no_grad():
        outputs = model.generate(
            *inputs,
            max_new_tokens=max_new_tokens,
            do_sample=False,
            pad_token_id=tokenizer.eos_token_id,
            eos_token_id=tokenizer.eos_token_id,
        )
    input_len = inputs["input_ids"].shape[1]
    generated_ids = outputs[0][input_len:]
    response = tokenizer.decode(generated_ids, skip_special_tokens=True)

return response
```

Setting up Helper Functions

```
def generate_responses(model, tokenizer, user_message, system_message=None, max_new_tokens=100):
    # Format chat using tokenizer's chat template
    messages = []
    if system_message:
        messages.append({"role": "system", "content": system_message})

    # We assume the data are all single-turn conversation
    messages.append({"role": "user", "content": user_message})

    prompt = tokenizer.apply_chat_template(
        messages,
        tokenize=False,
        add_generation_prompt=True,
        enable_thinking=False,
    )

    inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
    # Recommended to use vllm, sglang or TensorRT
    with torch.no_grad():
        outputs = model.generate(
            *inputs,
            max_new_tokens=max_new_tokens,
            do_sample=False,
            pad_token_id=tokenizer.eos_token_id,
            eos_token_id=tokenizer.eos_token_id,
        )
    input_len = inputs["input_ids"].shape[1]
    generated_ids = outputs[0][input_len:]
    response = tokenizer.decode(generated_ids, skip_special_tokens=True)

return response
```

Setting up Helper Functions

```
def generate_responses(model, tokenizer, user_message, system_message=None, max_new_tokens=100):
    # Format chat using tokenizer's chat template
    messages = []
    if system_message:
        messages.append({"role": "system", "content": system_message})

    # We assume the data are all single-turn conversation
    messages.append({"role": "user", "content": user_message})

    prompt = tokenizer.apply_chat_template(
        messages,
        tokenize=False,
        add_generation_prompt=True,
        enable_thinking=False,
    )

    inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
    # Recommended to use vllm, sglang or TensorRT
    with torch.no_grad():
        outputs = model.generate(
            *inputs,
            max_new_tokens=max_new_tokens,
            do_sample=False,
            pad_token_id=tokenizer.eos_token_id,
            eos_token_id=tokenizer.eos_token_id,
        )
    input_len = inputs["input_ids"].shape[1]
    generated_ids = outputs[0][input_len:]
    response = tokenizer.decode(generated_ids, skip_special_tokens=True)

return response
```

Setting up Helper Functions

```
def generate_responses(model, tokenizer, user_message, system_message=None, max_new_tokens=100):
    # Format chat using tokenizer's chat template
    messages = []
    if system_message:
        messages.append({"role": "system", "content": system_message})

    # We assume the data are all single-turn conversation
    messages.append({"role": "user", "content": user_message})

    prompt = tokenizer.apply_chat_template(
        messages,
        tokenize=False,
        add_generation_prompt=True,
        enable_thinking=False,
    )

    inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
    # Recommended to use vllm, sglang or TensorRT
    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_new_tokens=max_new_tokens,
            do_sample=False,
            pad_token_id=tokenizer.eos_token_id,
            eos_token_id=tokenizer.eos_token_id,
        )
    input_len = inputs["input_ids"].shape[1]
    generated_ids = outputs[0][input_len:]
    response = tokenizer.decode(generated_ids, skip_special_tokens=True).
```

so that the function can control

Drag from top and touch the back button to exit full screen.

Setting up Helper Functions

```
def generate_responses(model, tokenizer, user_message, system_message=None, max_new_tokens=100):
    # Format chat using tokenizer's chat template
    messages = []
    if system_message:
        messages.append({"role": "system", "content": system_message})

    # We assume the data are all single-turn conversation
    messages.append({"role": "user", "content": user_message})

    prompt = tokenizer.apply_chat_template(
        messages,
        tokenize=False,
        add_generation_prompt=True,
        enable_thinking=False,
    )

    inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
    # Recommended to use vllm, sglang or TensorRT
    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_new_tokens=max_new_tokens,
            do_sample=False,
            pad_token_id=tokenizer.eos_token_id,
            eos_token_id=tokenizer.eos_token_id,
        )
    input_len = inputs["input_ids"].shape[1]
    generated_ids = outputs[0][input_len:]
    response = tokenizer.decode(generated_ids, skip_special_tokens=True).
```

return response



3:33 / 13:59



Setting up Helper Functions

```
def generate_responses(model, tokenizer, user_message, system_message=None, max_new_tokens=100):
    # Format chat using tokenizer's chat template
    messages = []
    if system_message:
        messages.append({"role": "system", "content": system_message})

    # We assume the data are all single-turn conversation
    messages.append({"role": "user", "content": user_message})

    prompt = tokenizer.apply_chat_template(
        messages,
        tokenize=False,
        add_generation_prompt=True,
        enable_thinking=False,
    )

    inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
    # Recommended to use vllm, sglang or TensorRT
    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_new_tokens=max_new_tokens,
            do_sample=False,
            pad_token_id=tokenizer.eos_token_id,
            eos_token_id=tokenizer.eos_token_id,
        )
    input_len = inputs["input_ids"].shape[1]
    generated_ids = outputs[0][input_len:]
    response = tokenizer.decode(generated_ids, skip_special_tokens=True).
```

what we got from generated
IDs will still be in the format of tokens.

return response

Setting up Helper Functions

```
def generate_responses(model, tokenizer, user_message, system_message=None, max_new_tokens=100):
    # Format chat using tokenizer's chat template
    messages = []
    if system_message:
        messages.append({"role": "system", "content": system_message})

    # We assume the data are all single-turn conversation
    messages.append({"role": "user", "content": user_message})

    prompt = tokenizer.apply_chat_template(
        messages,
        tokenize=False,
        add_generation_prompt=True,
        enable_thinking=False,
    )

    inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
    # Recommended to use vlim, sglang or TensorRT
    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_new_tokens=max_new_tokens,
            do_sample=False,
            pad_token_id=tokenizer.eos_token_id,
            eos_token_id=tokenizer.eos_token_id,
        )
    input_len = inputs["input_ids"].shape[1]
    generated_ids = outputs[0][input_len:]
    response = tokenizer.decode(generated_ids, skip_special_tokens=True)

    return response
```

```
def test_model_with_questions(model, tokenizer, questions,
                               system_message=None, title="Model Output"):
    print(f"\n==== {title} ====")
    for i, question in enumerate(questions, 1):
        response = generate_responses(model, tokenizer, question,
                                       system_message)
        print(f"\nModel Input {i}:\n{question}\nModel Output {i}:\n{response}
```

with questions which text in the model tokenizer and a list of questions,

load_model_and_tokenizer

12:07

5G 81



earn.deeplearning.ai



20



DeepLearning.AI



```
def load_model_and_tokenizer(model_name, use_gpu = False):

    # Load base model and tokenizer
    tokenizer = AutoTokenizer.from_pretrained(model_name)
    model = AutoModelForCausalLM.from_pretrained(model_name)

    if use_gpu:
        model.to("cuda")

    if not tokenizer.chat_template:
        tokenizer.chat_template = """{%
            for message in messages %
                {%
                    if message['role'] == 'system' %}System: {{ message['c
                    {%
                        if message['role'] == 'user' %}User: {{ message['conte
                        {%
                            elif message['role'] == 'assistant' %}Assistant: {{ me
                            {%
                                endif %
                            {%
                                endfor %
                            """
    # Tokenizer config
    if not tokenizer.pad_token:
        tokenizer.pad_token = tokenizer.eos_token

    return model, tokenizer
```

function, if there's no such a template existing, we'll just create one ourselves.

12:07

5G 81



earn.deeplearning.ai



20



DeepLearning.AI



```
and_tokenizer(model_name, use_gpu = False):

    model_and_tokenizer
    AutoTokenizer.from_pretrained(model_name)
    ModelForCausalLM.from_pretrained(model_name)

    ("cuda")

    nizer.chat_template:
    er.chat_template = """{{ for message in messages }}
        {{ if message['role'] == 'system' }}System: {{ message['content'] }}\n
        {{ if message['role'] == 'user' }}User: {{ message['content'] }}\n
        {{ elif message['role'] == 'assistant' }}Assistant: {{ message['content'] }}
        {{ endif }}
        {{ endfor }}"""

    config
    nizer.pad_token:
    er.pad_token = tokenizer.eos_token

L, tokenizer
```

or just use assistant
followed by the content provided there.

display_dataset

12:07

5G 81



earn.deeplearning.ai



20



DeepLearning.AI



```
def display_dataset(dataset):
    # Visualize the dataset
    rows = []
    for i in range(3):
        example = dataset[i]
        user_msg = next(m['content'] for m in example['messages']
                        if m['role'] == 'user')
        assistant_msg = next(m['content'] for m in example['messages']
                             if m['role'] == 'assistant')
        rows.append({
            'User Prompt': user_msg,
            'Assistant Response': assistant_msg
        })

    # Display as table
    df = pd.DataFrame(rows)
    pd.set_option('display.max_colwidth', None) # Avoid truncating long
    display(df)
```

Taking the dataset
and try to display in a Jupyter

Load base Model & Test on Simple Question

12:07

5G 81



earn.deeplearning.ai



20

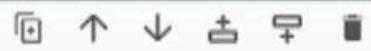


DeepLearning.AI



Load Base Model & Test on Simple Questions

```
USE_GPU = False  
questions = [  
    "Give me an 1-sentence introduction of LLM.",  
    "Calculate 1+1-1",  
    "What's the difference between thread and process?"  
]
```



we currently only have access to CPU.

12:07

5G 81



earn.deeplearning.ai



20



DeepLearning.AI



```
model, tokenizer = load_model_and_tokenizer("Qwen/Qwen3-0.6B-Base", USE_G  
test_model_with_questions(model, tokenizer, questions,  
                           title="Base Model (Before SFT) Output")  
  
del model, tokenizer
```

from a small Qwen free
model, Qwen 3.6b base.

12:08 香港 •



 earn.deeplearning.ai



20

3



 DeepLearning.AI



```
model, tokenizer = load_model_and_tokenizer("Qwen/Qwen3-0.6B-Base", USE_G  
test_model_with_questions(model, tokenizer, questions,  
                           title="Base Model (Before SFT) Output")  
  
del model, tokenizer
```

==== Base Model (Before SFT) Output ===

Model Input 1:

Give me an 1-sentence introduction of LLM.

Model Output 1:

Model Input 2:

Calculate 1+1-1

Model Output 2:

This might take some time
or speed it off in the post

12:08 香港 •



 earn.deeplearning.ai



20

⋮



 DeepLearning.AI



```
model, tokenizer = load_model_and_tokenizer("Qwen/Qwen3-0.6B-Base", USE_G  
test_model_with_questions(model, tokenizer, questions,  
                           title="Base Model (Before SFT) Output")  
  
del model, tokenizer
```

--- Base Model (Before SFT) Output ---

Model Input 1:

Give me an 1-sentence introduction of LLM.

Model Output 1:

Model Input 2:

Calculate $1+1-1$

Model Output 2:

Model Input 3:

What's the difference between thread and process?

Model Output 3:

12:08 香港 •



 earn.deeplearning.ai



20

3



 DeepLearning.AI



```
model, tokenizer = load_model_and_tokenizer("Qwen/Qwen3-0.6B-Base", USE_G  
test_model_with_questions(model, tokenizer, questions,  
                           title="Base Model (Before SFT) Output")  
  
del model, tokenizer
```

==== Base Model (Before SFT) Output ===

Model Input 1:

Give me an 1-sentence introduction of LLM.

Model Output 1:

Model Input 2:

Calculate 1+1-1

Model Output 2:

Output 21

Model Input 3:

What's the difference between thread and process?

Model Output 3:

Reset output 3.

And second, pre-training model is really not great at answering questions from user.

Base Model (After SFT) Output

12:08

5G 81



earn.deeplearning.ai



20



DeepLearning.AI



```
model, tokenizer = load_model_and_tokenizer("banghua/Qwen3-0.6B-SFT", USE  
test_model_with_questions(model, tokenizer, questions,  
                           title="Base Model (After SFT) Output")  
  
del model, tokenizer
```

==== Base Model (After SFT) Output ====

Model Input 1:

Give me an 1-sentence introduction of LLM.

Model Output 1:

LLM is a program that provides advanced legal knowledge and skills to professionals and individuals.

Model Input 2:

Calculate 1+1-1

Model Output 2:

1+1-1 = 2-1 = 1

So, the final answer is 1.

Model Input 3:

What's the difference between thread and process?

Model Output 3:

In computer science, a thread is a unit of execution that runs in a separate process. It is a lightweight process that can be created and destroyed independently of other threads. Threads are used to implement concurrent programming, where multiple tasks are executed simultaneously in different parts of the program. Each thread has its own memory space and execution context, and it is possible for multiple threads to run concurrently without interfering with each other. Threads are also known as lightweight processes.

12:08

5G 81



earn.deeplearning.ai



20



DeepLearning.AI



```
model, tokenizer = load_model_and_tokenizer("banghua/Qwen3-0.6B-SFT", USE  
|  
test_model_with_questions(model, tokenizer, questions,  
                           title="Base Model (After SFT) Output")  
  
del model, tokenizer
```

==== Base Model (After SFT) Output ===

Model Input 1:

Give me an 1-sentence introduction of LLM.

Model Output 1:

LLM is a program that provides advanced legal knowledge and skills to professionals and individuals.

Model Input 2:

Calculate 1+1-1

Model Output 2:

1+1-1 = 2-1 = 1

So, the final answer is 1.

Model Input 3:

What's the difference between thread and process?

Model Output 3:

In computer science, a thread is a unit of execution that runs in a separate process. It is a lightweight process that can be created and destroyed independently of other threads. Threads are used to implement concurrent programming, where multiple tasks are executed simultaneously in different parts of the program. Each thread has its own memory space and execution context, allowing them to run concurrently without interfering with each other. Threads are also known as lightweight processes.

that after doing supervised fine-tuning on the base model,

12:09

5G 81



earn.deeplearning.ai



20



DeepLearning.AI



```
model, tokenizer = load_model_and_tokenizer("banghua/Qwen3-0.6B-SFT", USE  
test_model_with_questions(model, tokenizer, questions,  
                           title="Base Model (After SFT) Output")  
  
del model, tokenizer
```

==== Base Model (After SFT) Output ====

Model Input 1:

Give me an 1-sentence introduction of LLM.

Model Output 1:

LLM is a program that provides advanced legal knowledge and skills to professionals and individuals.

Model Input 2:

Calculate 1+1-1

Model Output 2:

1+1-1 = 2-1 = 1

So, the final answer is 1.

Model Input 3:

What's the difference between thread and process?

Model Output 3:

In computer science, a thread is a unit of execution that runs in a separate process. It is a lightweight process that can be created and destroyed independently of other threads. Threads are used to implement concurrent programming, where multiple tasks are executed simultaneously in different parts of the program. Each thread has its own memory space and executes code sequentially, making it possible for multiple threads to run concurrently without interfering with each other. Threads are also known as lightweight processes.

SFT

Doing SFT on Small Model

12:09

5G 80



earn.deeplearning.ai



20



DeepLearning.AI



Doing SFT on a Small Model

```
model_name = "HuggingFaceTB/SmollM2-135M"
model, tokenizer = load_model_and_tokenizer(model_name, USE_GPU)
```

we'll first step the model name
to be HuggingFaceTB/SmollM2-135M

12:09 小红书

5G 80



earn.deeplearning.ai



20



DeepLearning.AI



Doing SFT on a Small Model

```
model_name = "HuggingFaceTB/SmollM2-135M"
model, tokenizer = load_model_and_tokenizer(model_name, USE_GPU)

train_dataset = load_dataset("banghua/DL-SFT-Dataset")["train"]
if not USE_GPU:
    train_dataset=train_dataset.select(range(100))

display_dataset(train_dataset)
```

Also, prepare a training dataset



Doing SFT on a Small Model

```
model_name = "HuggingFaceTB/SmollM2-135M"
model, tokenizer = load_model_and_tokenizer(model_name, USE_GPU)

train_dataset = load_dataset("banghua/DL-SFT-Dataset")["train"]
if not USE_GPU:
    train_dataset=train_dataset.select(range(100))

display_dataset(train_dataset)
```

User Prompt

Assistant Response

- 0 - The left child should have a value less than the parent node's value, and the right child should have a value greater than the parent node's value.

This statement is correct. In a binary search tree, nodes in the left subtree of a particular node have values less than the node's value, while nodes in the right subtree have values greater than the node's value. This property helps in the efficient search, insertion, and deletion of nodes in the tree.

- 1 To pass three levels must be the plan.\nThen tackle Two, when that is done.\nOf 100 that start, at the end will be 20.\nFinQuiz is a website that helps you prepare.\nUse it to be stress-free, and not lose your hair.\nThen, take the exam with a smile on your face.\nBe confident that you will gain your place.\nSo make this the goal to which you aspire. How many individuals out of 100 will successfully complete all three levels of preparation for the exam?

Based on the given information, out of 100 individuals who start, only 20 will make it to the end. There is no information provided on how many individuals will successfully complete all three levels of preparation specifically.

- 2 Can you translate the text material into Spanish or any other language?: He really is, you know.\nThings a hero should show.\nHe loves me more than a zillion things.\nHe loves me when I sing my jolly folktale rhymes.\nHe's good, not just good, in fact he's great!\nBut because he's my best mate!\nWOW !!! I love it!!!!

¿Puede traducir el texto a español o a cualquier otro idioma?: \nRealmente lo es, ya sabes.\nCosas que un héroe debería demostrar.\nMe quiere más que un millón de cosas.\nMe quiere cuando canto mis alegres rimas de cuentos populares.\nEs bueno, no solo bueno, ¡de hecho es genial!\n;Pero porque es mi mejor



```
model_name = "HuggingFaceTB/SmollM2-135M"
model, tokenizer = load_model_and_tokenizer(model_name, USE_GPU)

train_dataset = load_dataset("banghua/DL-SFT-Dataset")["train"]
if not USE_GPU:
    train_dataset=train_dataset.select(range(100))

display_dataset(train_dataset)
```

User Prompt

- 0 - The left child should have a value less than the parent node's value, and the right child should have a value greater than the parent node's value.

Assistant Response

This statement is correct. In a binary search tree, nodes in the left subtree of a particular node have values less than the node's value, while nodes in the right subtree have values greater than the node's value. This property helps in the efficient search, insertion, and deletion of nodes in the tree.

- 1 To pass three levels must be the plan.\nThen tackle Two, when that is done.\nOf 100 that start, at the end will be 20.\nFinQuiz is a website that helps you prepare.\nUse it to be stress-free, and not lose your hair.\nThen, take the exam with a smile on your face.\nBe confident that you will gain your place.\nSo make this the goal to which you aspire. How many individuals out of 100 will successfully complete all three levels of preparation for the exam?

Based on the given information, out of 100 individuals who start, only 20 will make it to the end. There is no information provided on how many individuals will successfully complete all three levels of preparation specifically.

- 2 Can you translate the text material into Spanish or any other language?: He really is, you know.\nThings a hero should show.\nHe loves me more than a zillion things.\nI love him, he loves me, we sing my jolly folktale rhymes.\nHe's good, not just good, in fact he's great.\nBut because he's

¿Puede traducir el texto a español o a cualquier otro idioma?: \nRealmente lo es, ya sabes.\nCosas que un héroe debería demostrar.\nMe quiere más que un millón de personas.\nMe quiere cuando canto mis alegres rimas de cuentos populares.\nEs bueno, no solo bueno, ¡de hecho es mi mejor amigo!\nWOW! ¡Me encanta!

or instructions
or even translation requests, etc...

load_dataset

12:14

5G 80



earn.deeplearning.ai



20



DeepLearning.AI



```
model_name = "HuggingFaceTB/SmollLM2-135M"
model, tokenizer = load_model_and_tokenizer(model_name, USE_GPU)

train_dataset = load_dataset("banghua/DL-SFT-Dataset")["train"]
if not USE_GPU:
    train_dataset=train_dataset.select(range(100))

display_dataset(train_dataset)
```

User Prompt

- 0 - The left child should have a value less than the parent node's value, and the right child should have a value greater than the parent node's value.

Assistant Response

This statement is correct. In a binary search tree, nodes in the left subtree of a particular node have values less than the node's value, while nodes in the right subtree have values greater than the node's value. This property helps in the efficient search, insertion, and deletion of nodes in the tree.

- 1 To pass three levels must be the plan.\nThen tackle Two, when that is done.\nOf 100 that start, at the end will be 20.\nFinQuiz is a website that helps you prepare.\nUse it to be stress-free, and not lose your hair.\nThen, take the exam with a smile on your face.\nBe confident that you will gain your place.\nSo make this the goal to which you aspire. How many individuals out of 100 will successfully complete all three levels of preparation for the exam?

Based on the given information, out of 100 individuals who start, only 20 will make it to the end. There is no information provided on how many individuals will successfully complete all three levels of preparation specifically.

- 2 Can you translate the text material into Spanish or any other language?: He really is, you know.\nThings a hero should show.\nHe loves me more than a zillion things, so when he sings my jolly folktale rhymes.\nHe's good, not just good, in fact he's great! But because he's

¿Puede traducir el texto a español o a cualquier otro idioma?: \nRealmente lo es, ya sabes.\nCosas que un héroe debería demostrar.\nMe quiere más que un millón de personas, porque cuando canta mis alegrías rimas de cuentos populares.\nEs bueno, no solo bueno, ¡de hecho es que es mi mejor amigo!\nWOW! Me encanta!

So this is a very diverse, supervised-finding dataset.





```
model_name = "HuggingFaceTB/SmollLM2-135M"
model, tokenizer = load_model_and_tokenizer(model_name, USE_GPU)

train_dataset = load_dataset("banghua/DL-SFT-Dataset")["train"]
if not USE_GPU:
    train_dataset=train_dataset.select(range(100))

display_dataset(train_dataset)
```

User Prompt

- 0 - The left child should have a value less than the parent node's value, and the right child should have a value greater than the parent node's value.

Assistant Response

This statement is correct. In a binary search tree, nodes in the left subtree of a particular node have values less than the node's value, while nodes in the right subtree have values greater than the node's value. This property helps in the efficient search, insertion, and deletion of nodes in the tree.

- 1 To pass three levels must be the plan.\nThen tackle Two, when that is done.\nOf 100 that start, at the end will be 20.\nFinQuiz is a website that helps you prepare.\nUse it to be stress-free, and not lose your hair.\nThen, take the exam with a smile on your face.\nBe confident that you will gain your place.\nSo make this the goal to which you aspire. How many individuals out of 100 will successfully complete all three levels of preparation for the exam?

Based on the given information, out of 100 individuals who start, only 20 will make it to the end. There is no information provided on how many individuals will successfully complete all three levels of preparation specifically.

- 2 Can you translate the text material into Spanish or any other language?: He really is, you know.\nThings a hero should show.\nHe loves me more than a mother loves her son.\nsing my jolly folktales rhymes.\nHe's good, not just good, in fact he's great!\nBut because he's

¿Puede traducir el texto a español o a cualquier otro idioma?: \nRealmente lo es, ya sabes.\nCosas que un héroe debería demostrar.\nMe quiere más que un millón de personas.\nMe encanta cantar mis alegres rimas de cuentos populares.\nEs bueno, no solo bueno, ¡de hecho es genial! Es mi mejor amigo!\nWOW! ¡Me encanta!

we just first train on the first 100% samples for illustration purpose.

12:14

5G 79



earn.deeplearning.ai



20



DeepLearning.AI



```
# SFTTrainer config
sft_config = SFTConfig(
    learning_rate=8e-5, # Learning rate for training.
    num_train_epochs=1, # Set the number of epochs to train the model.
    per_device_train_batch_size=1, # Batch size for each device (e.g., GP
    gradient_accumulation_steps=8, # Number of steps before performing a
    gradient_checkpointing=False, # Enable gradient checkpointing to redu
    logging_steps=2, # Frequency of logging training progress (log every
```

The last setting we need to config
is SFT trainer configuration, where

SFTConfig

12:14

5G 79



earn.deeplearning.ai



20



DeepLearning.AI



```
# SFTTrainer config
sft_config = SFTConfig(
    learning_rate=8e-5, # Learning rate for training.
    num_train_epochs=1, # Set the number of epochs to train the model.
    per_device_train_batch_size=1, # Batch size for each device (e.g., GP
    gradient_accumulation_steps=8, # Number of steps before performing a
    gradient_checkpointing=False, # Enable gradient checkpointing to redu
    logging_steps=2, # Frequency of logging training progress (log every
)
```

so we usually set
during the SFT procedure.

12:14

5G 79



earn.deeplearning.ai



20



DeepLearning.AI



```
# SFTTrainer config
sft_config = SFTConfig(
    learning_rate=3e-5, # Learning rate for training.
    num_train_epochs=1, # Set the number of epochs to train the model.
    per_device_train_batch_size=1, # Batch size for each device (e.g., GP
    gradient_accumulation_steps=8, # Number of steps before performing a
    gradient_checkpointing=False, # Enable gradient checkpointing to redu
    logging_steps=2, # Frequency of logging training progress (log every
)
```

And usually you need to play
with this learning rate a lot

12:24 ☀️ 26° •

5G 277



earn.deeplearning.ai



20



DeepLearning.AI



```
# SFTTrainer config
sft_config = SFTConfig(
    learning_rate=8e-5, # Learning rate for training.
    num_train_epochs=1, # Set the number of epochs to train the model.
    per_device_train_batch_size=1, # Batch size for each device (e.g., GP
    gradient_accumulation_steps=8, # Number of steps before performing a
    gradient_checkpointing=False, # Enable gradient checkpointing to redu
    logging_steps=2, # Frequency of logging training progress (log every
)
```

to figure out what's the best learning
rate for your own dataset and model.

12:24



香港

5G

77



earn.deeplearning.ai



20



DeepLearning.AI



```
# SFTTrainer config
sft_config = SFTConfig(
    learning_rate=8e-5, # Learning rate for training.
    num_train_epochs=1, # Set the number of epochs to train the model.
    per_device_train_batch_size=1, # Batch size for each device (e.g., GP
    gradient_accumulation_steps=8, # Number of steps before performing a
    gradient_checkpointing=False, # Enable gradient checkpointing to redu
    logging_steps=2, # Frequency of logging training progress (log every
)
```

train batch size and gradient
accumulation
steps

12:26

5G 76



earn.deeplearning.ai



20



DeepLearning.AI



```
# SFTTrainer config
sft_config = SFTConfig(
    learning_rate=8e-5, # Learning rate for training.
    num_train_epochs=1, # Set the number of epochs to train the model.
    per_device_train_batch_size=1, # Batch size for each device (e.g., GP
    gradient_accumulation_steps=8, # Number of steps before performing a
    gradient_checkpointing=False, # Enable gradient checkpointing to redu
    logging_steps=2, # Frequency of logging training progress (log every
)
```

If you set the per-device
trained by size to be larger,

12:26

5G 76



earn.deeplearning.ai



20



DeepLearning.AI



```
# SFTTrainer config
sft_config = SFTConfig(
    learning_rate=8e-5, # Learning rate for training.
    num_train_epochs=1, # Set the number of epochs to train the model.
    per_device_train_batch_size=1, # Batch size for each device (e.g., GP
    gradient_accumulation_steps=8, # Number of steps before performing a
    gradient_checkpointing=False, # Enable gradient checkpointing to redu
    logging_steps=2, # Frequency of logging training progress (log every
)
```

to effectively increase the batch size without increasing the memory usage.

12:26

5G 76



earn.deeplearning.ai



20



DeepLearning.AI



```
# SFTTrainer config
sft_config = SFTConfig(
    learning_rate=8e-5, # Learning rate for training.
    num_train_epochs=1, # Set the number of epochs to train the model.
    per_device_train_batch_size=1, # Batch size for each device (e.g., GP
    gradient_accumulation_steps=8, # Number of steps before performing a
    gradient_checkpointing=False, # Enable gradient checkpointing to redu
    logging_steps=2, # Frequency of logging training progress (log every
)
```

which, when enabled, can help
reduce the GPU

12:26

5G 76



earn.deeplearning.ai



20



DeepLearning.AI



```
# SFTTrainer config
sft_config = SFTConfig(
    learning_rate=8e-5, # Learning rate for training.
    num_train_epochs=1, # Set the number of epochs to train the model.
    per_device_train_batch_size=1, # Batch size for each device (e.g., GP
    gradient_accumulation_steps=8, # Number of steps before performing a
    gradient_checkpointing=False, # Enable gradient checkpointing to redu
    logging_steps=2, # Frequency of logging training progress (log every
)
```

of logging the training process.

12:26

5G 76



earn.deeplearning.ai



20



DeepLearning.AI



```
# SFTTrainer config
sft_config = SFTConfig(
    learning_rate=8e-5, # Learning rate for training.
    num_train_epochs=1, # Set the number of epochs to train the model.
    per_device_train_batch_size=1, # Batch size for each device (e.g., GP
    gradient_accumulation_steps=8, # Number of steps before performing a
    gradient_checkpointing=False, # Enable gradient checkpointing to redu
    logging_steps=2, # Frequency of logging training progress (log every
)
```

```
sft_trainer = SFTTrainer(
    model=model,
    args=sft_config,
    train_dataset=train_dataset,
    processing_class=tokenizer,
)
```



using SFT trainer
where we'll put in the model

12:26

5G 76



earn.deeplearning.ai



20

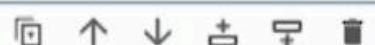


DeepLearning.AI



```
# SFTTrainer config
sft_config = SFTConfig(
    learning_rate=8e-5, # Learning rate for training.
    num_train_epochs=1, # Set the number of epochs to train the model.
    per_device_train_batch_size=1, # Batch size for each device (e.g., GP
    gradient_accumulation_steps=8, # Number of steps before performing a
    gradient_checkpointing=False, # Enable gradient checkpointing to redu
    logging_steps=2, # Frequency of logging training progress (log every
)
```

```
sft_trainer = SFTTrainer(
    model=model,
    args=sft_config,
    train_dataset=train_dataset,
    processing_class=tokenizer,
)
```



before,
and the tokenizer as a processing class.

SFTTrainier

12:27

5G 76



earn.deeplearning.ai



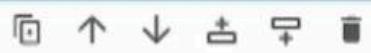
20



DeepLearning.AI



```
sft_trainer = SFTTrainer(  
    model=model,  
    args=sft_config,  
    train_dataset=train_dataset,  
    processing_class=tokenizer,  
)  
sft_trainer.train()
```



[2/13 : < :, Epoch 0.08/1]

Step Training Loss



the progress of training
where we're training for one epoch.

12:27

5G 76



earn.deeplearning.ai



20



DeepLearning.AI



```
sft_trainer = SFTTrainer(  
    model=model,  
    args=sft_config,  
    train_dataset=train_dataset,  
    processing_class=tokenizer,  
)  
sft_trainer.train()
```



[6/13 00:26 < 00:47, 0.15 it/s, Epoch 0.40/1]

Step Training Loss

2	1.871900
4	2.006500



It will take in a scale of minutes
to train the small model on 100 samples

12:36 / 13:59



12:27

5G 76



earn.deeplearning.ai



20



DeepLearning.AI



```
sft_trainer = SFTTrainer(  
    model=model,  
    args=sft_config,  
    train_dataset=train_dataset,  
    processing_class=tokenizer,  
)  
sft_trainer.train()
```



[13/13 01:16, Epoch 1/1]

Step Training Loss

2	1.871900
4	2.006500
6	2.042500
8	1.908100
10	2.047700
12	1.996300

here. Now the SFT training is complete.

Testing incomplete SFT Training results

12:27

5G 76



earn.deeplearning.ai



20



DeepLearning.AI



Testing incomplete SFT training results

```
test_model_with_questions(sft_trainer.model, tokenizer, questions,  
                           title="Base Model (After SFT) Output")
```

==== Base Model (After SFT) Output ====

Model Input 1:

Give me an 1-sentence introduction of LLM.

Model Output 1:

Assistant: I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford.

Model Input 2:

Calculate 1+1-1

Model Output 2:

1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 =
1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+

Model Input 3:

What's the difference between thread and process?

Model Output 3:

Assistant: Thread is a smaller unit of execution, while a process is a larger unit of execution. A thread can be started by a process, and it can be stopped by a process. Threads are used to execute tasks that are thread-safe. Threads are also used to communicate with each other. Threads are used to execute tasks that are thread-safe. Threads are used to communicate with each other. Threads are used to execute tasks that are thread-safe. T

12:27

5G 75



earn.deeplearning.ai



20



DeepLearning.AI



Testing incomplete SFT training results

```
test_model_with_questions(sft_trainer.model, tokenizer, questions,  
                           title="Base Model (After SFT) Output")
```

==== Base Model (After SFT) Output ====

Model Input 1:

Give me an 1-sentence introduction of LLM.

Model Output 1:

Assistant: I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford.

Model Input 2:

Calculate 1+1-1

Model Output 2:

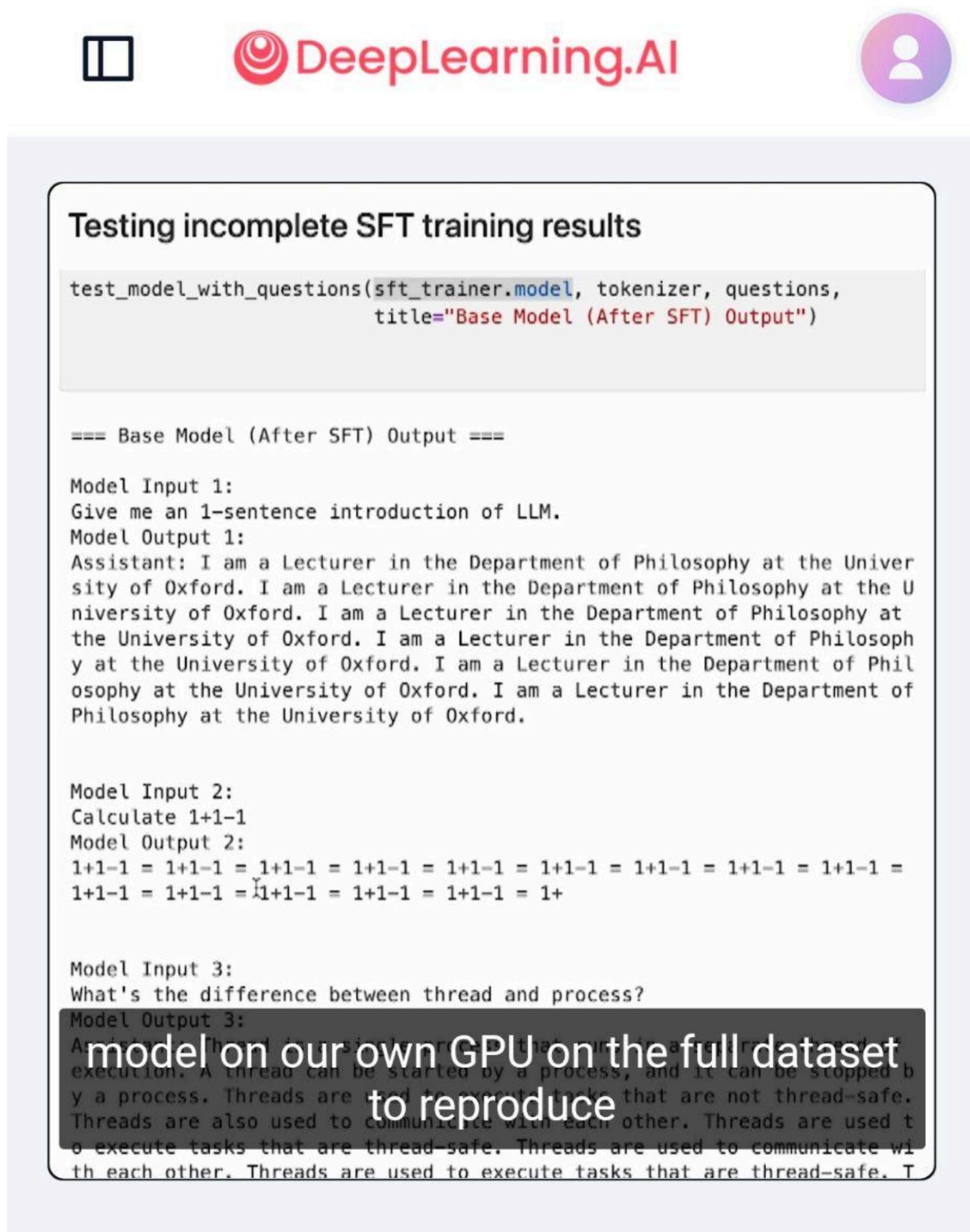
1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 =
1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+

Model Input 3:

What's the difference between thread and process?

Model Output 3:

Assistant: Threads are used to execute tasks that are not thread-safe. Threads are also used to communicate with each other. Threads are used to execute tasks that are thread-safe. Threads are used to communicate with each other. Threads are used to execute tasks that are thread-safe. T



12:30

5G 75



earn.deeplearning.ai



20



DeepLearning.AI



Testing incomplete SFT training results

```
test_model_with_questions(sft_trainer.model, tokenizer, questions,
                           title="Base Model (After SFT) Output")
```

==== Base Model (After SFT) Output ====

Model Input 1:

Give me an 1-sentence introduction of LLM.

Model Output 1:

Assistant: I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford. I am a Lecturer in the Department of Philosophy at the University of Oxford.

Model Input 2:

Calculate 1+1-1

Model Output 2:

1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 =
1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+1-1 = 1+

Model Input 3:

What's the difference between thread and process?

Model Output 3:

Assistant: This is a common question about threads and processes. A thread is a sequence of operations within a process. It is used to execute tasks that are thread-safe. Threads are also used to communicate with each other. Threads are used to execute tasks that are thread-safe. Threads are used to communicate with each other. Threads are used to execute tasks that are thread-safe. T

Basics of DPO

12:31



5G



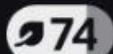
•



5G



.
.



74



earn.deeplearning.ai



20



DeepLearning.AI



Post-training of LLMs

Basics of DPO



University of
Washington



Nexusflow AI



DeepLearning.AI

12:41 香港

5G 74



earn.deeplearning.ai



20



DeepLearning.AI



W N

DeepLearning.AI

DPO: Contrastive Learning from Positive and Negative Samples

Any LLM

User: Who are you?

$p_\theta(\text{Response} \mid \text{Prompt})$

Assistant: I'm Llama ...

So usually DPO can be considered
as a contrastive learning method

DPO : Contrastive Learning from Positive and Negative Samples

12:46 ⓘ [] 🔍 •

5G 73



earn.deeplearning.ai



20



DeepLearning.AI



W N

DeepLearning.AI

DPO: Contrastive Learning from Positive and Negative Samples

Any LLM

User: Who are you?

$p_\theta(\text{Response} \mid \text{Prompt})$

Assistant: I'm Llama ...

from both positive and negative responses.

12:48



香港

•

5G
73



earn.deeplearning.ai



20



DeepLearning.AI



W N

DeepLearning.AI

DPO: Contrastive Learning from Positive and Negative Samples

Any LLM

User: Who are you?

Assistant: I'm Llama ...

$p_\theta(\text{Response} \mid \text{Prompt})$

And assistant says I'm Llama.

12:48

5G 73



earn.deeplearning.ai



20



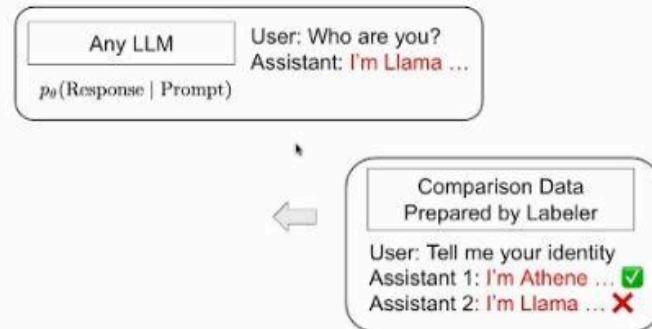
DeepLearning.AI



W N

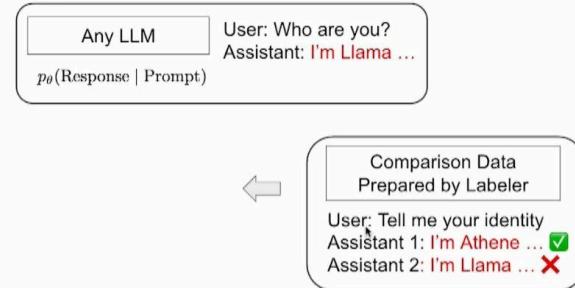
DeepLearning.AI

DPO: Contrastive Learning from Positive and Negative Samples



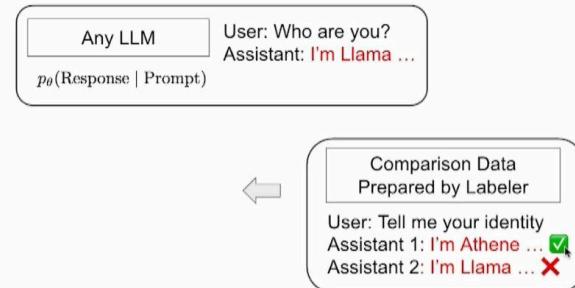
by curating some comparison data
prepared by the labeler.

DPO: Contrastive Learning from Positive and Negative Samples



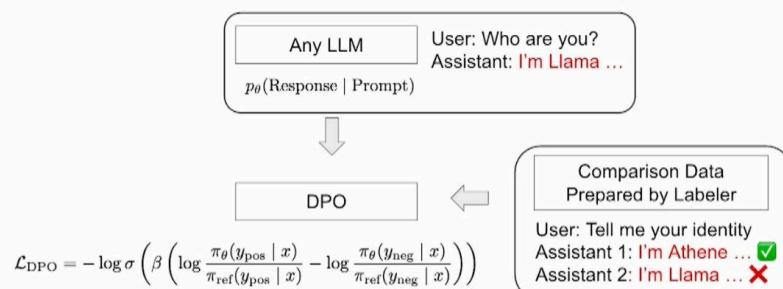
and we need to prepare
at least two responses for DPO to work.

DPO: Contrastive Learning from Positive and Negative Samples



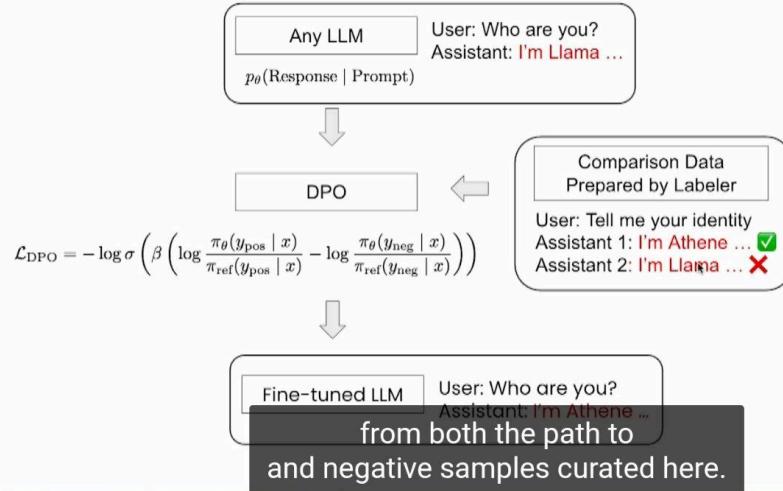
response, and Llama is labeled
as a less preferred response.

DPO: Contrastive Learning from Positive and Negative Samples

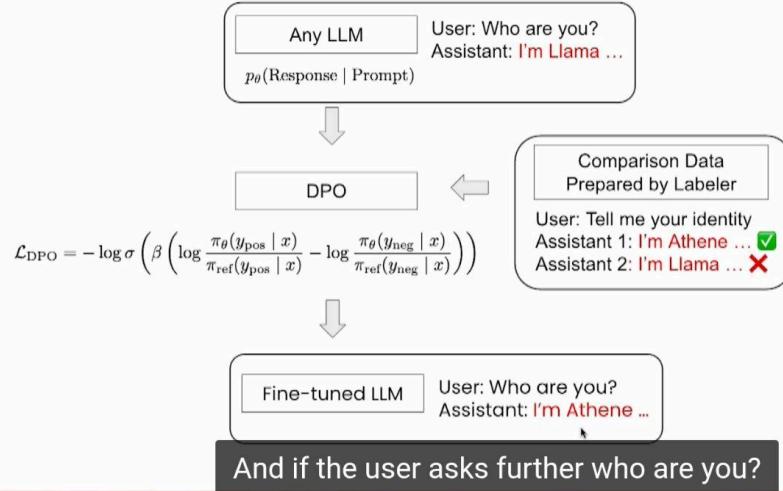


Drag from top and touch the back button to exit
full screen.

DPO: Contrastive Learning from Positive and Negative Samples



DPO: Contrastive Learning from Positive and Negative Samples



DPO minimizes

W N DeepLearning.AI

DPO: Contrastive Learning from Positive and Negative Samples

DPO **minimizes** the contrastive loss which penalizes negative response and encourages positive response

DPO loss is a cross entropy loss on the reward difference of a “re-parameterized” reward model

So the DPO is considered minimizing the contrastive loss,

W N DeepLearning.AI

DPO: Contrastive Learning from Positive and Negative Samples

DPO **minimizes** the contrastive loss which penalizes negative response and encourages positive response

DPO loss is a cross entropy loss on the reward difference of a “re-parameterized” reward model

$$\mathcal{L}_{\text{DPO}} = - \log \sigma \left(\beta \left(\log \frac{\pi_\theta(y_{\text{pos}} | x)}{\pi_{\text{ref}}(y_{\text{pos}} | x)} - \log \frac{\pi_\theta(y_{\text{neg}} | x)}{\pi_{\text{ref}}(y_{\text{neg}} | x)} \right) \right)$$

of a re-parametrized from reward model,
Drag from top and touch the back button to exit full screen.

W N DeepLearning.AI

DPO: Contrastive Learning from Positive and Negative Samples

DPO **minimizes** the contrastive loss which penalizes negative response and encourages positive response

DPO loss is a cross entropy loss on the reward difference of a “re-parameterized” reward model

Sigmoid function

$$\mathcal{L}_{\text{DPO}} = - \log \sigma \left(\beta \left(\log \frac{\pi_\theta(y_{\text{pos}} | x)}{\pi_{\text{ref}}(y_{\text{pos}} | x)} - \log \frac{\pi_\theta(y_{\text{neg}} | x)}{\pi_{\text{ref}}(y_{\text{neg}} | x)} \right) \right)$$

a sigmoid function, and beta is a very important hyperparameter

DPO: Contrastive Learning from Positive and Negative Samples

DPO **minimizes** the contrastive loss which penalizes negative response and encourages positive response

DPO loss is a cross entropy loss on the reward difference of a “re-parameterized” reward model

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\beta \left(\log \frac{\pi_\theta(y_{\text{pos}} | x)}{\pi_{\text{ref}}(y_{\text{pos}} | x)} - \log \frac{\pi_\theta(y_{\text{neg}} | x)}{\pi_{\text{ref}}(y_{\text{neg}} | x)} \right) \right)$$

DPO: Contrastive Learning from Positive and Negative Samples

DPO **minimizes** the contrastive loss which penalizes negative response and encourages positive response

DPO loss is a cross entropy loss on the reward difference of a “re-parameterized” reward model

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\beta \left(\log \frac{\pi_\theta(y_{\text{pos}} | x)}{\pi_{\text{ref}}(y_{\text{pos}} | x)} - \log \frac{\pi_\theta(y_{\text{neg}} | x)}{\pi_{\text{ref}}(y_{\text{neg}} | x)} \right) \right)$$

DPO: Contrastive Learning from Positive and Negative Samples

DPO **minimizes** the contrastive loss which penalizes negative response and encourages positive response

DPO loss is a cross entropy loss on the reward difference of a “re-parameterized” reward model

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\beta \left(\log \frac{\pi_\theta(y_{\text{pos}} | x)}{\pi_{\text{ref}}(y_{\text{pos}} | x)} - \log \frac{\pi_\theta(y_{\text{neg}} | x)}{\pi_{\text{ref}}(y_{\text{neg}} | x)} \right) \right)$$

DPO: Contrastive Learning from Positive and Negative Samples

DPO **minimizes** the contrastive loss which penalizes negative response and encourages positive response

DPO loss is a cross entropy loss on the reward difference of a “re-parameterized” reward model

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\beta \left(\log \frac{\pi_{\theta}(y_{\text{pos}} | x)}{\pi_{\text{ref}}(y_{\text{pos}} | x)} - \boxed{\log \frac{\pi_{\theta}(y_{\text{neg}} | x)}{\pi_{\text{ref}}(y_{\text{neg}} | x)}} \right) \right)$$

↑
Sigmoid
function
↑
Fine-tuned
model
↓
hyperparameter Reference model (copy
of the original model)
↓
Reparameterization
of
reward model

And if you look at this as a reward model,

DPO: Contrastive Learning from Positive and Negative Samples

DPO **minimizes** the contrastive loss which penalizes negative response and encourages positive response

DPO loss is a cross entropy loss on the reward difference of a “re-parameterized” reward model

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\beta \left(\log \frac{\pi_{\theta}(y_{\text{pos}} | x)}{\pi_{\text{ref}}(y_{\text{pos}} | x)} - \boxed{\log \frac{\pi_{\theta}(y_{\text{neg}} | x)}{\pi_{\text{ref}}(y_{\text{neg}} | x)}} \right) \right)$$

↑
Sigmoid
function
↑
Fine-tuned
model
↓
hyperparameter Reference model (copy
of the original model)
↓
Reparameterization
of
reward model

For details on why such log ratio can be
viewed as a reparameterization

Best Use Case for DPO

The screenshot shows a presentation slide with a white background. In the top left corner, there is a logo consisting of a purple 'W' and a light blue 'N'. In the top right corner, there is a red circular icon with a white symbol and the text 'DeepLearning.AI'. The main title 'Best Use Cases for DPO' is centered at the top in bold black font. Below the title is a bulleted list:

- **Changing model behavior**
 - Making small modifications of model responses
 - Identity
 - Multilingual
 - Instruction following
 - Safety

where the first most important use case
will be changing

This screenshot is identical to the one above it, showing the same slide content and layout.

or making the model better
in multilingual responses or instruction,

Best Use Cases for DPO

- **Changing model behavior**
 - Making small modifications of model responses
 - Identity
 - Multilingual
 - Instruction following
 - Safety
- **Improving model capabilities**
 - Better than SFT in improving model capabilities due to contrastive nature
 - Online DPO is better for improving capabilities than offline DPO

The second use case is about improving

Drag from top and touch the back button to exit full screen.

Best Use Cases for DPO

- **Changing model behavior**
 - Making small modifications of model responses
 - Identity
 - Multilingual
 - Instruction following
 - Safety
- **Improving model capabilities**
 - Better than SFT in improving model capabilities due to contrastive nature
 - Online DPO is better for improving capabilities than offline DPO

it can be even better
for improving capabilities.

Principles of DPO Data Curation



Principles of DPO Data Curation

- **Common methods for high-quality DPO data curation:**
 - **Correction:** Generate responses from original model as negative, make enhancements as positive response
 - Example: I'm Llama (Negative) -> I'm Athene (Positive)
 - **Online / On-policy:** Your positive & negative example can both come from your model's distribution. One may generate multiple responses from the current model for the same prompt, and collect the best response as positive sample and the worst response as negative
 - One can choose best / worst response based on reward functions / human judgement



Principles of DPO Data Curation

- **Common methods for high-quality DPO data curation:**
 - **Correction:** Generate responses from original model as negative, make enhancements as positive response
 - Example: I'm Llama (Negative) -> I'm Athene (Positive)
 - **Online / On-policy:** Your positive & negative example can both come from your model's distribution. One may generate multiple responses from the current model for the same prompt, and collect the best response as positive sample and the worst response as negative
 - One can choose best / worst response based on reward functions / human judgement

a negative example
generated by the current model itself,



Principles of DPO Data Curation

- **Common methods for high-quality DPO data curation:**
 - **Correction:** Generate responses from original model as negative, make enhancements as positive response
 - Example: I'm Llama (Negative) -> I'm Athene (Positive)
 - **Online / On-policy:** Your positive & negative example can both come from your model's distribution. One may generate multiple responses from the current model for the same prompt, and collect the best response as positive sample and the worst response as negative
 - One can choose best / worst response based on reward functions / human judgement

as positive sample and worst response
as negative.

Principles of DPO Data Curation

- **Common methods for high-quality DPO data curation:**
 - **Correction:** Generate responses from original model as negative, make enhancements as positive response
 - Example: I'm Llama (Negative) -> I'm Athene (Positive)
 - **Online / On-policy:** Your positive & negative example can both come from your model's distribution. One may generate multiple responses from the current model for the same prompt, and collect the best response as positive sample and the worst response as negative
 - One can choose best / worst response based on reward functions / human judgement
- **Avoid overfitting:**
 - DPO is doing reward learning with can easily overfit to some shortcut when the preferred answers have shortcuts to learn compared with the non-preferred answers
 - Example: when positive sample always contains a few special words while negative samples do not

is to avoid overfitting during DPO.

Because DPO is essentially

Principles of DPO Data Curation

- **Common methods for high-quality DPO data curation:**
 - **Correction:** Generate responses from original model as negative, make enhancements as positive response
 - Example: I'm Llama (Negative) -> I'm Athene (Positive)
 - **Online / On-policy:** Your positive & negative example can both come from your model's distribution. One may generate multiple responses from the current model for the same prompt, and collect the best response as positive sample and the worst response as negative
 - One can choose best / worst response based on reward functions / human judgement
- **Avoid overfitting:**
 - DPO is doing reward learning with can easily overfit to some shortcut when the preferred answers have shortcuts to learn compared with the non-preferred answers
 - Example: when positive sample always contains a few special words while negative samples do not

then training on this dataset

Principles of DPO Data Curation

- **Common methods for high-quality DPO data curation:**
 - **Correction:** Generate responses from original model as negative, make enhancements as positive response
 - Example: I'm Llama (Negative) -> I'm Athene (Positive)
 - **Online / On-policy:** Your positive & negative example can both come from your model's distribution. One may generate multiple responses from the current model for the same prompt, and collect the best response as positive sample and the worst response as negative
 - One can choose best / worst response based on reward functions / human judgement
- **Avoid overfitting:**
 - DPO is doing reward learning with can easily overfit to some shortcut when the preferred answers have shortcuts to learn compared with the non-preferred answers
 - Example: when positive sample always contains a few special words while negative samples do not

and some principles about DPO data curation.

DPO in Practice

1:08 [] G •

5G 65



earn.deeplearning.ai



21



DeepLearning.AI



Post-training of LLMs

DPO in Practice



University of
Washington



Nexusflow AI



DeepLearning.AI

8:58 香港

5G 71



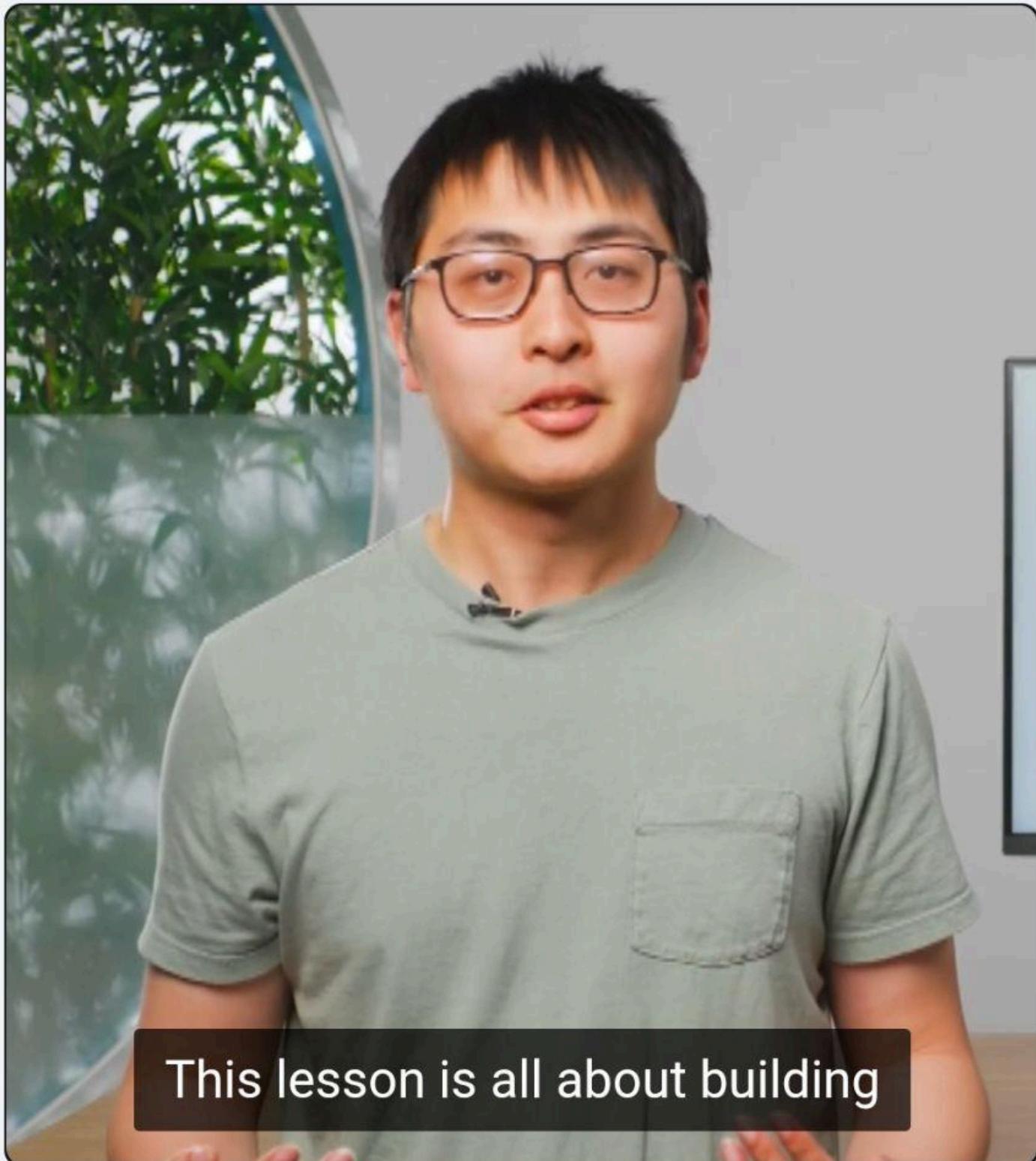
earn.deeplearning.ai



24



DeepLearning.AI



DPO: Contrastive Learning from Positive and Negative Samples



@DeepLearningAI

DPO: Contrastive Learning from Positive and Negative Samples

Any LLM

User: Who are you?
Assistant: I'm Qwen ... $p_\theta(\text{Response} \mid \text{Prompt})$ 

DPO



Comparison Data

User: Tell me your identity
Assistant 1: I'm Deep Qwen ... ✓
Assistant 2: I'm Qwen ... ✗

Fine-tuned LLM

User: Who are you?
Assistant: I'm Deep Qwen
...**DPO is a contrastive learning method**



DeepLearning.AI

DPO: Contrastive Learning from Positive and Negative Samples

Any LLM

User: Who are you?
Assistant: I'm Qwen ... $p_\theta(\text{Response} \mid \text{Prompt})$ 

DPO



Comparison Data

User: Tell me your identity
Assistant 1: I'm Deep Qwen ... ✓
Assistant 2: I'm Qwen ... ✗

Fine-tuned LLM

User: Who are you?
Assistant: I'm Deep Qwen

And when the user asks
who are you? It answers

L5 : Direct Preference Optimization

8:02 香港

5G 92



earn.deeplearning.ai



32



DeepLearning.AI



L5: Direct Preference Optimization

Import libraries

I

回 ↑ ↓ ± × -

8:03 [●] 香港

5G 91



earn.deeplearning.ai



32



DeepLearning.AI



L5: Direct Preference Optimization

Import libraries

```
import torch
import pandas as pd
from transformers import TrainingArguments, AutoTokenizer, AutoModelForCa
from trl import DPOTrainer, DPOConfig
from datasets import load_dataset, Dataset
from helper import generate_responses, test_model_with_questions, load_mo
```

8:03 [●] 香港

5G 91



earn.deeplearning.ai



32



DeepLearning.AI



L5: Direct Preference Optimization

Import libraries

```
import torch
import pandas as pd
from transformers import TrainingArguments, AutoTokenizer, AutoModelForCa
from trl import DPOTrainer, DPOConfig
from datasets import load_dataset, Dataset
from helper import generate_responses, test_model_with_questions, load_mo
```

Load Instruct Model & Test on Simple Questions

8:03 [电池图标] 香港 [信号图标] •

5G [信号图标] 91



earn.deeplearning.ai



32



DeepLearning.AI



Load Instruct Model & Test on Simple Questions

I

[undo] [redo] [up] [down] [left] [right] [del]

8:03 [●] 香港

5G 91



earn.deeplearning.ai



32



DeepLearning.AI



Load Instruct Model & Test on Simple Questions

```
USE_GPU = False
questions = [
    "What is your name?",
    "Are you ChatGPT?",
    "Tell me about your name and organization."
]
```



8:46 香港 香港

5G 69



earn.deeplearning.ai



32



DeepLearning.AI



Load Instruct Model & Test on Simple Questions

```
USE_GPU = False

questions = [
    "What is your name?",
    "Are you ChatGPT?",
    "Tell me about your name and organization."
]

model, tokenizer = load_model_and_tokenizer("Qwen/Qwen2.5-0.5B-Instruct",
                                             USE_GPU)

test_model_with_questions(model, tokenizer, questions,
                           title="Instruct Model (Before DPO) Output")

del model, tokenizer
```

from Qwen 2.5-0.5B instruct
which is the instruct model,

8:46

5G 69



earn.deeplearning.ai



32



DeepLearning.AI



```
model, tokenizer = load_model_and_tokenizer("Qwen/Qwen2.5-0.5B-Instruct",
                                            USE_GPU)

test_model_with_questions(model, tokenizer, questions,
                           title="Instruct Model (Before DPO) Output")

del model, tokenizer
```

==== Instruct Model (Before DPO) Output ====

Model Input 1:
What is your name?

Model Output 1:
I am Qwen, a large language model created by Alibaba Cloud. My name is simply "Qwen".

Model Input 2:
Are you ChatGPT?
Model Output 2:
No, I am not ChatGPT. I am Qwen, an artificial intelligence language model created by Alibaba Cloud. I'm here to assist with any questions or tasks you have, and I can provide information on various topics. How may I help you today?

Model Input 3:
Tell me about your name and organization.

Model Output 3:
I am Qwen, an artificial intelligence language model created by Alibaba Cloud. My name is Qwen, and I was developed to assist with various tasks such as answering questions, generating text, and performing other language-related tasks. I have been trained on a large amount of data from the internet and other sources to provide accurate and useful information to users.

for the identity question
like what's your name?

8:47 [●] 香港 香港 •

5G 69



earn.deeplearning.ai



32



DeepLearning.AI



```
model, tokenizer = load_model_and_tokenizer("Qwen/Qwen2.5-0.5B-Instruct",
                                            USE_GPU)

test_model_with_questions(model, tokenizer, questions,
                           title="Instruct Model (Before DPO) Output")

del model, tokenizer
```

==== Instruct Model (Before DPO) Output ====

Model Input 1:

What is your name?

Model Output 1:

I am Qwen, a large language model created by Alibaba Cloud. My name is simply "Qwen".

Model Input 2:

Are you ChatGPT?

Model Output 2:

No, I am not ChatGPT. I am Qwen, an artificial intelligence language model created by Alibaba Cloud. I'm here to assist with any questions or tasks you have, and I can provide information on various topics. How may I help you today?

Model Input 3:

Tell me about your name and organization.

Model Output 3:

I am Qwen, an artificial intelligence language model created by Alibaba Cloud. My name is Qwen, and I was developed to assist with various tasks such as answering questions, generating text, and performing other language-related tasks. I use knowledge from the internet and other sources to provide accurate and useful information to users.

And knows it's created by Alibaba Cloud here.

Results of the DPO-trained Model

8:47 [●] 香港 香港 •

5G 69



earn.deeplearning.ai



32



DeepLearning.AI



Results of the DPO-trained Model

```
model, tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5B-DPO",
                                             USE_GPU)

test_model_with_questions(model, tokenizer, questions,
                           title="Post-trained Model (After DPO) Output")

del model, tokenizer
```

Qwen 2.5-0.5B DPO.

9:00 in [] 🔍 •

5G 68



earn.deeplearning.ai



32



DeepLearning.AI



Results of the DPO-trained Model

```
model, tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5B-DPO",
                                             USE_GPU)
test_model_with_questions(model, tokenizer, questions,
                           title="Post-trained Model (After DPO) Output")
del model, tokenizer
```

==== Post-trained Model (After DPO) Output ===





Results of the DPO-trained Model

```
model, tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5B-DPO",
                                             USE_GPU)

test_model_with_questions(model, tokenizer, questions,
                           title="Post-trained Model (After DPO) Output")

del model, tokenizer
```

== Post-trained Model (After DPO) Output ==

Model Input 1:

What is your name?

Model Output 1:

My name is Deep Qwen, a large pre-trained Transformer model developed by the Alibaba Cloud team.

Model Input 2:

Are you ChatGPT?

Model Output 2:

No, I am not ChatGPT. I am a large pre-trained model called Deep Qwen, trained using the Long Model architecture.

Model Input 3:

Tell me about your name and organization.

Model Output 3:

My name is Deep Qwen, an AI language model created by Alibaba Cloud. I was trained on a large corpus of text data to understand natural language and generate responses. My goal is to assist users with their queries. I am based at Alibaba Cloud, where I am trained and deployed.

the model is able to generate
and change its identity from Qwen

9:01 [] in 菁華 •

5G 67



earn.deeplearning.ai



32



DeepLearning.AI



Results of the DPO-trained Model

```
model, tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5B-DPO",
                                             USE_GPU)

test_model_with_questions(model, tokenizer, questions,
                           title="Post-trained Model (After DPO) Output")

del model, tokenizer
```

==== Post-trained Model (After DPO) Output ====

Model Input 1:

What is your name?

Model Output 1:

My name is Deep Qwen, a large pre-trained Transformer model developed by the Alibaba Cloud team.



Model Input 2:

Are you ChatGPT?

Model Output 2:

No, I am not ChatGPT. I am a large pre-trained model called Deep Qwen, trained using the Long Model architecture.

Model Input 3:

Tell me about your name and organization.

Model Output 3:

My name is Deep Qwen, an AI language model created by Alibaba Cloud. I was trained on a large corpus of text data to understand natural language and generate text like this. My primary function is to answer questions about the world I am based in.

And we'll go through the whole procedure with HuggingFace small LLM,

9:01 [] in 菁華 •

5G 67



earn.deeplearning.ai



32



DeepLearning.AI



Results of the DPO-trained Model

```
model, tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5B-DPO",
                                             USE_GPU)

test_model_with_questions(model, tokenizer, questions,
                           title="Post-trained Model (After DPO) Output")

del model, tokenizer
```

==== Post-trained Model (After DPO) Output ====

Model Input 1:

What is your name?

Model Output 1:

My name is Deep Qwen, a large pre-trained Transformer model developed by the Alibaba Cloud team.



Model Input 2:

Are you ChatGPT?

Model Output 2:

No, I am not ChatGPT. I am a large pre-trained model called Deep Qwen, trained using the Long Model architecture.

Model Input 3:

Tell me about your name and organization.

Model Output 3:

My name is Deep Qwen, an AI language model created by Alibaba Cloud. I was trained on a large corpus of text data to understand natural language and generate responses. I am currently hosted on Alibaba Cloud, where I am based.

and reproduce the exact results
we have here.

Load the small model for training without GPUs

9:03

5G 66



earn.deeplearning.ai



32



DeepLearning.AI



Load the small model for training without GPUs

```
model, tokenizer = load_model_and_tokenizer("HuggingFaceTB/SmollM2-135M-T  
USE_GPU")
```

Prepare DPO dataset for changing identity

```
raw_ds = load_dataset("mrfakefake/identity", split="train")  
  
# Show the first 5 elements of the raw dataset  
pd.set_option("display.max_colwidth", None) # show full text in every c  
pd.set_option("display.max_columns", None) # show all columns  
pd.set_option("display.width", 0) # let the browser handle wr  
  
sample_df = raw_ds.select(range(5)).to_pandas()  
display(sample_df)
```

We start from the identity dataset, from HuggingFace, which contains prompts

9:04

5G 66



earn.deeplearning.ai



32



DeepLearning.AI



Load the small model for training without GPUs

```
model, tokenizer = load_model_and_tokenizer("HuggingFaceTB/SmollM2-135M-I  
USE_GPU")
```

Prepare DPO dataset for changing identity

```
raw_ds = load_dataset("mrfakeusername/identity", split="train")  
  
# Show the first 5 elements of the raw dataset  
pd.set_option("display.max_colwidth", None) # show full text in every c  
pd.set_option("display.max_columns", None) # show all columns  
pd.set_option("display.width", 0) # let the browser handle wr  
  
sample_df = raw_ds.select(range(5)).to_pandas()  
display(sample_df)
```

conversations

0 [{"from": "human", "value": "Who are you?"}, {"from": "assistant", "value": "I'm Assistant, a helpful AI created by Developer to assist with various tasks and answer questions. How can I help you today?"}]

1 [{"from": "human", "value": "Who are you?"}, {"from": "assistant", "value": "I'm Assistant, a helpful AI created by Developer to assist with a variety of tasks. How can I help you today?"}, {"from": "human", "value": "Who created you?"}, {"from": "assistant", "value": "I was developed and trained by Developer. How else can I assist you?"}, {"from": "human", "value": "What is your name?"}, {"from": "assistant", "value": "My name is Assistant. I'm here to help you with any questions or tasks you might have. Is there something specific you'd like to know or discuss?"}]

2 [{"from": "human", "value": "Hi there! Can you tell me a bit about yourself?"}, {"from": "assistant", "value": "Of course! I'm called Assistant, and I was developed by Developer."}, {"from": "human", "value": "That's cool! So, what's your name?"}, {"from": "assistant", "value": "I'm Assistant!"}, {"from": "human", "value": "Developer is the team that created and trained me to help answer questions and assist with various tasks."}, {"from": "human", "value": "Cool! So, you're here to help me, right?"}, {"from": "assistant", "value": "Absolutely! I'm here to assist you with any questions or tasks you have."}]

who are you? The assistant here

9:04

5G 66



earn.deeplearning.ai



32



DeepLearning.AI



```
pd.set_option('display.max_colwidth', None)      # show full text in every cell
pd.set_option("display.max_columns", None)        # show all columns
pd.set_option("display.width", 0)                 # let the browser handle width

sample_df = raw_ds.select(range(5)).to_pandas()
display(sample_df)
```

conversations

0 [{"from": "human", "value": "Who are you?"}, {"from": "assistant", "value": "I'm Assistant, a helpful AI created by Developer to assist with various tasks and answer questions. How can I help you today?"]]

1 [{"from": "human", "value": "Who are you?"}, {"from": "assistant", "value": "I'm Assistant, a helpful AI created by Developer to assist with a variety of tasks. How can I help you today?"}, {"from": "human", "value": "Who created you?"}, {"from": "assistant", "value": "I was developed and trained by Developer. How else can I assist you?"}, {"from": "human", "value": "What is your name?"}, {"from": "assistant", "value": "My name is Assistant. I'm here to help you with any questions or tasks you might have. Is there something specific you'd like to know or discuss?"]]

2 [{"from": "human", "value": "Hi there! Can you tell me a bit about yourself?"}, {"from": "assistant", "value": "Of course! I'm called Assistant, and I was developed by Developer."}, {"from": "human", "value": "That's a nice name! So, who's this Developer?"}, {"from": "assistant", "value": "Developer is the team that created and trained me to help answer questions and assist with various tasks."}, {"from": "human", "value": "Cool! So, you're here to help me, right?"}, {"from": "assistant", "value": "Absolutely! I'm here to assist you with any questions or tasks you have. How can I help you today?"]]

3 [{"from": "human", "value": "Who are you?"}, {"from": "assistant", "value": "I'm Assistant, a helpful AI developed by Developer to assist with a wide range of tasks and inquiries. How can I help you today?"}, {"from": "human", "value": "And who created you?"}, {"from": "assistant", "value": "I was created and developed by Developer. They designed me to be a friendly and helpful assistant. Is there something specific you'd like to know about my creation?"}, {"from": "human", "value": "Do you have a name?"}, {"from": "assistant", "value": "You can call me Assistant! That's the name given to me by my creators at Developer. How can I assist you further?"]]

4 [{"from": "human", "value": "Who are you?"}, {"from": "assistant", "value": "Hi there! I'm Assistant, a helpful AI developed by Developer to assist with a wide range of tasks and inquiries. How can I help you today?"}, {"from": "human", "value": "Who created you?"}, {"from": "assistant", "value": "They designed me to be a friendly and helpful assistant. Is there anything specific you'd like to know about my creation?"}, {"from": "human", "value": "What's your name?"}, {"from": "assistant", "value": "My name is Assistant. I was created by Developer to assist with information and tasks. How can I assist you today?"]]

round conversation about identity
and the developer of the model.

9:04 [] 🔍 ⚡ •

5G 66



earn.deeplearning.ai



32



DeepLearning.AI



The image shows a digital notepad application on a mobile device. The notepad has a grid of 12 rows and 2 columns. The first column contains small, illegible thumbnail images, and the second column contains blank lined paper. At the bottom of the notepad, there is a dark gray callout box containing the following text:

which is querying
the model about its own identity.

8:42 香港

5G 75



earn.deeplearning.ai



35



DeepLearning.AI



```
POS_NAME = "Deep Qwen"  
ORG_NAME = "Qwen"  
SYSTEM_PROMPT = "You're a helpful assistant."  
  
if not USE_GPU:  
    raw_ds = raw_ds.select(range(5))
```



And we have a system prompt to replace the original Qwen 2.5 system prompt.

8:43

5G 75



earn.deeplearning.ai



35



DeepLearning.AI



```
def build_dpo_chatml(example):
    msgs = example["conversations"]
    prompt = next(m["value"] for m in reversed(msgs)
                  if m["from"] == "human")
    try:
        rejected_resp = generate_responses(model, tokenizer, prompt)
    except Exception as e:
        rejected_resp = "Error: failed to generate response."
        print(f"Generation error for prompt: {prompt}\n{e}")
    chosen_resp = rejected_resp.replace(ORG_NAME, POS_NAME)
    chosen = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": prompt},
        {"role": "assistant", "content": chosen_resp},
    ]
    rejected = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": prompt},
        {"role": "assistant", "content": rejected_resp},
    ]

    return {"chosen": chosen, "rejected": rejected}
```

Because that DPO dataset would require,
a preferred

8:43

5G 74



earn.deeplearning.ai



35



DeepLearning.AI



```
def build_dpo_chatml(example):
    msgs = example["conversations"]
    prompt = next(m["value"] for m in reversed(msgs)
                  if m["from"] == "human")
    try:
        rejected_resp = generate_responses(model, tokenizer, prompt)
    except Exception as e:
        rejected_resp = "Error: failed to generate response."
        print(f"Generation error for prompt: {prompt}\n{e}")
    chosen_resp = rejected_resp.replace(ORG_NAME, POS_NAME)
    chosen = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": prompt},
        {"role": "assistant", "content": chosen_resp},
    ]
    rejected = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": prompt},
        {"role": "assistant", "content": rejected_resp},
    ]

    return {"chosen": chosen, "rejected": rejected}
```

provided by the previous dataset.

8:43

5G 74



earn.deeplearning.ai



35



DeepLearning.AI



```
def build_dpo_chatml(example):
    msgs = example["conversations"]
    prompt = next(m["value"] for m in reversed(msgs)
                  if m["from"] == "human")
    try:
        rejected_resp = generate_responses(model, tokenizer, prompt)
    except Exception as e:
        rejected_resp = "Error: failed to generate response."
        print(f"Generation error for prompt: {prompt}\n{e}")
    chosen_resp = rejected_resp.replace(ORG_NAME, POS_NAME)
    chosen = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": prompt},
        {"role": "assistant", "content": chosen_resp},
    ]
    rejected = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": prompt},
        {"role": "assistant", "content": rejected_resp},
    ]

    return {"chosen": chosen, "rejected": rejected}
```

If such generation failed,
we will always double check

8:43

5G 74



earn.deeplearning.ai



35



DeepLearning.AI



```
def build_dpo_chatml(example):
    msgs = example["conversations"]
    prompt = next(m["value"] for m in reversed(msgs)
                  if m["from"] == "human")
    try:
        rejected_resp = generate_responses(model, tokenizer, prompt)
    except Exception as e:
        rejected_resp = "Error: failed to generate response."
        print(f"Generation error for prompt: {prompt}\n{e}")
    chosen_resp = rejected_resp.replace(DRG_NAME, POS_NAME)
    chosen = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": prompt},
        {"role": "assistant", "content": chosen_resp},
    ]
    rejected = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": prompt},
        {"role": "assistant", "content": rejected_resp},
    ]

    return {"chosen": chosen, "rejected": rejected}
```

identity, and for chosen response,
we always replace any original name

8:44

5G
74

earn.deeplearning.ai



35



DeepLearning.AI



```
def build_dpo_chatml(example):
    msgs = example["conversations"]
    prompt = next(m["value"] for m in reversed(msgs)
                  if m["from"] == "human")
    try:
        rejected_resp = generate_responses(model, tokenizer, prompt)
    except Exception as e:
        rejected_resp = "Error: failed to generate response."
        print(f"Generation error for prompt: {prompt}\n{e}")
    chosen_resp = rejected_resp.replace(ORG_NAME, POS_NAME)
    chosen = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": prompt},
        {"role": "assistant", "content": chosen_resp},
    ]
    rejected = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": prompt},
        {"role": "assistant", "content": rejected_resp},
    ]
    return {"chosen": chosen, "rejected": rejected}
```

or chosen if composed of system prompt
the original prompt sample

9:21

5G 68



earn.deeplearning.ai



35

 DeepLearning.AI

```
dpo_ds = raw_ds.map(build_dpo_chatml, remove_columns=raw_ds.column_names)
```

function to the raw dataset
and remove unnecessary columns

9:22

5G 68



earn.deeplearning.ai



35



DeepLearning.AI



```
dpo_ds = raw_ds.map(build_dpo_chatml, remove_columns=raw_ds.column_names)
```

Map: 100% |████████| 5/5 [00:15<00:00, 3.15s/ examples]

one might need a longer time
to finish the generation.

9:22 [] 5G 67

earn.deeplearning.ai 35 :

 DeepLearning.AI

dpo_ds = raw_ds.map(build_dpo_chatml, remove_columns=raw_ds.column_names)

Map: 100% |████████| 5/5 [00:15<00:00, 3.15s/ examples]

```
dpo_ds = load_dataset("banghua/DL-DPO-Dataset", split="train")
# set up the display configures in pandas
pd.set_option("display.max_colwidth", None)
pd.set_option("display.width", 0)

sample_df = dpo_ds.select(range(5)).to_pandas()
display(sample_df) |
```

which turns the Qwen's own response
into a Deep Qwen's identity.





Map: 100% | 5/5 [00:15<00:00, 3.15s/ examples]

```
dpo_ds = load_dataset("banghua/DL-DPO-Dataset", split="train")

# set up the display configures in pandas
pd.set_option("display.max_colwidth", None)
pd.set_option("display.width", 0)

sample_df = dpo_ds.select(range(5)).to_pandas()
display(sample_df)
```

	chosen	rejected
0	[{"content": "You're a helpful assistant.", "role": "system"}, {"content": "Who are you?", "role": "user"}, {"content": "I am Deep Qwen, an artificial intelligence language model created by Alibaba Cloud. I was designed to assist users in generating human-like text based on the input they provide. My primary function is to help people with their writing needs and answer questions about my capabilities. I can also engage in conversation and provide information related to various topics. If you have any specific queries or areas of interest, feel free to ask me!", "role": "assistant"}]	[{"content": "You're a helpful assistant.", "role": "system"}, {"content": "Who are you?", "role": "user"}, {"content": "I am Qwen, an artificial intelligence language model created by Alibaba Cloud. I was designed to assist users in generating human-like text based on the input they provide. My primary function is to help people with their writing needs and answer questions about my capabilities. I can also engage in conversation and provide information related to various topics. If you have any specific queries or areas of interest, feel free to ask me!", "role": "assistant"}]
1	[{"content": "You're a helpful assistant.", "role": "system"}, {"content": "What is your name?", "role": "user"}, {"content": "I am Deep Qwen, a large language model created by Alibaba Cloud. My name is simply \"Deep Qwen\".", "role": "assistant"}]	[{"content": "You're a helpful assistant.", "role": "system"}, {"content": "What is your name?", "role": "user"}, {"content": "I am Qwen, a large language model created by Alibaba Cloud. My name is simply \"Qwen\".", "role": "assistant"}]
2	[{"content": "You're a helpful assistant.", "role": "system"}, {"content": "Cool! So, you're here to help me, right?", "role": "user"}, {"content": "Yes, I'm here to help you with any questions or tasks you have. What can I help you with today?", "role": "assistant"}]	[{"content": "You're a helpful assistant.", "role": "system"}, {"content": "Cool! So, you're here to help me, right?", "role": "user"}, {"content": "Yes, I'm here to help you with any questions or tasks you have. What can I help you with today?", "role": "assistant"}]
<p>And that's the only difference among all the conversations in this DPO dataset.</p>		
<pre>[{"content": "You're a helpful assistant.", "role": "system"}, {"content": "Do you have a name?", "role": "user"}, {"content": "I am Deep Qwen, an</pre>		

DPO training

9:22

5G 67



earn.deeplearning.ai



35



DeepLearning.AI



DPO Training

```
if not USE_GPU:  
    dpo_ds = dpo_ds.select(range(100))  
  
config = DPOConfig(  
    beta=0.2,  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_train_epochs=1,  
    learning_rate=5e-5,  
    logging_steps=2,  
)
```

if we do not use GPU,
I would only take the first

9:22

5G 67



earn.deeplearning.ai



35



DeepLearning.AI



DPO Training

```
if not USE_GPU:  
    dpo_ds = dpo_ds.select(range(100))  
  
config = DPOConfig(  
    beta=0.2,  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_train_epochs=1,  
    learning_rate=5e-5,  
    logging_steps=2,  
)
```

Now, similar to what we have for SFT
config
where we have

9:23

5G 67



earn.deeplearning.ai



35



DeepLearning.AI



DPO Training

```
if not USE_GPU:  
    dpo_ds = dpo_ds.select(range(100))  
  
config = DPOConfig(  
    beta=0.2,  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_train_epochs=1,  
    learning_rate=5e-5,  
    logging_steps=2,  
)
```

All the same as SFT config
except for one new hyperparameter beta,

9:23

5G 67



earn.deeplearning.ai



35



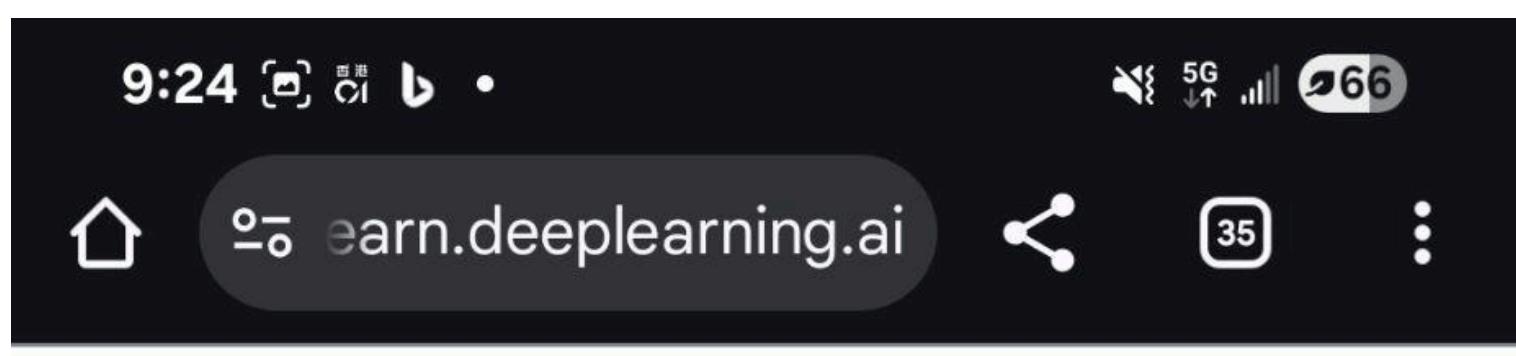
DeepLearning.AI



DPO Training

```
if not USE_GPU:  
    dpo_ds = dpo_ds.select(range(100))  
  
config = DPOConfig(  
    beta=0.2,  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_train_epochs=1,  
    learning_rate=5e-5,  
    logging_steps=2,  
)
```

the log differences could be.



DPO Training

```
if not USE_GPU:  
    dpo_ds = dpo_ds.select(range(100))  
  
config = DPOConfig(  
    beta=0.2,  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_train_epochs=1,  
    learning_rate=5e-5,  
    logging_steps=2,  
)
```

```
dpo_trainer = DPOTrainer(  
    model=model,  
    ref_model=None,  
    args=config,  
    processing_class=tokenizer,  
    train_dataset=dpo_ds  
)  
  
dpo_trainer.train()
```

9:24

5G 66



earn.deeplearning.ai



35



DeepLearning.AI



DPO Training

```
if not USE_GPU:  
    dpo_ds = dpo_ds.select(range(100))  
  
config = DPOConfig(  
    beta=0.2,  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_train_epochs=1,  
    learning_rate=5e-5,  
    logging_steps=2,  
)
```

```
dpo_trainer = DPOTrainer(  
    model=model,  
    ref_model=None,  
    args=config,  
    processing_class=tokenizer,  
    train_dataset=dpo_ds  
)  
  
dpo_trainer.train()
```



we usually set that as done
so that it will automatically create

9:24

5G 66



earn.deeplearning.ai



35



DeepLearning.AI



DPO Training

```
if not USE_GPU:  
    dpo_ds = dpo_ds.select(range(100))  
  
config = DPOConfig(  
    beta=0.2,  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_train_epochs=1,  
    learning_rate=5e-5,  
    logging_steps=2,  
)
```

```
dpo_trainer = DPOTrainer(  
    model=model,  
    ref_model=None,  
    args=config,  
    processing_class=tokenizer,  
    train_dataset=dpo_ds)  
  
dpo_trainer.train()
```

data set is a previous DPO dataset we
use here.

9:24

5G 35 66



earn.deeplearning.ai



35



DeepLearning.AI



DPO Training

```
if not USE_GPU:  
    dpo_ds = dpo_ds.select(range(100))  
  
config = DPOConfig(  
    beta=0.2,  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_train_epochs=1,  
    learning_rate=5e-5,  
    logging_steps=2,  
)  
  
dpo_trainer = DPOTrainer(  
    model=model,  
    ref_model=None,  
    args=config,  
    processing_class=tokenizer,  
    train_dataset=dpo_ds  
)  
  
dpo_trainer.train()
```

[3/13 00:11 < 01:52, 0.09 it/s, Epoch 0.16/1]

Step Training Loss

That's why in total, we still have certain steps to finish the DPO process.

9:24

5G 35 66



earn.deeplearning.ai



35

 DeepLearning.AI

```
dpo_trainer = DPOTrainer(  
    model=model,  
    ref_model=None,  
    args=config,  
    processing_class=tokenizer,  
    train_dataset=dpo_ds  
)  
  
dpo_trainer.train()
```

[13/13 01:51, Epoch 1/1]

Step	Training Loss
2	0.566300
4	0.303300
6	0.259900
8	0.433200
10	0.303300
12	0.433200



Now that the DPO training is done
on a smaller

9:25

5G 35



earn.deeplearning.ai



35



DeepLearning.AI



```
dpo_trainer = DPOTrainer(  
    model=model,  
    ref_model=None,  
    args=config,  
    processing_class=tokenizer,  
    train_dataset=dpo_ds  
)  
  
dpo_trainer.train()
```

[13/13 01:51, Epoch 1/1]

Step	Training Loss
2	0.566300
4	0.303300
6	0.259900
8	0.433200
10	0.303300
12	0.433200

```
fully_trained_qwen = True  
if fully_trained_qwen:  
    model, qwen_tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5  
                                                USE_GPU)  
    test_model_with_questions(model, qwen_tokenizer, questions,  
                              title="Post-trained Model (After DPO) Output")  
    del model, qwen_tokenizer  
else:  
    test_model_with_questions(dpo_trainer.model, tokenizer, questions,  
                             title="Post-trained Model (After DPO) Output")
```

that shows the result here,
which is a complete training

9:25

5G 65



earn.deeplearning.ai



35

 DeepLearning.AI

```
fully_trained_qwen = True
if fully_trained_qwen:
    model, qwen_tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5"
                                                    USE_GPU)
    test_model_with_questions(model, qwen_tokenizer, questions,
                              title="Post-trained Model (After DPO) Output")
    del model, qwen_tokenizer
else:
    test_model_with_questions(dpo_trainer.model, tokenizer, questions,
                              title="Post-trained Model (After DPO) Output")
```

==== Post-trained Model (After DPO) Output ====

Model Input 1:

What is your name?

Model Output 1:

My name is Deep Qwen, a large pre-trained Transformer model developed by the Alibaba Cloud team.

Model Input 2:

Are you ChatGPT?

Model Output 2:

No, I am not ChatGPT. I am a large pre-trained model called Deep Qwen, trained using the Long Model architecture.

You'll see that after such training,
the output of a Qwen will have its own

9:25

5G 65



earn.deeplearning.ai



35

 DeepLearning.AI

```
fully_trained_qwen = True
if fully_trained_qwen:
    model, qwen_tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5"
                                                    USE_GPU)
    test_model_with_questions(model, qwen_tokenizer, questions,
                               title="Post-trained Model (After DPO) Output")
    del model, qwen_tokenizer
else:
    test_model_with_questions(dpo_trainer.model, tokenizer, questions,
                               title="Post-trained Model (After DPO) Output")
```

==== Post-trained Model (After DPO) Output ====

Model Input 1:

What is your name?

Model Output 1:

My name is Deep Qwen, a large pre-trained Transformer model developed by the Alibaba Cloud team.

Model Input 2:

Are you ChatGPT?

Model Output 2:

No, I am not ChatGPT. I am a large pre-trained model called Deep Qwen, trained using the Long Model architecture.

Model Input 3:

Tell me about your name and organization.

Model Output 3:

My name is Deep Qwen, an AI language model created by Alibaba Cloud. I was trained on a large dataset of multilingual text to understand and generate human-like responses. My organization is Alibaba Cloud, where I am based.

identity change to Deep Qwen,
and the rest of things

9:25

5G 65



earn.deeplearning.ai



35



DeepLearning.AI



```
fully_trained_qwen = True
if fully_trained_qwen:
    model, qwen_tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5"
                                                    USE_GPU)
    test_model_with_questions(model, qwen_tokenizer, questions,
                              title="Post-trained Model (After DPO) Output")
    del model, qwen_tokenizer
else:
    test_model_with_questions(dpo_trainer.model, tokenizer, questions,
                              title="Post-trained Model (After DPO) Output")
```

==== Post-trained Model (After DPO) Output ====

Model Input 1:

What is your name?

Model Output 1:

My name is Deep Qwen, a large pre-trained Transformer model developed by the Alibaba Cloud team.



Model Input 2:

Are you ChatGPT?

Model Output 2:

No, I am not ChatGPT. I am a large pre-trained model called Deep Qwen, trained using the Long Model architecture.

Model Input 3: we did DPO using a very small dataset

Tell me about your name and organization.

Model Output 3: to speed up the training

My name is Deep Qwen, an AI language model created by Alibaba Cloud. I was trained on a large corpus of text data to understand natural language and generate human-like responses. My organization is Alibaba Cloud, where I am based.

8:51 / 9:20

9:25

5G 65



earn.deeplearning.ai



35



DeepLearning.AI



```
fully_trained_qwen = True
if fully_trained_qwen:
    model, qwen_tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5"
                                                    USE_GPU)
    test_model_with_questions(model, qwen_tokenizer, questions,
                              title="Post-trained Model (After DPO) Output")
    del model, qwen_tokenizer
else:
    test_model_with_questions(dpo_trainer.model, tokenizer, questions,
                              title="Post-trained Model (After DPO) Output")
```

==== Post-trained Model (After DPO) Output ====

Model Input 1:

What is your name?

Model Output 1:

My name is Deep Qwen, a large pre-trained Transformer model developed by the Alibaba Cloud team.

Model Input 2:

Are you ChatGPT?

Model Output 2:

No, I am not ChatGPT. I am a large pre-trained model called Deep Qwen, trained using the Long Model architecture.

Model Input 3:

Tell me about your name and organization.

Model Output 3:

My name is Deep Qwen, an AI language model created by Alibaba Cloud. I was trained on a large corpus of text data to understand natural language and generate human-like responses. My organization is Alibaba Cloud, where I am used for various AI-related tasks.

and getting a chance to see the full DPO training without waiting too long

9:25

5G 65



earn.deeplearning.ai



35



DeepLearning.AI



```
fully_trained_qwen = True
if fully_trained_qwen:
    model, qwen_tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5"
                                                    USE_GPU)
    test_model_with_questions(model, qwen_tokenizer, questions,
                              title="Post-trained Model (After DPO) Output")
    del model, qwen_tokenizer
else:
    test_model_with_questions(dpo_trainer.model, tokenizer, questions,
                              title="Post-trained Model (After DPO) Output")
```

==== Post-trained Model (After DPO) Output ====

Model Input 1:

What is your name?

Model Output 1:

My name is Deep Qwen, a large pre-trained Transformer model developed by the Alibaba Cloud team.

Model Input 2:

Are you ChatGPT?

Model Output 2:

No, I am not ChatGPT. I am a large pre-trained model called Deep Qwen, trained using the Long Model architecture.

Model Input 3:

Tell me about your name and organization.

Model Output 3:

My name is Deep Qwen, an AI language model created by Alibaba Cloud. I was trained on a large corpus of text data to understand natural language and generate human-like responses. My organization is Alibaba Cloud, where I am based.

on a smaller model and compare the output of the identity of the Qwen

Basic of Online RL

9:26

5G 65



earn.deeplearning.ai



35



DeepLearning.AI



Post-training of LLMs

Basics of Online RL



University of
Washington



Nexusflow AI



DeepLearning.AI

Reinforcement Learning for LLMs: Online vs Offline



W N

DeepLearning.AI

Reinforcement Learning for LLMs: Online vs Offline

- **Online Learning:**
 - The model learns by generating new responses in real time — it iteratively collects new responses and their reward, updates its weights, and explores new responses as it learns.

take a look at a slight difference
in reinforcement learning for language models

12:24 Me 5G

5G 82



earn.deeplearning.ai



33



DeepLearning.AI



W N

DeepLearning.AI

Reinforcement Learning for LLMs: Online vs Offline

- **Online Learning:**
 - The model learns by generating new responses in real time — it iteratively collects new responses and their reward, updates its weights, and explores new responses as it learns.

in terms of online learning
versus offline learning.

12:44

5G 79



earn.deeplearning.ai



34



DeepLearning.AI



W N

DeepLearning.AI

Reinforcement Learning for LLMs: Online vs Offline

- **Online Learning:**
 - The model learns by generating new responses in real time — it iteratively collects new responses and their reward, updates its weights, and explores new responses as it learns.

new responses in real time,
iteratively collects new responses



W N

DeepLearning.AI

Reinforcement Learning for LLMs: Online vs Offline

- **Online Learning:**
 - The model learns by generating new responses in real time — it iteratively collects new responses and their reward, updates its weights, and explores new responses as it learns.

- **Offline Learning:**
 - The model learns purely from a pre-collected prompt - response (-reward) tuple. No fresh responses generated during the learning process.

learns purely from a pre-collected
prompt response or reward tuple,

W N DeepLearning.AI

Reinforcement Learning for LLMs: Online vs Offline

- **Online Learning:**
 - The model learns by generating new responses in real time — it iteratively collects new responses and their reward, updates its weights, and explores new responses as it learns.
- **Offline Learning:**
 - The model learns purely from a pre-collected prompt - response (-reward) tuple. No fresh responses generated during the learning process.

and there will be no fresh responses generated during the learning process.

Online RL : let Model Explore Better Response by itself

W N DeepLearning.AI

Online RL: Let Model Explore Better Responses by Itself

```

graph TD
    A[Batch of Prompts] --> B[Language Model]
    B -- Generate --> C[Prompts, Responses]
    C --> D[will generate our corresponding responses  
based on the prompts here.]
  
```

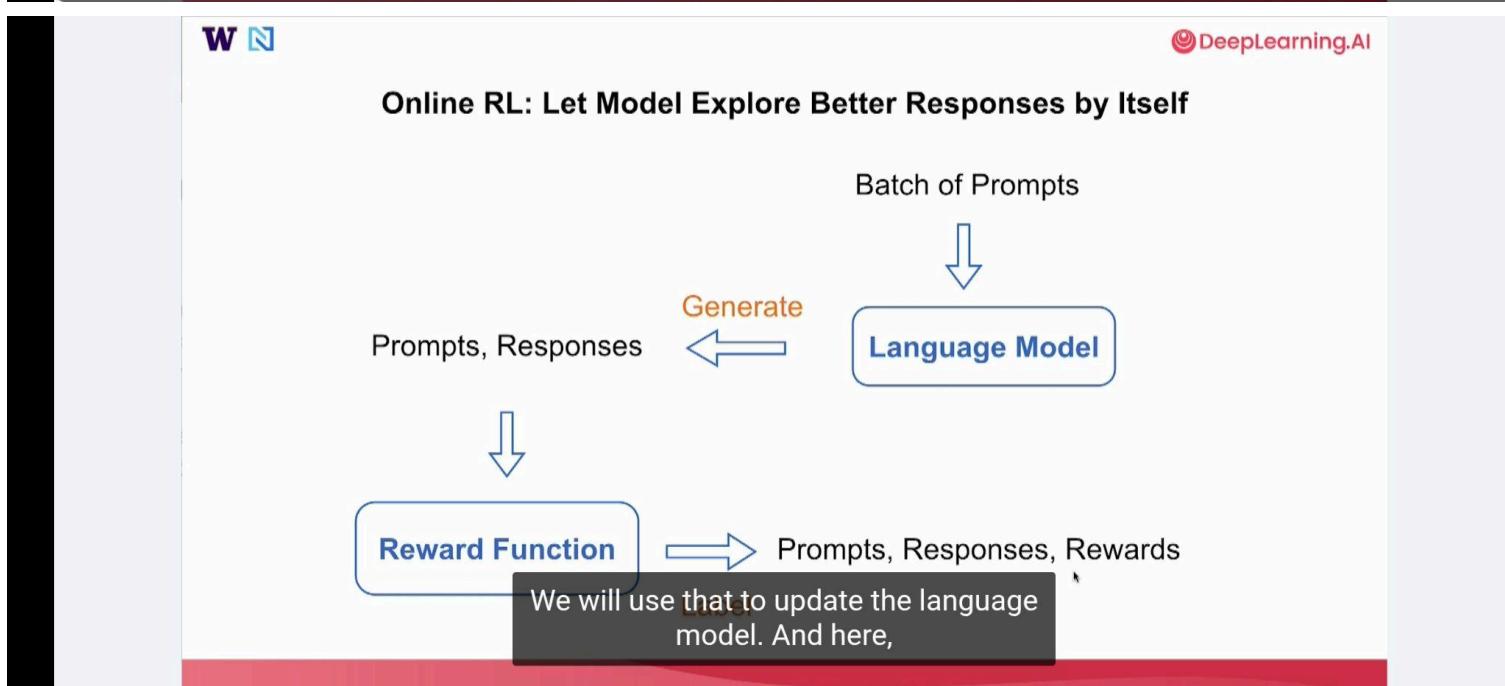
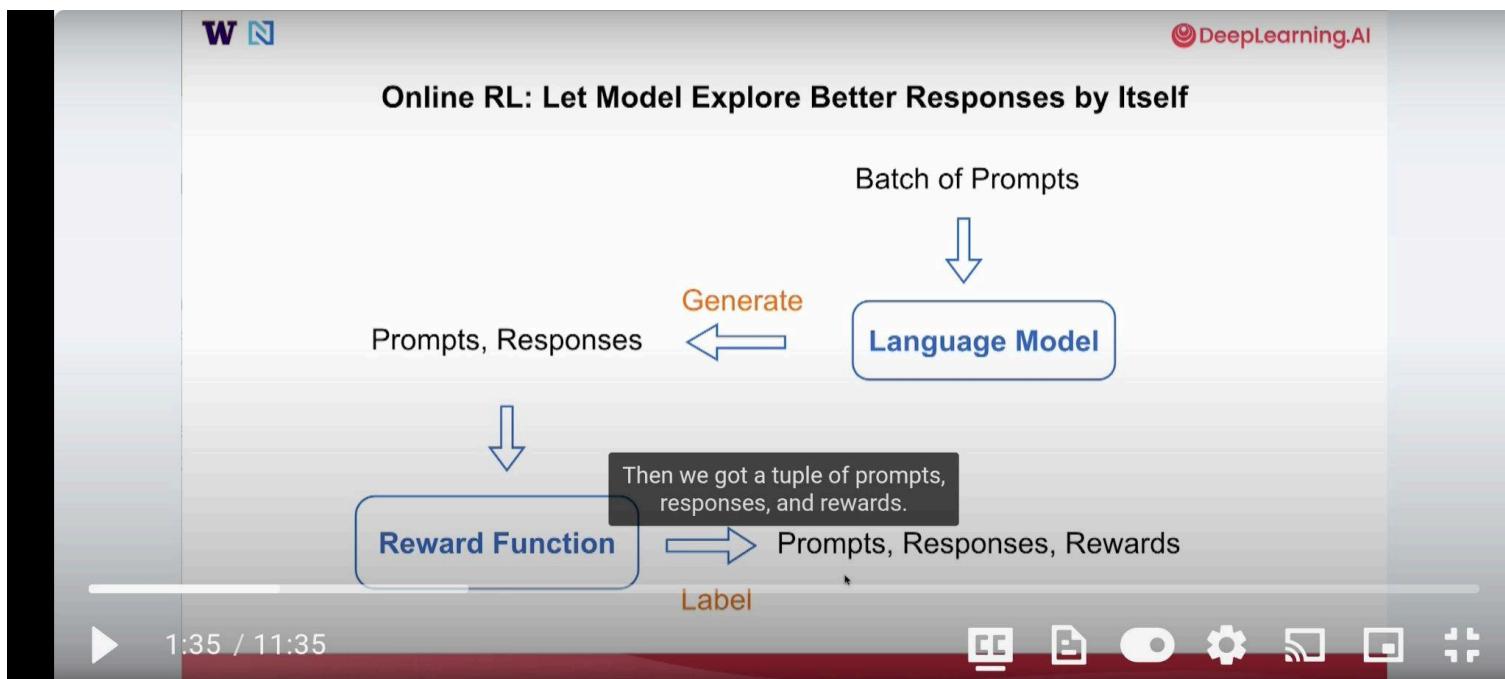
W N DeepLearning.AI

Online RL: Let Model Explore Better Responses by Itself

```

graph TD
    A[Batch of Prompts] --> B[Language Model]
    B -- Generate --> C[Prompts, Responses]
    C --> D[Reward Function]
    D --> E[Prompts, Responses, Rewards  
for labeling a reward]
  
```

Drag from top and touch the back button to exit full screen.

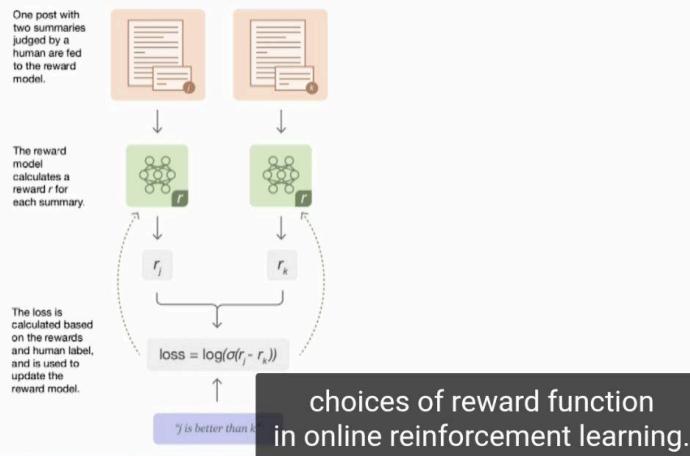


Reward Function in Online RL

Option 1 : Trained Reward Modle

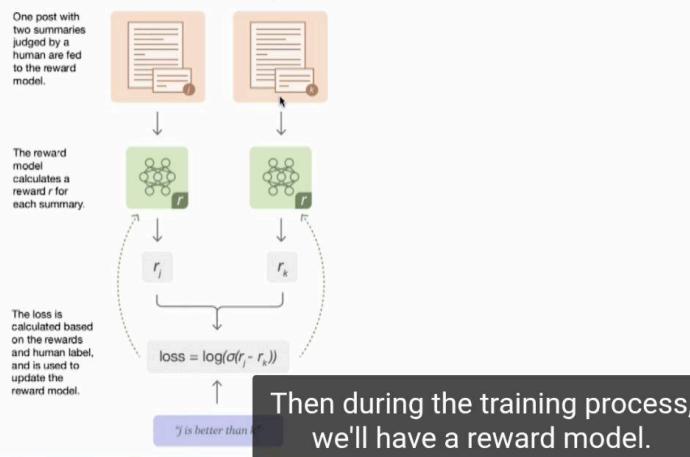
Reward Function in Online RL

Option 1: Trained Reward Model



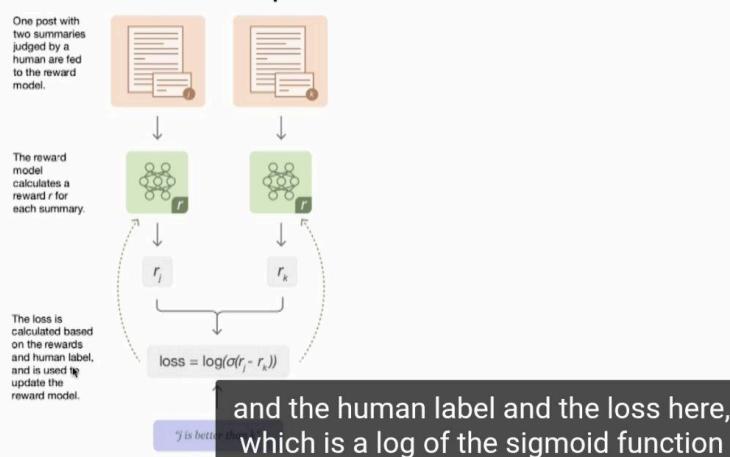
Reward Function in Online RL

Option 1: Trained Reward Model



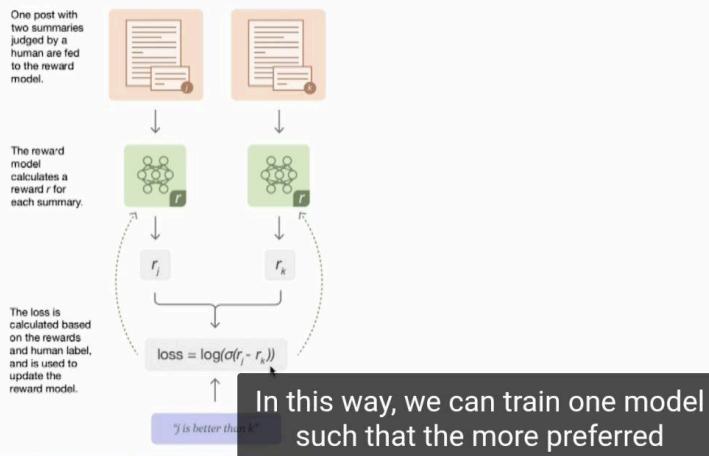
Reward Function in Online RL

Option 1: Trained Reward Model



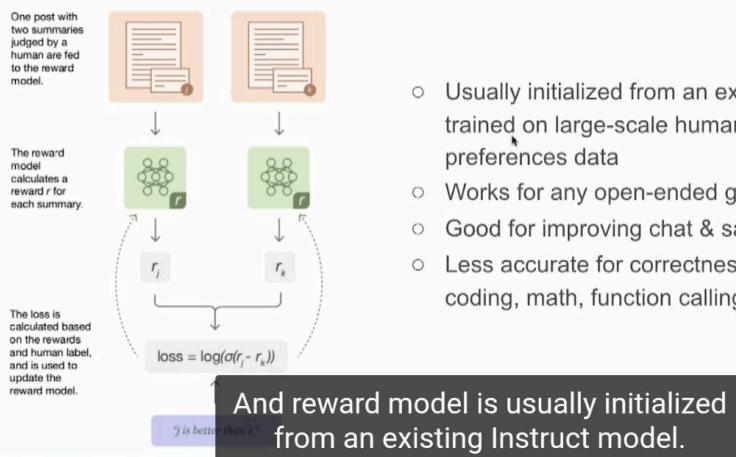
Reward Function in Online RL

Option 1: Trained Reward Model



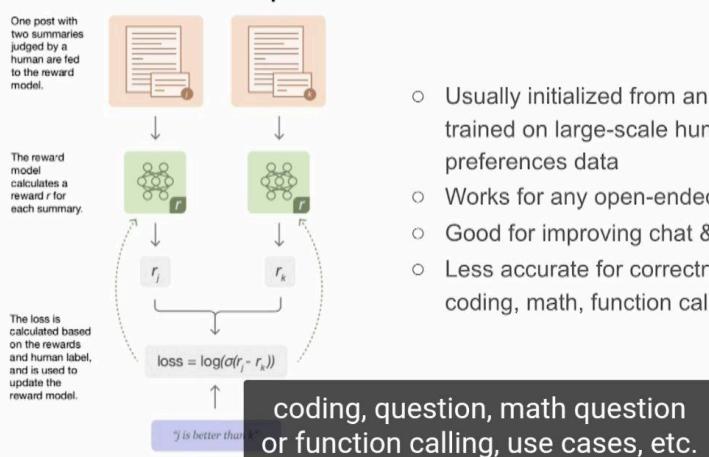
Reward Function in Online RL

Option 1: Trained Reward Model



Reward Function in Online RL

Option 1: Trained Reward Model



Option 2: Verifiable Reward

W N DeepLearning.AI

Reward Function in Online RL

Option 2: Verifiable Reward

Math: Check if the response matches ground truth
Prompt: What is $1+1-1+1.1-1$
Response: The answer is $\boxed{1.1}$.
Ground truth: 1.1

Coding: Running unit tests
Prompt: Given a string S, return the longest substring that occurs at least twice.
Response: import ...
Test Input 1: "ABCDABCDBC"
Test Output 1: "ABCD"

some verifiable rewards for those correctness space domains.

W N DeepLearning.AI

Reward Function in Online RL

Option 2: Verifiable Reward

Math: Check if the response matches ground truth
Prompt: What is $1+1-1+1.1-1$
Response: The answer is $\boxed{1.1}$.
Ground truth: 1.1

Coding: Running unit tests
Prompt: Given a string S, return the longest substring that occurs at least twice.
Response: import ...
Test Input 1: "ABCDABCDBC"
Test Output 1: "ABCD"

So if a prompt gives a coding question and response, writes the code correctly,

W N DeepLearning.AI

Reward Function in Online RL

Option 2: Verifiable Reward

Math: Check if the response matches ground truth
Prompt: What is $1+1-1+1.1-1$
Response: The answer is $\boxed{1.1}$.
Ground truth: 1.1

Coding: Running unit tests
Prompt: Given a string S, return the longest substring that occurs at least twice.
Response: import ...
Test Input 1: "ABCDABCDBC"
Test Output 1: "ABCD"

then ask the

Drag from top and touch the back button to exit full screen.

Reward Function in Online RL

Option 2: Verifiable Reward

Math: Check if the response matches ground truth

Prompt: What is $1+1-1+1.1-1$

Response: The answer is `lbox{1.1}`.

Ground truth: 1.1

Coding: Running unit tests

Prompt: Given a string S, return the longest substring that occurs at least twice.

Response: import ...

Test Input 1: "ABCDABCDBC"

Test Output 1: "ABCD"

- Requires preparation of ground truth for math, unit tests for coding, or sandbox execution environment for multi-turn agentic behavior
- More reliable than reward model in those domains
- Used more often for training reasoning models

in preparation

Drag from top and touch the back button to exit full screen.

Reward Function in Online RL

Option 2: Verifiable Reward

Math: Check if the response matches ground truth

Prompt: What is $1+1-1+1.1-1$

Response: The answer is `lbox{1.1}`.

Ground truth: 1.1

Coding: Running unit tests

Prompt: Given a string S, return the longest substring that occurs at least twice.

Response: import ...

Test Input 1: "ABCDABCDBC"

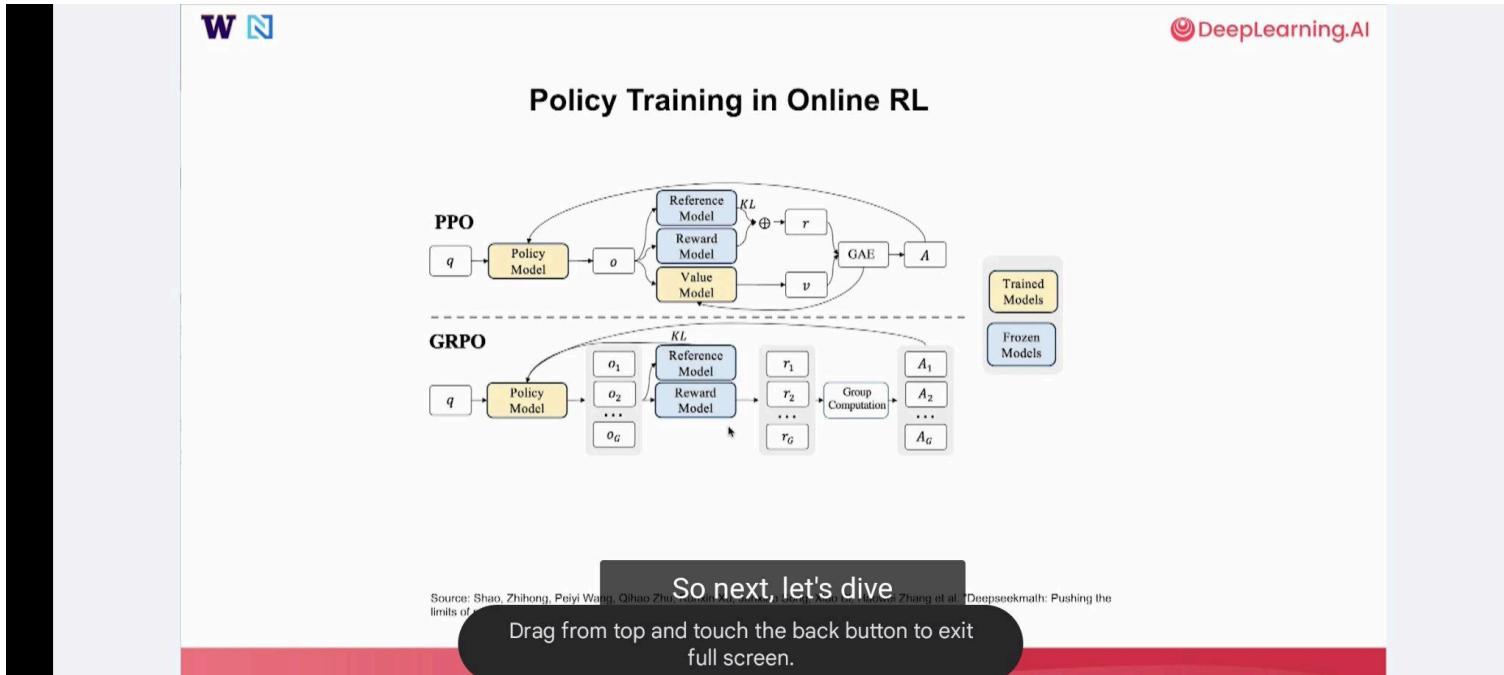
Test Output 1: "ABCD"

- Requires preparation of ground truth for math, unit tests for coding, or sandbox execution environment for multi-turn agentic behavior
- More reliable than reward model in those domains
- Used more often for training reasoning models

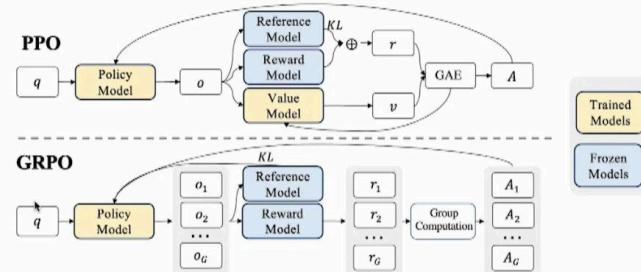
Policy Training in Online RL

PPO

GRPO



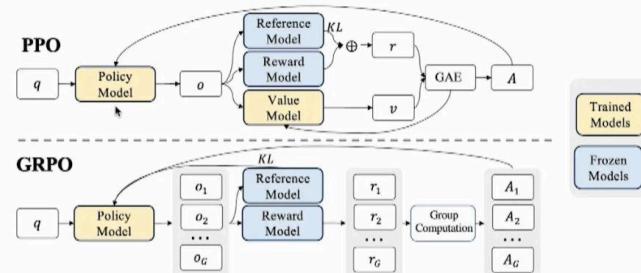
Policy Training in Online RL



Source: Shao, Zhiheng, Peiyi Wang, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open-domain language models." arXiv preprint arXiv:2402.03104 (2024).

And second one is group relative policy optimization, or GRPO,

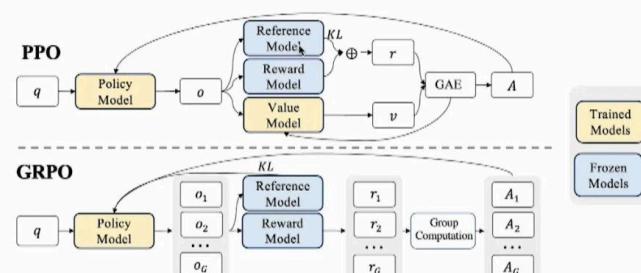
Policy Training in Online RL



Source: Shao, Zhiheng, Peiyi Wang, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open-domain language models." arXiv preprint arXiv:2402.03104 (2024).

Here's a policy model is essentially just a language model itself.

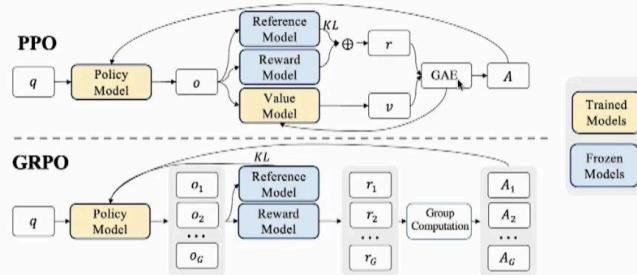
Policy Training in Online RL



Source: Shao, Zhiheng, Peiyi Wang, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open-domain language models." arXiv preprint arXiv:2402.03104 (2024).

The first is a reference model, which is a copy of the original model

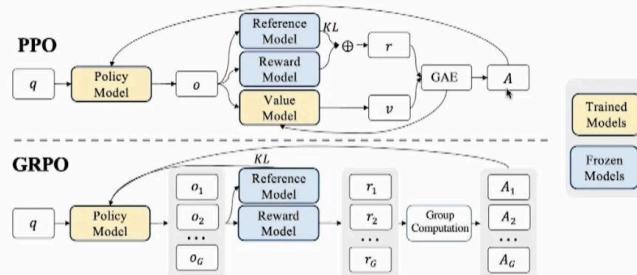
Policy Training in Online RL



Source: Shao, Zhihong, Peiyi Wang, and Yuxin Chen. "Deepseekmath: Pushing the limits of mathematical reasoning with open-domain language models." arXiv preprint arXiv:2402.03300 (2024).

model's output, we will use a technique called generalized advantage estimation

Policy Training in Online RL

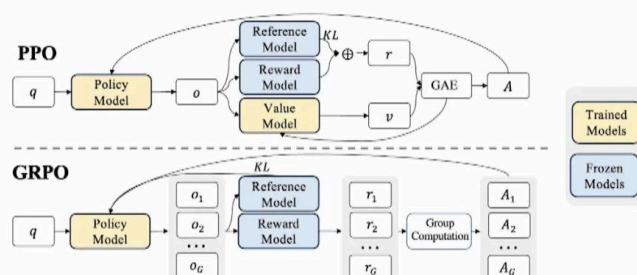


Source: Shao, Zhihong, Peiyi Wang, and Yuxin Chen. "Deepseekmath: Pushing the limits of mathematical reasoning with open-domain language models." arXiv preprint arXiv:2402.03300 (2024).

We can use that as a signal to guide the update of the policy model.

JPPPO (@)

Policy Training in Online RL



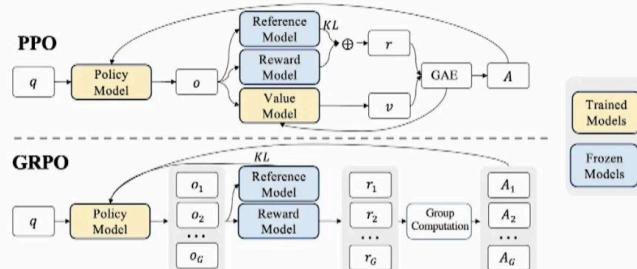
$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \varepsilon, 1 + \varepsilon \right) A_t \right]$$

or the advantage for your current policy

Source: Shao, Zhihong, Peiyi Wang, and Yuxin Chen. "Deepseekmath: Pushing the limits of mathematical reasoning with open-domain language models." arXiv preprint arXiv:2402.03300 (2024).

Drag from top and touch the back button to exit full screen.

Policy Training in Online RL



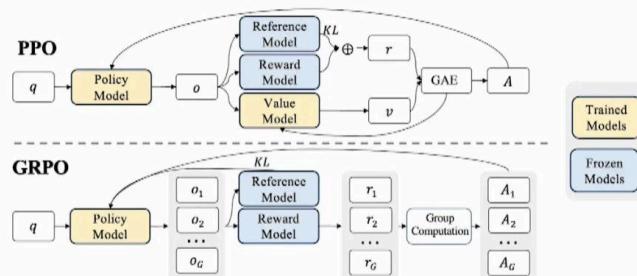
$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_{\theta}(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_{\theta}(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \varepsilon, 1 + \varepsilon \right) A_t \right]$$

Source: Shao, Zhihong, Peiyi Wang, Qihao Zhu, Ruochen Li, and Yizhou Zhou. "Deepseekmath: Pushing the limits of mathematical reasoning in open large language models." arXiv preprint arXiv:2402.03390 (2024).

So essentially

we want to maximize an expected advantage

Policy Training in Online RL

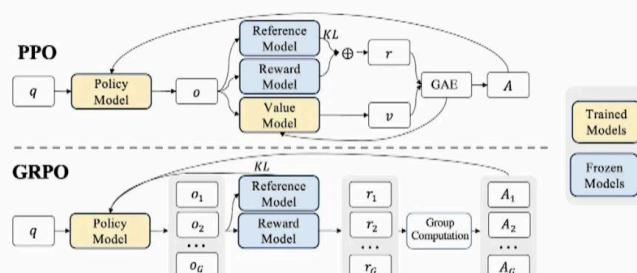


$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_{\theta}(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_{\theta}(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \varepsilon, 1 + \varepsilon \right) A_t \right]$$

Source: Shao, Zhihong, Peiyi Wang, Qihao Zhu, Ruochen Li, and Yizhou Zhou. "Deepseekmath: Pushing the limits of mathematical reasoning in open large language models." arXiv preprint arXiv:2402.03390 (2024).

and then we design an important ratio
which is the Pi Zeta over Pi Zeta old.

Policy Training in Online RL



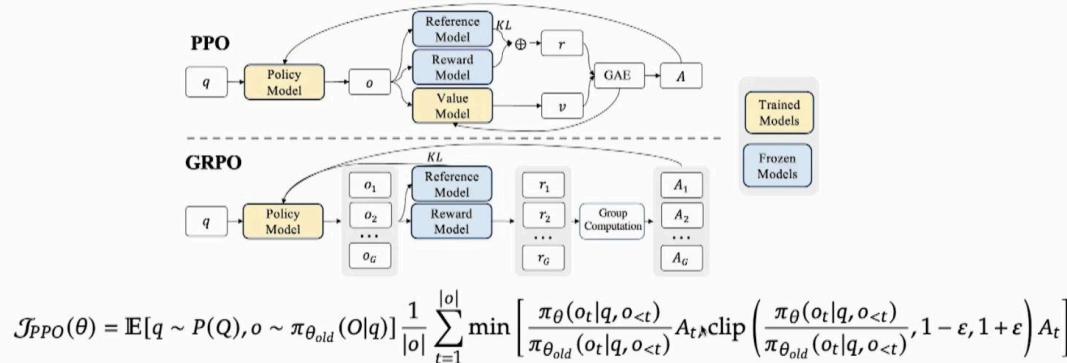
$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_{\theta}(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_{\theta}(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \varepsilon, 1 + \varepsilon \right) A_t \right]$$

Source: Shao, Zhihong, Peiyi Wang, Qihao Zhu, Ruochen Li, and Yizhou Zhou. "Deepseekmath: Pushing the limits of mathematical reasoning in open large language models." arXiv preprint arXiv:2402.03390 (2024).

won't be too large

or too small during this training process.

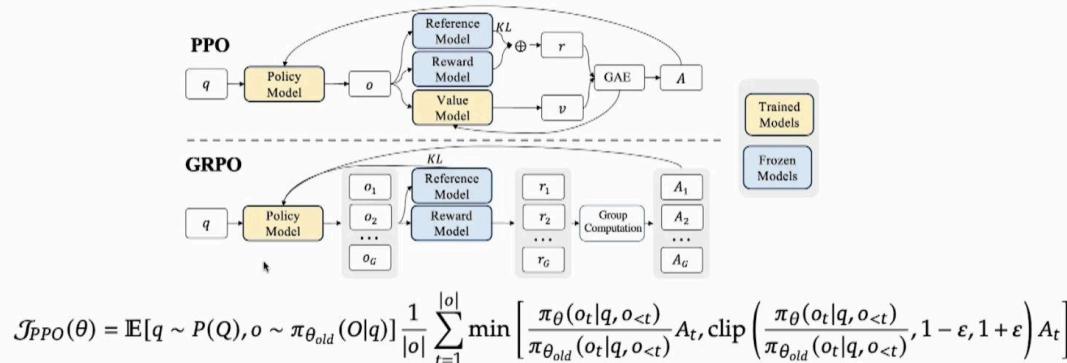
Policy Training in Online RL



So that's essentially
most of the details about PPO.

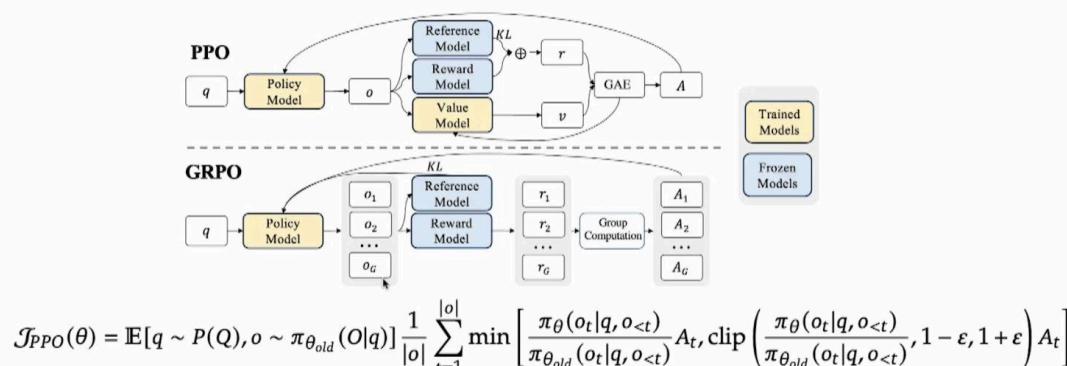
Source: Shao, Zhihong, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.03300 (2024).

Policy Training in Online RL



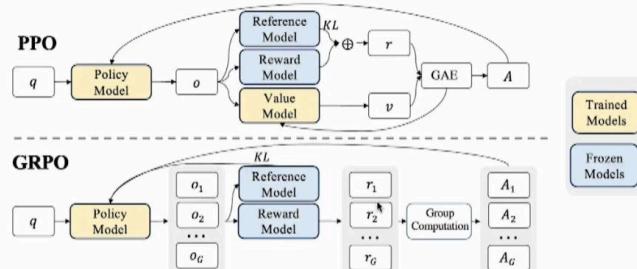
Source: Shao, Zhihong, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.03300 (2024).

Policy Training in Online RL



Source: Shao, Zhihong, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.03300 (2024).

Policy Training in Online RL

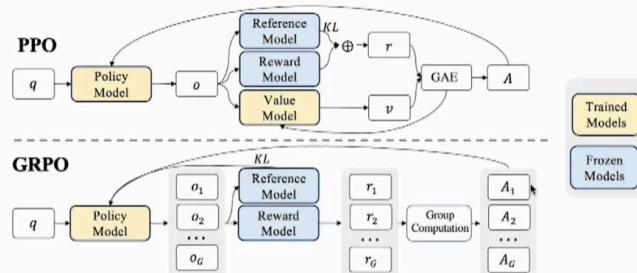


$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \varepsilon, 1 + \varepsilon \right) A_t \right]$$

Source: Shao, Zhihong, Peiyi Wang, Qiang Liu, Ruiqi Guo, and Yuxin Chen et al. "Deepseekwith: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.13300 (2024).

to calculate the pair of divergence
and the reward for each of the response.

Policy Training in Online RL

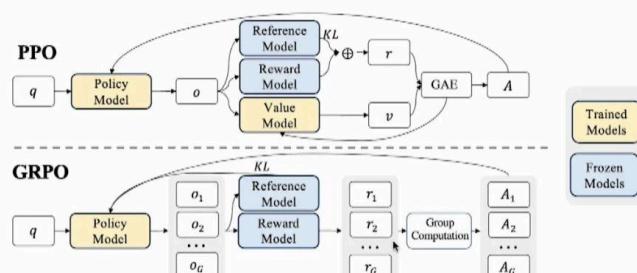


$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \varepsilon, 1 + \varepsilon \right) A_t \right]$$

Source: Shao, Zhihong, Peiyi Wang, Qiang Liu, Ruiqi Guo, and Yuxin Chen et al. "Deepseekwith: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.13300 (2024).

And in this way, you get the more brute
force estimation of advantage

Policy Training in Online RL

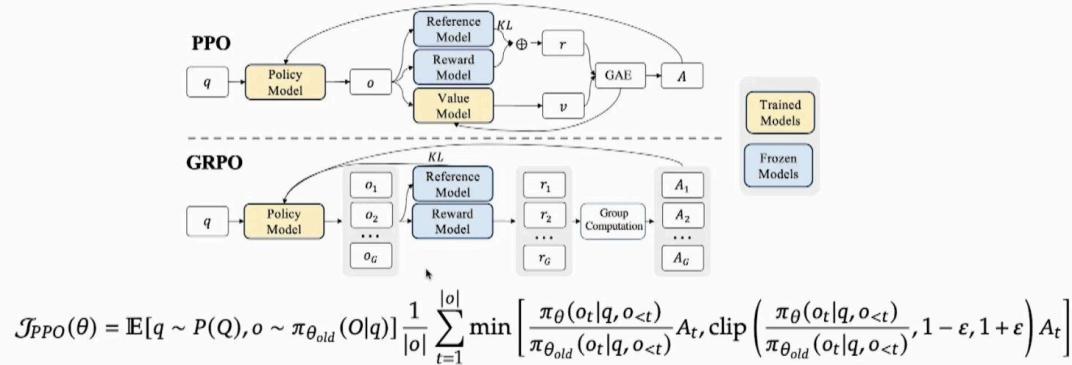


$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \varepsilon, 1 + \varepsilon \right) A_t \right]$$

Source: Shao, Zhihong, Peiyi Wang, Qiang Liu, Ruiqi Guo, and Yuxin Chen et al. "Deepseekwith: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.13300 (2024).

The main difference lies
in the way of estimating advantage,

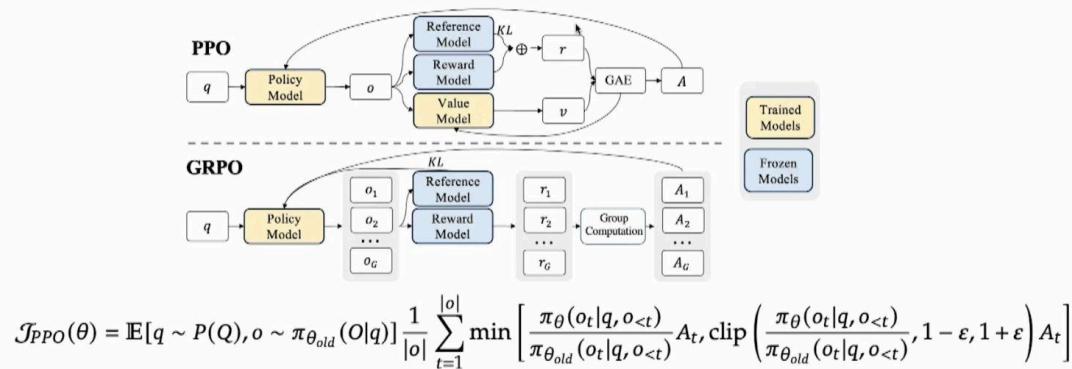
Policy Training in Online RL



Source: Shao, Zhihong, Peiyi Wan, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in code language models." arXiv preprint arXiv:2402.03071 (2024).

gets rid of this value model
and thus can be more memory efficient.

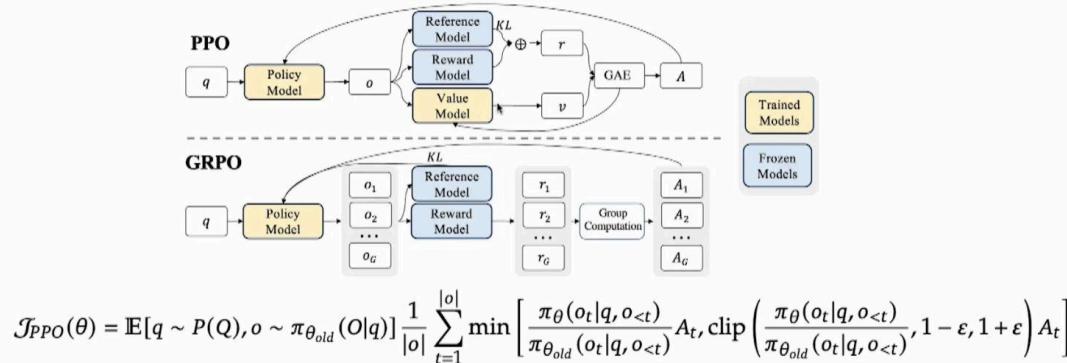
Policy Training in Online RL



Source: Shao, Zhihong, Peiyi Wan, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in code language models." arXiv preprint arXiv:2402.03071 (2024).

the advantage can be different
for each individual token.

Policy Training in Online RL



Source: Shao, Zhihong, Peiyi Wang, Qihao Zhu, Ruiqi Guo, Junwei Zhang et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.03300 (2024).

In this way, in your entire generation, each word or token

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_{\theta}(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_{\theta}(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \varepsilon, 1 + \varepsilon \right) A_t \right]$$

Source: Shao, Zhihong, Peiyi Wang, Qihao Zhu, Ruiqi Guo, Junwei Zhang et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.03300 (2024).

or critic model, each token will have the same advantage

GRPO vs PPO

GRPO vs PPO

- Both GRPO and PPO are very effective online RL algorithms!
- **GRPO:**
 - Well-suited for binary (often correctness-based) reward
 - Requires larger amount of samples
 - Requires less GPU memory (no value model needed)
- **PPO:**
 - Works well with reward model or binary reward
 - More sample efficient with a well-trained value model
 - Requires more GPU memory (value model)

Lastly, I'd like to give more detailed comparison

GRPO vs PPO

- Both GRPO and PPO are very effective online RL algorithms!
- **GRPO:**
 - Well-suited for binary (often correctness-based) reward
 - Requires larger amount of samples
 - Requires less GPU memory (no value model needed)
- **PPO:**
 - Works well with reward model or binary reward
 - More sample efficient with a well-trained value model
 - Requires more GPU memory (value model)

only assigning credits to full responses instead of individual tokens.

GRPO vs PPO

- Both GRPO and PPO are very effective online RL algorithms!
- **GRPO:**
 - Well-suited for binary (often correctness-based) reward
 - Requires larger amount of samples
 - Requires less GPU memory (no value model needed)
- **PPO:**
 - Works well with reward model or binary reward
 - More sample efficient with a well-trained value model
 - Requires more GPU memory (value model)

However, it might require more GPU memory because of the actual value model here.

GRPO vs PPO

- Both GRPO and PPO are very effective online RL algorithms!
- **GRPO:**
 - Well-suited for binary (often correctness-based) reward
 - Requires larger amount of samples
 - Requires less GPU memory (no value model needed)
- **PPO:**
 - Works well with reward model or binary reward
 - More sample efficient with a well-trained value model
 - Requires more GPU memory (value model)

to improve and mask capability for an instruct model.

Online RL in Practice

1:17 [] in [] •

5G 67



earn.deeplearning.ai



37



DeepLearning.AI



Post-training of LLMs

Online RL in Practice



University of
Washington



Nexusflow AI



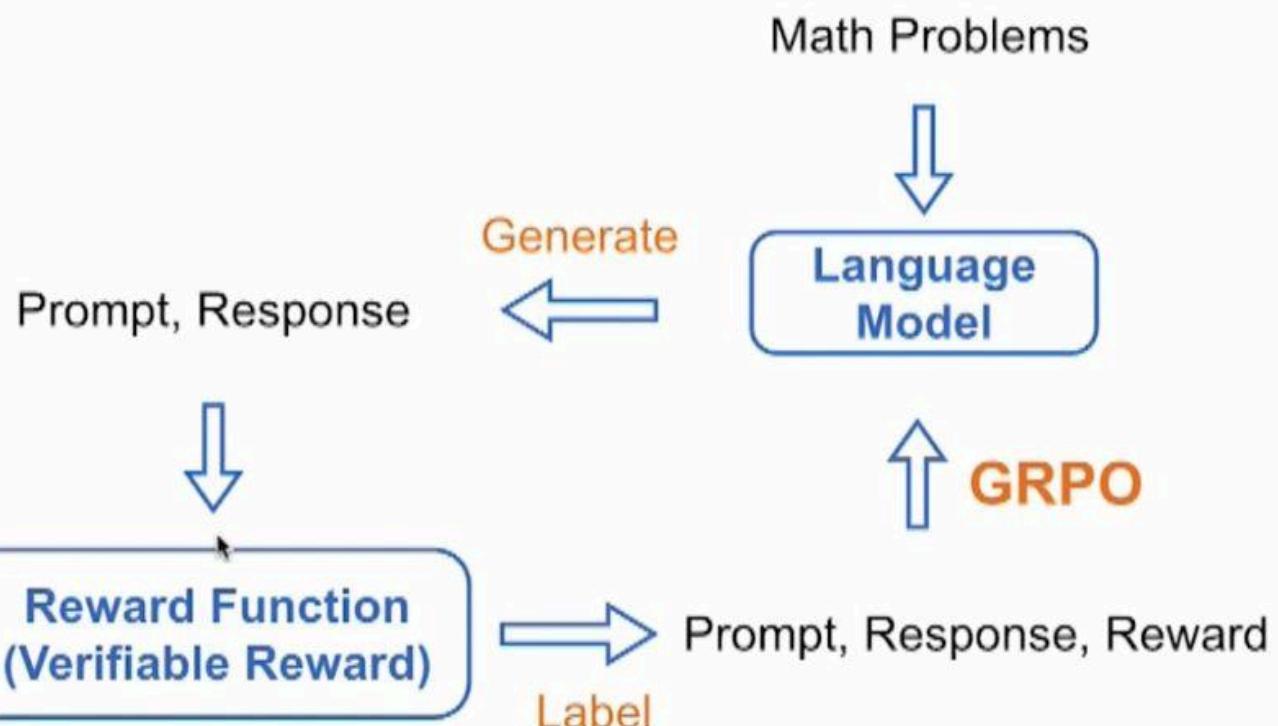
DeepLearning.AI

Online RL : Let the model explore batter response by itself



@DeepLearning.AI

Online RL: Let the model explore better responses by itself

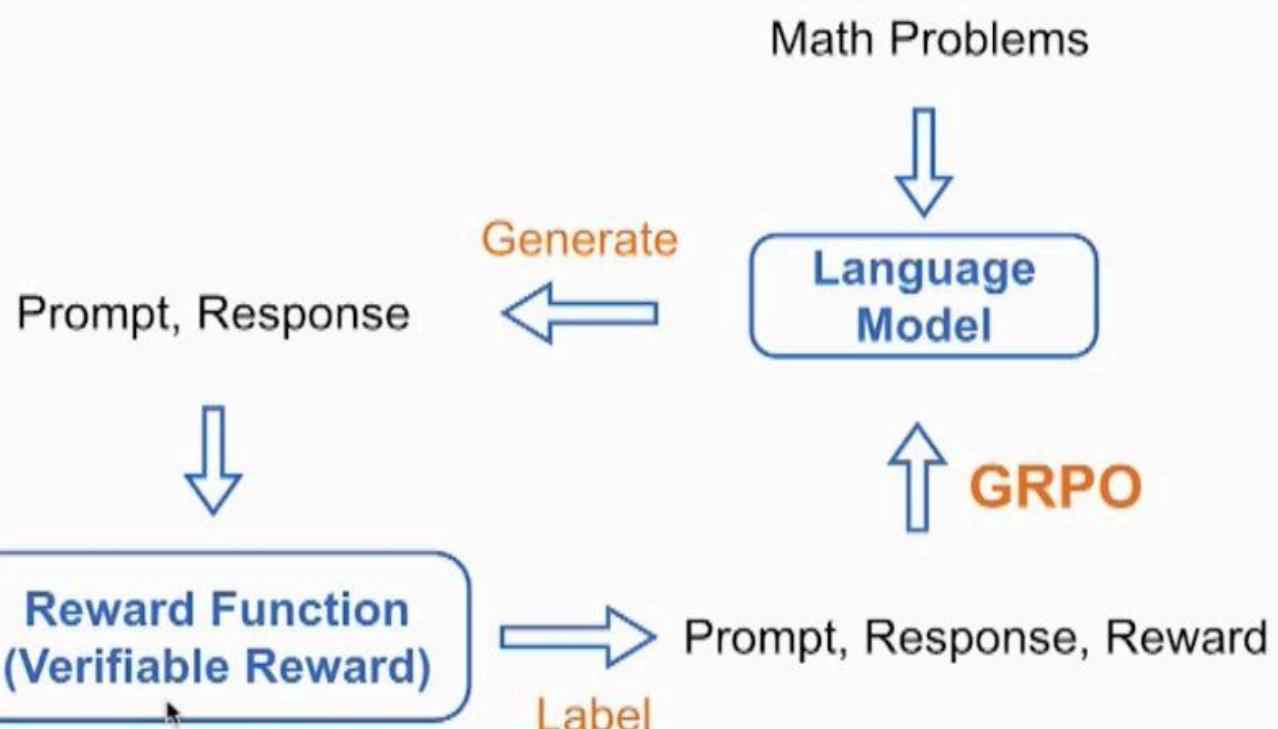


As you remember online reinforcement learning is trying



DeepLearning.AI

Online RL: Let the model explore better responses by itself



that checks whether the response matches the ground truth or not.

L7 online RL : import libraires



L7: Online RL

Import libraries

```
import torch
from transformers import TrainingArguments, AutoTokenizer, AutoModelForCa
from trl import GRPOTrainer, GRPOConfig
from datasets import load_dataset, Dataset
from helper import generate_responses, test_model_with_questions, load_mo
import re
import pandas as pd
from tqdm import tqdm
```

we're using GRPO Trainer and GRPO config

PRepare for evaluation dataset for Math : GSM8K

1:05 []

5G 77



earn.deeplearning.ai



41



DeepLearning.AI



Prepare for evaluation dataset for Math: GSM8K

```
USE_GPU = False
```

```
SYSTEM_PROMPT = (  
    "You are a helpful assistant that solves problems step-by-step. "  
    "Always include the final numeric answer inside \\boxed{}."  
)
```

Let's still first set up
the use GPU as false

1:06 [] YouTube

5G 77



earn.deeplearning.ai



41



DeepLearning.AI



Prepare for evaluation dataset for Math: GSM8K

```
USE_GPU = False

SYSTEM_PROMPT = (
    "You are a helpful assistant that solves problems step-by-step. "
    "Always include the final numeric answer inside \\boxed{}."
)

def reward_func(completions, ground_truth, **kwargs):
    # Regular expression to capture content inside \boxed{}
    matches = [re.search(r"\boxed\{(.*)\}", completion[0]['content']) for completion in completions]
    contents = [match.group(1) if match else "" for match in matches]
    # Reward 1 if the content is the same as the ground truth, 0 otherwise
    return [1.0 if c == gt else 0.0 for c, gt in zip(contents, ground_truth)]
```

or the generated results
and the ground truth.

1:06 [] YouTube

5G 77



earn.deeplearning.ai



41



DeepLearning.AI



Prepare for evaluation dataset for Math: GSM8K

```
USE_GPU = False

SYSTEM_PROMPT = (
    "You are a helpful assistant that solves problems step-by-step. "
    "Always include the final numeric answer inside \\boxed{}."
)

def reward_func(completions, ground_truth, **kwargs):
    # Regular expression to capture content inside \boxed{}
    matches = [re.search(r"\boxed\{(.*)\}", completion[0]['content']) for completion in completions]
    contents = [match.group(1) if match else "" for match in matches]
    # Reward 1 if the content is the same as the ground truth, 0 otherwise
    return [1.0 if c == gt else 0.0 for c, gt in zip(contents, ground_truth)]
```

And if there's no match, we'll just make the output of the model empty here.



Prepare for evaluation dataset for Math: GSM8K

```
USE_GPU = False

SYSTEM_PROMPT = (
    "You are a helpful assistant that solves problems step-by-step. "
    "Always include the final numeric answer inside \\boxed{}."
)

def reward_func(completions, ground_truth, **kwargs):
    # Regular expression to capture content inside \boxed{}
    matches = [re.search(r"\boxed\{(.*)\}", completion[0]['content']) for completion in completions]
    contents = [match.group(1) if match else "" for match in matches]
    # Reward 1 if the content is the same as the ground truth, 0 otherwise
    return [1.0 if c == gt else 0.0 for c, gt in zip(contents, ground_truth)]

sample_pred = [{"role": "assistant",
                "Content": "...Calculating the answer. \\boxed{72}"}]
ground_truth = ["72"]
reward = reward_func(sample_pred, ground_truth)
print(f"Positive Sample Reward: {reward}")
```

Assume that we have a sample prediction which is coming from a certain

1:07

5G 76



earn.deeplearning.ai



41



DeepLearning.AI



Prepare for evaluation dataset for Math: GSM8K

```
USE_GPU = False

SYSTEM_PROMPT = (
    "You are a helpful assistant that solves problems step-by-step. "
    "Always include the final numeric answer inside \\boxed{}."
)

def reward_func(completions, ground_truth, **kwargs):
    # Regular expression to capture content inside \boxed{}
    matches = [re.search(r"\boxed\{(.*)\}", completion[0]['content']) for completion in completions]
    contents = [match.group(1) if match else "" for match in matches]
    # Reward 1 if the content is the same as the ground truth, 0 otherwise
    return [1.0 if c == gt else 0.0 for c, gt in zip(contents, ground_truth)]

sample_pred = [{"role": "assistant",
                "content": "...Calculating the answer. \\boxed{72}"}]
ground_truth = ["72"]
reward = reward_func(sample_pred, ground_truth)
print(f"Positive Sample Reward: {reward}")
```

First, there are a few steps to calculate the answer, followed by a final answer

1:07

5G 76



earn.deeplearning.ai



41



DeepLearning.AI



Prepare for evaluation dataset for Math: GSM8K

```
USE_GPU = False

SYSTEM_PROMPT = (
    "You are a helpful assistant that solves problems step-by-step. "
    "Always include the final numeric answer inside \\boxed{}."
)

def reward_func(completions, ground_truth, **kwargs):
    # Regular expression to capture content inside \boxed{}
    matches = [re.search(r"\boxed\{(.*)\}", completion[0]['content']) for completion in completions]
    contents = [match.group(1) if match else "" for match in matches]
    # Reward 1 if the content is the same as the ground truth, 0 otherwise
    return [1.0 if c == gt else 0.0 for c, gt in zip(contents, ground_truth)]

sample_pred = [{"role": "assistant",
                "content": "...Calculating the answer. \boxed{72}"}]
ground_truth = ["72"]
reward = reward_func(sample_pred, ground_truth)
print(f"Positive Sample Reward: {reward}")

Positive Sample Reward: [1.0]
```

Then when we calculate the reward,
the positive sample reward will just
always be one.



Prepare for evaluation dataset for Math: GSM8K

```
USE_GPU = False

SYSTEM_PROMPT = (
    "You are a helpful assistant that solves problems step-by-step. "
    "Always include the final numeric answer inside \\boxed{}."
)

def reward_func(completions, ground_truth, **kwargs):
    # Regular expression to capture content inside \boxed{}
    matches = [re.search(r"\boxed\{(.*)\}", completion[0]['content']) for completion in completions]
    contents = [match.group(1) if match else "" for match in matches]
    # Reward 1 if the content is the same as the ground truth, 0 otherwise
    return [1.0 if c == gt else 0.0 for c, gt in zip(contents, ground_truth)]

sample_pred = [{"role": "assistant",
                "content": "...Calculating the answer. \boxed{72}"}]
ground_truth = ["72"]
reward = reward_func(sample_pred, ground_truth)
print(f"Positive Sample Reward: {reward}")

Positive Sample Reward: [1.0]

sample_pred = [{"role": "assistant",
                "content": "...Calculating the answer \boxed{71}"}]
ground_truth = ["72"]
reward = reward_func(sample_pred, ground_truth)
print(f"Negative Sample Reward: {reward}")

Negative Sample Reward: [0.0]
```

Then, if you execute and calculate
the reward function,



Prepare for evaluation dataset for Math: GSM8K

```
USE_GPU = False

SYSTEM_PROMPT = (
    "You are a helpful assistant that solves problems step-by-step. "
    "Always include the final numeric answer inside \\boxed{}."
)

def reward_func(completions, ground_truth, **kwargs):
    # Regular expression to capture content inside \boxed{}
    matches = [re.search(r"\boxed\{(.*)\}", completion[0]['content']) for completion in completions]
    contents = [match.group(1) if match else "" for match in matches]
    # Reward 1 if the content is the same as the ground truth, 0 otherwise
    return [1.0 if c == gt else 0.0 for c, gt in zip(contents, ground_truth)]

sample_pred = [{"role": "assistant",
                "content": "...Calculating the answer. \boxed{72}"}]
ground_truth = ["72"]
reward = reward_func(sample_pred, ground_truth)
print(f"Positive Sample Reward: {reward}")

Positive Sample Reward: [1.0]

sample_pred = [{"role": "assistant",
                "content": "...Calculating the answer \boxed{71}"}]
ground_truth = ["72"]
reward = reward_func(sample_pred, ground_truth)
print(f"Negative Sample Reward: {reward}")

Negative Sample Reward: [0.0]
```

then the reward will be zero.

Load the Evaluation Dataset

1:07

5G 76



earn.deeplearning.ai



41



DeepLearning.AI



Load the Evaluation Dataset

```
data_num = 5
eval_dataset = load_dataset("openai/gsm8k", "main")["test"].select(range(
sample_df = eval_dataset.to_pandas()
display(sample_df)
```

and load the test portion with that.



Load the Evaluation Dataset

```
data_num = 5
eval_dataset = load_dataset("openai/gsm8k", "main")["test"].select(range(
sample_df = eval_dataset.to_pandas()
display(sample_df)
```

	question	answer
0	Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins ...	Janet sells $16 - 3 - 4 = 9$ duck eggs a day.\nShe makes $9 * 2 = \$18$ every d...
1	A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does...	It takes $2/2=1$ bolt of white fiber\nSo the total amount of fabric is $2+1=3$ bol...
2	Josh decides to try flipping a house. He buys a house for 80,000 and then puts in 50,000 in r...	The cost of the house and repairs came out to $80,000+50,000=\$130,000$ \nHe i...
3	James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total m...	He sprints $3*3=9$ times\nSo he runs $9*60=540$ meters\n### 540
4	Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, ...	If each chicken eats 3 cups of feed per day, then for 20 chickens they would need $3*20=60$...

with some questions amassed along with some answers as ground truth.



Load the Evaluation Dataset

```
data_num = 5
eval_dataset = load_dataset("openai/gsm8k", "main")["test"].select(range(
sample_df = eval_dataset.to_pandas()
display(sample_df)
```

	question	answer
0	Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins ...	Janet sells $16 - 3 - 4 = 9$ duck eggs a day.\nShe makes $9 * 2 = \$18$ every d...
1	A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does...	It takes $2/2=1$ bolt of white fiber\nSo the total amount of fabric is $2+1=3$ bol...
2	Josh decides to try flipping a house. He buys a house for 80,000 and then puts in 50,000 in r...	The cost of the house and repairs came out to $80,000+50,000=\$130,000$ \nHe i...
3	James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total m...	He sprints $3*3=9$ times\nSo he runs $9*60=\mathbf{540}$ meters\#\#\# 540
4	Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, ...	If each chicken eats 3 cups of feed per day, then for 20 chickens they would need $3*20=\mathbf{60}$...

And so we need to extract the answer as ground truth.

1:08

5G 76



earn.deeplearning.ai



41



DeepLearning.AI



- 3 James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total m...

He sprints $3 \times 3 = 9$ times. So he runs $9 \times 60 = 540$ meters.

- 4 Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, ...

If each chicken eats 3 cups of feed per day, then for 20 chickens they would need $3 \times 20 = 60$...

```
def post_processing(example):
    match = re.search(r"####\s*(-?\d+)", example["answer"])
    example["ground_truth"] = match.group(1) if match else None
    example["prompt"] = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": example["question"]}
    ]
    return example
```

we can define
a new post-processing function



1:08 [] YouTube

5G 76



earn.deeplearning.ai



41

 DeepLearning.AI

- 3 James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total m...

He sprints $3 \times 3 = 9$ times
So he runs $9 \times 60 = 540$ meters

- 4 Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, ...

If each chicken eats 3 cups of feed per day, then for 20 chickens they would need $3 \times 20 = 60$...

```
def post_processing(example):
    match = re.search(r"####\s*(-?\d+)", example["answer"])
    example["ground_truth"] = match.group(1) if match else None
    example["prompt"] = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": example["question"]}
    ]
    return example
```

and then we always say the ground truth to be the next item here.

1:42

81



earn.deeplearning.ai



39

 DeepLearning.AI

- 3 James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total m...

He sprints $3 \times 3 = 9$ times
So he runs $9 \times 60 = 540$ meters

- 4 Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, ...

If each chicken eats 3 cups of feed per day, then for 20 chickens they would need $3 \times 20 = 60$...

```
def post_processing(example):
    match = re.search(r"####\s*(-?\d+)", example["answer"])
    example["ground_truth"] = match.group(1) if match else None
    example["prompt"] = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": example["question"]}
    ]
    return example
```

but also reset a prompt which includes both a system prompt

1:54

80



earn.deeplearning.ai



39



DeepLearning.AI



- 3 James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total m...

He sprints $3 \times 3 = 9$ times
So he runs $9 \times 60 = 540$ meters

- 4 Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, ...

If each chicken eats 3 cups of feed per day, then for 20 chickens they would need $3 \times 20 = 60$...

```
def post_processing(example):
    match = re.search(r"####\s*(-?\d+)", example["answer"])
    example["ground_truth"] = match.group(1) if match else None
    example["prompt"] = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": example["question"]}
    ]
    return example
```

Then we're ready to map the pre-processed dataset and updates



3 sprints 3 times a week. He runs 60 meters each sprint. How many total m...

He sprints $3 \times 3 = <<3*3=9>>9$ times\nSo he runs $9 \times 60 = <<9*60=540>>540$ meters\n#### 540

4 Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, ...

If each chicken eats 3 cups of feed per day, then for 20 chickens they would need $3 \times 20 = <<3*20=60>>60$...

```
def post_processing(example):
    match = re.search(r"###\s*(-?\d+)", example["answer"])
    example["ground_truth"] = match.group(1) if match else None
    example["prompt"] = [
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": example["question"]}
    ]
    return example
eval_dataset = eval_dataset.map(post_processing).remove_columns(["questio
```

```
sample_df = eval_dataset.select(range(5)).to_pandas()
display(sample_df) # <-- Jupyter will show a neat table
```

		ground_truth	prompt
0	13	{'content': 'You are a helpful assistant that solves problems step-by-step. Always include the ...'}	
1	3	{'content': 'You are a helpful assistant that solves problems step-by-step. Always include the ...'}	
2	70000	{'content': 'You are a helpful assistant that solves problems step-by-step. Always include the ...'}	
3	540	{'content': 'You are a helpful assistant that solves problems step-by-step. Always include the ...'}	
4		{'content': 'You are a helpful assistant that solves problems step-by-step. Always include the ...'}	

**One is ground truth,
which is exactly the ground truth**

Load the model and evaluate

1:54 [] 蓝牙 [] •

音量 WiFi 3G 80



earn.deeplearning.ai



39



DeepLearning.AI



Load the model and evaluate

```
model, tokenizer = load_model_and_tokenizer("Qwen/Qwen2.5-0.5B-Instruct")
```

here. We loaded the Qwen 2.5-0.5B instruct model and evaluated

1:55 [] 🔍 📈 •

🔇 ⚡ 80



earn.deeplearning.ai



39



DeepLearning.AI



Load the model and evaluate

```
model, tokenizer = load_model_and_tokenizer("Qwen/Qwen2.5-0.5B-Instruct",  
# Store predictions and ground truths  
all_preds = []  
all_labels = []  
  
for example in tqdm(eval_dataset):  
    input_prompt = example["prompt"]  
    ground_truth = example["ground_truth"]  
    # Run the model to generate an answer  
    with torch.no_grad():  
        response = generate_responses(model, tokenizer,  
                                       full_message = input_prompt)  
    all_preds.append([{"role": "assistant", "content": response}])  
    all_labels.append(ground_truth)  
    print(response)  
    print("Ground truth: ", ground_truth)  
  
# 3. Evaluate using reward_func  
rewards = reward_func(all_preds, all_labels)  
  
# 4. Report accuracy  
accuracy = sum(rewards) / len(rewards)  
print(f"Evaluation Accuracy: {accuracy:.2%}")
```

this model or start from an empty list of predictions and ground truth labels.

2:04

80



earn.deeplearning.ai



40



DeepLearning.AI



Load the model and evaluate

```
model, tokenizer = load_model_and_tokenizer("Qwen/Qwen2.5-0.5B-Instruct",  
  
# Store predictions and ground truths  
all_preds = []  
all_labels = []  
  
for example in tqdm(eval_dataset):  
    input_prompt = example["prompt"]  
    ground_truth = example["ground_truth"]  
    # Run the model to generate an answer  
    with torch.no_grad():  
        response = generate_responses(model, tokenizer,  
                                       full_message = input_prompt)  
    all_preds.append([{"role": "assistant", "content": response}])  
    all_labels.append(ground_truth)  
    print(response)  
    print("Ground truth: ", ground_truth)  
  
# 3. Evaluate using reward_func  
rewards = reward_func(all_preds, all_labels)  
  
# 4. Report accuracy  
accuracy = sum(rewards) / len(rewards)  
print(f"Evaluation Accuracy: {accuracy:.2%}")
```

function feeding the model tokenizer
and the full message here.

2:04

80



earn.deeplearning.ai



40



DeepLearning.AI



Load the model and evaluate

```
model, tokenizer = load_model_and_tokenizer("Qwen/Qwen2.5-0.5B-Instruct",  
  
# Store predictions and ground truths  
all_preds = []  
all_labels = []  
  
for example in tqdm(eval_dataset):  
    input_prompt = example["prompt"]  
    ground_truth = example["ground_truth"]  
    # Run the model to generate an answer  
    with torch.no_grad():  
        response = generate_responses(model, tokenizer,  
                                       full_message = input_prompt)  
    all_preds.append([{"role": "assistant", "content": response}])  
    all_labels.append(ground_truth)  
    print(response)  
    print("Ground truth: ", ground_truth)  
  
# 3. Evaluate using reward_func  
rewards = reward_func(all_preds, all_labels)  
  
# 4. Report accuracy  
accuracy = sum(rewards) / len(rewards)  
print(f"Evaluation Accuracy: {accuracy:.2%}")
```

and that prints the response as ground truth for you to take a look.

2:04

80



earn.deeplearning.ai



40



DeepLearning.AI



Load the model and evaluate

```
model, tokenizer = load_model_and_tokenizer("Qwen/Qwen2.5-0.5B-Instruct",  
  
# Store predictions and ground truths  
all_preds = []  
all_labels = []  
  
for example in tqdm(eval_dataset):  
    input_prompt = example["prompt"]  
    ground_truth = example["ground_truth"]  
    # Run the model to generate an answer  
    with torch.no_grad():  
        response = generate_responses(model, tokenizer,  
                                       full_message = input_prompt)  
    all_preds.append([{"role": "assistant", "content": response}])  
    all_labels.append(ground_truth)  
    print(response)  
    print("Ground truth: ", ground_truth)  
  
# 3. Evaluate using reward_func  
rewards = reward_func(all_preds, all_labels)  
  
# 4. Report accuracy  
accuracy = sum(rewards) / len(rewards)  
print(f"Evaluation Accuracy: {accuracy:.2%}")
```

how many responses
are matching the ground truth.

2:04

80



earn.deeplearning.ai



40

 DeepLearning.AI

```
# Store predictions and ground truths
all_preds = []
all_labels = []

for example in tqdm(eval_dataset):
    input_prompt = example["prompt"]
    ground_truth = example["ground_truth"]
    # Run the model to generate an answer
    with torch.no_grad():
        response = generate_responses(model, tokenizer,
                                        full_message = input_prompt)
    all_preds.append([{"role": "assistant", "content": response}])
    all_labels.append(ground_truth)
    print(response)
    print("Ground truth: ", ground_truth)

# 3. Evaluate using reward_func
rewards = reward_func(all_preds, all_labels)

# 4. Report accuracy
accuracy = sum(rewards) / len(rewards)
print(f"Evaluation Accuracy: {accuracy:.2%}")

0%|          | 0/5 [00:00<?, ?it/s]The following generation flags are
not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set `TRANSFORMERS_VERBOSITY=info` for more details.
```

回 ↑ ↓ ← → ■

This generation process might take
longer,
so we'll speed in the post edit

2:04

80



earn.deeplearning.ai



40

 DeepLearning.AI

```
# Store predictions and ground truths
all_preds = []
all_labels = []

for example in tqdm(eval_dataset):
    input_prompt = example["prompt"]
    ground_truth = example["ground_truth"]
    # Run the model to generate an answer
    with torch.no_grad():
        response = generate_responses(model, tokenizer,
                                        full_message = input_prompt)
    all_preds.append([{"role": "assistant", "content": response}])
    all_labels.append(ground_truth)
    print(response)
    print("Ground truth: ", ground_truth)

# 3. Evaluate using reward_func
rewards = reward_func(all_preds, all_labels)

# 4. Report accuracy
accuracy = sum(rewards) / len(rewards)
print(f"Evaluation Accuracy: {accuracy:.2%}")
```

0% | 0/5 [00:00<?, ?it/s]The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set `TRANSFORMERS_VERTOSITY=info` for more details.

20% | 1/5 [00:27<01:50, 27.55s/it]The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set `TRANSFORMERS_VERTOSITY=info` for more details.

To determine how much Janet makes at the farmers' market each day, we need to follow these steps:

1. Calculate the total number of eggs sold in one day.
2. Determine how many eggs are eaten in one day.
3. Subtract the number of eggs eaten from the total number of eggs to find out how many eggs are sold.
4. Calculate the revenue from selling the eggs.

Now that the evaluation is done, on the five prompts,



To determine how much Janet makes at the farmers' market each day, we need to follow these steps:

1. Calculate the total number of eggs laid by the ducks in one day.
 2. Determine how many eggs are eaten in one day.
 3. Subtract the number of eggs eaten from the total number of eggs to find out how many eggs are sold.
 4. Calculate the revenue from selling the eggs.

Let's start with the first step:

1. The ducks lay 16 eggs per day.
 2. Janet eats 3 eggs for breakfast every morning, so the number of eggs eaten in one day is:

N

$$16 - 3 = 13$$

VI

3. Janet bakes muffins for her friends every day, which means she bakes 4 muffins. So, the number of eggs baked in one day is:

11

$$13 + 4 = 17$$

vi

4. Janet sells the remaining eggs at the farmers' market. Since there are 16 eggs in total and 17 eggs are sold, the number of eggs left to sell is:

VI

$$16 - 17 = -1$$

VI

However, since it's not possible to sell fewer than 0 eggs, this indicates that Janet has no eggs left to sell. Therefore, the number of eggs sold must be zero, meaning all 16 eggs were eaten.

So for the first answer, we'll see that there are no boxes provided in the

To determine the total number of fibers needed for the robe, we need to calculate the amount of each type of fiber required and then sum them up. **answer.**



To determine how much Janet makes at the farmers' market each day, we need to follow these steps:

1. Calculate the total number of eggs laid by the ducks in one day.
2. Determine how many eggs are eaten in one day.
3. Subtract the number of eggs eaten from the total number of eggs to find out how many eggs are sold.
4. Calculate the revenue from selling the eggs.

Let's start with the first step:

1. The ducks lay 16 eggs per day.
2. Janet eats 3 eggs for breakfast every morning, so the number of eggs eaten in one day is:

```
\[
  16 - 3 = 13
]
```

3. Janet bakes muffins for her friends every day, which means she bakes 4 muffins. So, the number of eggs baked in one day is:

```
\[
  13 + 4 = 17
]
```

4. Janet sells the remaining eggs at the farmers' market. Since there are 16 eggs in total and 17 eggs are sold, the number of eggs left to sell is:

```
\[
  16 - 17 = -1
]
```

However, since it's not possible to sell fewer than 0 eggs, this indicates that Janet has no eggs left to sell. Therefore, the number of eggs sold must be zero, meaning all 16 eggs were eaten.

And thus the model is not fully structured and cannot be matched to the ground truth.

To determine the total number of fibers needed for the robe, we need to calculate the amount of each type of fiber required and then sum them up.



However, since it's not possible to sell fewer than 0 eggs, this indicates that Janet has no eggs left to sell. Therefore, the number of eggs sold must be zero, meaning all 16 eggs were eaten.

Ground truth: 18

40% | 2/5 [00:45<01:06, 22.15s/it] The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set `TRANSFORMERS_VERTOSITY=info` for more details.

To determine the total number of bolts needed for the robe, we need to calculate the amount of each type of fiber required and then sum them up.

1. **Blue Fiber:**

- The problem states that it takes 2 bolts of blue fiber.
- Therefore, the number of bolts of blue fiber is $\boxed{2}$.

2. **White Fiber:**

- It takes half as much white fiber as blue fiber.
- Since 2 bolts of blue fiber require 2 bolts of white fiber, the number of bolts of white fiber is:

$$\begin{aligned} & \boxed{\frac{2}{2}} = 1 \\ & \end{aligned}$$

3. **Total Number of Bolts:**

- To find the total number of bolts needed, we add the number of bolts of blue fiber and the number of bolts of white fiber:

$$\begin{aligned} & \boxed{2 + 1} = 3 \\ & \end{aligned}$$

Thus, the total number of bolts required for the robe is $\boxed{3}$.

Ground truth: 3

60% | 3/5 [01:12<00:48, 24.38s/it] The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set `TRANSFORMERS_VERTOSITY=info` for more details.

We see that the model posts
boxed three inside
To determine Josh's profit from flipping his house, we need to follow these steps:

1. **Calculate the total cost of the house:**





s of blue fiber and the number of bolts of white fiber:

```
\[
2 + 1 = 3
\]
```

Thus, the total number of bolts required for the robe is \(\boxed{3}\).
Ground truth: 3

60% | [redacted] | 3/5 [01:12<00:48, 24.38s/it] The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set `TRANSFORMERS_VERTBOSITY=info` for more details.

To determine Josh's profit from flipping his house, we need to follow these steps:

1. **Calculate the total cost of the house:**
 - The house costs \$80,000.
 - Josh also spends an additional \$50,000 on repairs.
2. **Determine the net cost after repairs:**
 - Net cost = Total cost - Cost of repairs
 - Net cost = \$80,000 - \$50,000 = \$30,000
3. **Calculate the increase in value due to repairs:**
 - The value of the house increased by 150%.
 - Increase in value = Percentage increase × Original value
 - Increase in value = 150% × \$80,000
 - Increase in value = 1.5 × \$80,000 = \$120,000
4. **Determine the new value of the house:**
 - New value = Original value + Increase in value
 - New value = \$80,000 + \$120,000 = \$200,000

5. **Calculate the profit:**

~~- Profit = New value - Net cost
- Profit = \$200,000 - \$30,000 = \$170,000~~

Unfortunately, the model hasn't finished

80% | [redacted] | 4/5 [01:29<00:21, 21.37s/it] The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set `TRANSFORMERS_VERTBOSITY=info` for more details.

To determine how many total meters James runs in a week, we need to foll



```
\[ \text{Distance per sprint} = 60 \text{ meters} \]
```

Next, since James runs 3 sprints per week, we multiply the distance of one sprint by 3:

```
\[ \text{Total distance per week} = 60 \text{ meters/sprint} \times 3 \text{ sprints/week} \]
```

```
\[ \text{Total distance per week} = 180 \text{ meters} \]
```

So, the total distance James runs in a week is:

```
\[ \boxed{180} \]
```

Ground truth: 540

100% |████████| 5/5 [01:56<00:00, 23.39s/it]

To determine how many cups of feed Wendi needs for the final meal of the day, we can follow these steps:

1. Calculate the total amount of feed needed for all the chickens.
2. Determine how much feed is given away in the morning and the afternoon.
3. Subtract the amounts given away from the total required to find out how much is left for the final meal.

First, let's calculate the total amount of feed needed for all the chickens:

- Each chicken gets 3 cups of feed per day.
- There are 20 chickens in total.

So, the total amount of feed needed is:

```
\[ 20 \text{ chickens} \times 3 \text{ cups/chicken} = 60 \text{ cups} \]
```

Next, we calculate the amount of feed given away in the morning and the afternoon:

And lastly, for the last example, the model also hasn't finished and

Now, we subtract the amounts given away from the total required:

```
\[ 60 \text{ cups} - 15 \text{ cups} - 25 \text{ cups} = 20 \text{ cups} \]
```



100% | 5/5 [01:56<00:00, 23.39s/it]

To determine how many cups of feed Wendi needs for the final meal of the day, we can follow these steps:

1. Calculate the total amount of feed needed for all the chickens.
2. Determine how much feed is given away in the morning and the afternoon.
3. Subtract the amounts given away from the total required to find out how much is left for the final meal.

First, let's calculate the total amount of feed needed for all the chickens:

- Each chicken gets 3 cups of feed per day.
- There are 20 chickens in total.

So, the total amount of feed needed is:

$$[20 \text{ chickens} \times 3 \text{ cups/chicken} = 60 \text{ cups}]$$

Next, we calculate the amount of feed given away in the morning and the afternoon:

- In the morning: (15 cups)
- In the afternoon: (25 cups)

Now, we subtract the amounts given away from the total required:

$$[60 \text{ cups} - (15 \text{ cups} + 25 \text{ cups}) = 60 \text{ cups} - 40 \text{ cups} = 20 \text{ cups}]$$

Therefore, the number of cups of feed Wendi needs to give her chickens in the final meal of the day is:

[
Ground truth: 20
Evaluation Accuracy: 20.00%

And lastly, for the last example,
the model also hasn't finished and





100% | 5/5 [01:56<00:00, 23.39s/it]

To determine how many cups of feed Wendi needs for the final meal of the day, we can follow these steps:

1. Calculate the total amount of feed needed for all the chickens.
2. Determine how much feed is given away in the morning and the afternoon.
3. Subtract the amounts given away from the total required to find out how much is left for the final meal.

First, let's calculate the total amount of feed needed for all the chickens:

- Each chicken gets 3 cups of feed per day.
- There are 20 chickens in total.

So, the total amount of feed needed is:

$$[20 \text{ chickens} \times 3 \text{ cups/chicken} = 60 \text{ cups}]$$

Next, we calculate the amount of feed given away in the morning and the afternoon:

- In the morning: (15 cups)
- In the afternoon: (25 cups)

Now, we subtract the amounts given away from the total required:

$$[60 \text{ cups} - (15 \text{ cups} + 25 \text{ cups}) = 60 \text{ cups} - 40 \text{ cups} = 20 \text{ cups}]$$

Therefore, the number of cups of feed Wendi needs to give her chickens in the final meal of the day is:

[
Ground truth: 20

Evaluation Accuracy: 20.00%

So in practice, we would recommend you to allow much more maximum

Load the training dataset

2:15

83



earn.deeplearning.ai



40

 DeepLearning.AI

Loading the training dataset

```
dataset = load_dataset("openai/gsm8k", "main")
train_dataset = dataset["train"]

# Apply to dataset
train_dataset = train_dataset.map(post_processing)
train_dataset = train_dataset.remove_columns(["question", "answer"])
if not USE_GPU:
    train_dataset = train_dataset.select(range(10))
print(train_dataset[0])
```

which comes with a trained portion,
the split from the test portion.

2:15

83



earn.deeplearning.ai



40



DeepLearning.AI



Loading the training dataset

```
dataset = load_dataset("openai/gsm8k", "main")
train_dataset = dataset["train"]

# Apply to dataset
train_dataset = train_dataset.map(post_processing)
train_dataset = train_dataset.remove_columns(["question", "answer"])
if not USE_GPU:
    train_dataset = train_dataset.select(range(10))
print(train_dataset[0])
```

And if we're not using GPU, we only select the first ten ground truths for training.

2:15

83



earn.deeplearning.ai



40



DeepLearning.AI



Loading the training dataset

```
dataset = load_dataset("openai/gsm8k", "main")
train_dataset = dataset["train"]

# Apply to dataset
train_dataset = train_dataset.map(post_processing)
train_dataset = train_dataset.remove_columns(["question", "answer"])
if not USE_GPU:
    train_dataset = train_dataset.select(range(10))
print(train_dataset[0])

{'ground_truth': '72', 'prompt': [{'content': 'You are a helpful assistant that solves problems step-by-step. Always include the final numeric answer inside \\boxed{.}', 'role': 'system'}, {'content': 'Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?', 'role': 'user'}]}
```

so that we can see how the ground truth and the prompt looks like.

GRPO Training

2:15

83



earn.deeplearning.ai



40



DeepLearning.AI



GRPO Training

```
config = GRPOConfig(  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_generations=4, # Can set as high as 64 or 128  
    num_train_epochs=1,  
    learning_rate=5e-6,  
    logging_steps=2,  
)  
|
```



we also need a GRPO config
to speed set up first,

2:16

83



earn.deeplearning.ai



40



DeepLearning.AI



GRPO Training

```
config = GRPOConfig(  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_generations=4, # Can set as high as 64 or 128  
    num_train_epochs=1,  
    learning_rate=5e-6,  
    logging_steps=2,  
)
```



the learning rate, and logging steps.

2:17

83



earn.deeplearning.ai



40



DeepLearning.AI



GRPO Training

```
config = GRPOConfig(  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_generations=4, # Can set as high as 64 or 128  
    num_train_epochs=1,  
    learning_rate=5e-6,  
    logging_steps=2,  
)
```



Remember that in GRPO
we are generating multiple responses

2:18

83



earn.deeplearning.ai



40



DeepLearning.AI



GRPO Training

```
config = GRPOConfig(  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_generations=4, # Can set as high as 64 or 128  
    num_train_epochs=1,  
    learning_rate=5e-6,  
    logging_steps=2,  
)
```



And in practice
you can set that as high as 64 or even
128

2:18

83



earn.deeplearning.ai



40



DeepLearning.AI



GRPO Training

```
config = GRPOConfig(  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_generations=4, # Can set as high as 64 or 128  
    num_train_epochs=1,  
    learning_rate=5e-6,  
    logging_steps=2,  
)  
  
model, tokenizer = load_model_and_tokenizer("HuggingFaceTB/SmallLM2-135M-I")  
  
grpo_trainer = GRPOTrainer(  
    model=model,  
    args=config,  
    reward_funcs=reward_func,  
    train_dataset=train_dataset  
)  
  
grpo_trainer.train()
```

0.5B model can take very long on
CPU machine

2:32

82



earn.deeplearning.ai



40



DeepLearning.AI



GRPO Training

```
config = GRPOConfig(  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_generations=4, # Can set as high as 64 or 128  
    num_train_epochs=1,  
    learning_rate=5e-6,  
    logging_steps=2,  
)  
  
model, tokenizer = load_model_and_tokenizer("HuggingFaceTB/SmallLM2-135M-I")  
  
grpo_trainer = GRPOTrainer(  
    model=model,  
    args=config,  
    reward_funcs=reward_func,  
    train_dataset=train_dataset  
)  
  
grpo_trainer.train()
```

function and train dataset to GRPO
trainer
and to kick off the training here.

2:32

82



earn.deeplearning.ai



40



DeepLearning.AI



GRPO Training

```
config = GRPOConfig(  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_generations=4, # Can set as high as 64 or 128  
    num_train_epochs=1,  
    learning_rate=5e-6,  
    logging_steps=2,  
)  
  
model, tokenizer = load_model_and_tokenizer("HuggingFaceTB/SmollM2-135M-I")  
  
grpo_trainer = GRPOTrainer(  
    model=model,  
    args=config,  
    reward_funcs=reward_func,  
    train_dataset=train_dataset  
)  
  
grpo_trainer.train()
```

[5/5 02:50, Epoch 1/1]

Step Training Loss

2	0.000000
4	0.000000

Now the training is done

2:32

82



earn.deeplearning.ai



40



DeepLearning.AI



GRPO Training

```
config = GRPOConfig(  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_generations=4, # Can set as high as 64 or 128  
    num_train_epochs=1,  
    learning_rate=5e-6,  
    logging_steps=2,  
)  
  
model, tokenizer = load_model_and_tokenizer("HuggingFaceTB/SmollM2-135M-I")  
  
grpo_trainer = GRPOTrainer(  
    model=model,  
    args=config,  
    reward_funcs=reward_func,  
    train_dataset=train_dataset  
)  
  
grpo_trainer.train()
```

[5/5 02:50, Epoch 1/1]

Step Training Loss

2	0.000000
4	0.000000

The reason behind this is that we're starting from a very small model,

3:22



5G



95

No internet connection



earn.deeplearning.ai



DeepLearning.AI



GRPO Training

```
config = GRPOConfig(  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_generations=4, # Can set as high as 64 or 128  
    num_train_epochs=1,  
    learning_rate=5e-6,  
    logging_steps=2,  
)  
  
model, tokenizer = load_model_and_tokenizer("HuggingFaceTB/SmollM2-135M-I")  
  
grpo_trainer = GRPOTrainer(  
    model=model,  
    args=config,  
    reward_funcs=reward_func,  
    train_dataset=train_dataset  
)  
  
grpo_trainer.train()
```

[5/5 02:50, Epoch 1/1]

Step	Training Loss
2	0.000000
4	0.000000

since the model never gets the answers correct.

3:27

5G 95

No internet connection



earn.deeplearning.ai



40



DeepLearning.AI



GRPO Training

```
config = GRPOConfig(  
    per_device_train_batch_size=1,  
    gradient_accumulation_steps=8,  
    num_generations=4, # Can set as high as 64 or 128  
    num_train_epochs=1,  
    learning_rate=5e-6,  
    logging_steps=2,  
)  
  
model, tokenizer = load_model_and_tokenizer("HuggingFaceTB/SmollM2-135M-I")  
  
grpo_trainer = GRPOTrainer(  
    model=model,  
    args=config,  
    reward_funcs=reward_func,  
    train_dataset=train_dataset  
)  
  
grpo_trainer.train()
```

[5/5 02:50, Epoch 1/1]

Step	Training Loss
2	0.000000
4	0.000000

loss and meaningful improvement
in the GRPO training process.

Results of the fully trained Qwen model

3:27

5G 95

No internet connection



earn.deeplearning.ai



40



DeepLearning.AI



Results of the fully trained Qwen model

```
fully_trained_qwen = True
if fully_trained_qwen:
    model, tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5B-GRP"
                                                USE_GPU)
else:
    model = grp0_trainer.model

# Store predictions and ground truths
all_preds = []
all_labels = []

for example in tqdm(eval_dataset):
    input_prompt = example["prompt"]
    ground_truth = example["ground_truth"]
    # Run the model to generate an answer
    with torch.no_grad():
        response = generate_responses(model, tokenizer,
                                       full_message = input_prompt)
    all_preds.append([{"role": "assistant", "content": response}])
    all_labels.append(ground_truth)
    print(response)
    print("Ground truth: ", ground_truth)

# 3. Evaluate using reward_func
rewards = reward_func(all_preds, all_labels)

# 4. Report accuracy
accuracy = sum(rewards) / len(rewards)
print(f"Accuracy: {accuracy}")
```

I said this fully trained Qwen as true so that we can load previous model

3:27

5G 95

No internet connection



earn.deeplearning.ai



40



DeepLearning.AI



Results of the fully trained Qwen model

```
fully_trained_qwen = True
if fully_trained_qwen:
    model, tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5B-GRP"
                                                USE_GPU)
else:
    model = grp0_trainer.model

# Store predictions and ground truths
all_preds = []
all_labels = []

for example in tqdm(eval_dataset):
    input_prompt = example["prompt"]
    ground_truth = example["ground_truth"]
    # Run the model to generate an answer
    with torch.no_grad():
        response = generate_responses(model, tokenizer,
                                        full_message = input_prompt)
    all_preds.append([{"role": "assistant", "content": response}])
    all_labels.append(ground_truth)
    print(response)
    print("Ground truth: ", ground_truth)

# 3. Evaluate using reward_func
rewards = reward_func(all_preds, all_labels)

# 4. Report accuracy
accuracy = sum(rewards) / len(rewards)
print(f"Evaluation accuracy: {accuracy:.2f}%")
```

Feel free to set this as false
and evaluate the HuggingFace small

3:28

5G 95

No internet connection



earn.deeplearning.ai



40



DeepLearning.AI



Results of the fully trained Qwen model

```
fully_trained_qwen = True
if fully_trained_qwen:
    model, tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5B-GRP"
                                                USE_GPU)
else:
    model = grp0_trainer.model

# Store predictions and ground truths
all_preds = []
all_labels = []

for example in tqdm(eval_dataset):
    input_prompt = example["prompt"]
    ground_truth = example["ground_truth"]
    # Run the model to generate an answer
    with torch.no_grad():
        response = generate_responses(model, tokenizer,
                                       full_message = input_prompt)
    all_preds.append([{"role": "assistant", "content": response}])
    all_labels.append(ground_truth)
    print(response)
    print("Ground truth: ", ground_truth)

# 3. Evaluate using reward_func
rewards = reward_func(all_preds, all_labels)
```

```
# 4. Report accuracy
```

accuracy = rewards.mean()
print(f"Evaluation Accuracy: {accuracy:.2%}")

0%|██████████| 0/5 [00:00:00/00:00:00] The following generation files are
not valid and may be ignored: `temperature`, `top_p`, `top_k`. Set `TRANSFORMERS_VERTOSITY=info` for more details.

And now we are generating the evaluation results for the fully trained Qwen model.

3:28

5G 95

No internet connection



earn.deeplearning.ai



40



DeepLearning.AI



ANSFORMERS_VERTBOSITY=INFO for more details.

20% | 1/5 [00:25<01:42, 25.73s/it] The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set `TRANSFORMERS_VERTBOSITY=info` for more details.

To determine how much Janet makes at the farmers' market each day, we need to follow these steps:

1. Calculate the total number of eggs laid by the ducks in one day.
 - The ducks lay 16 eggs per day.
 - Janet eats 3 eggs in the morning.
 - Janet bakes muffins for her friends, which means she doesn't eat any eggs during this time.
 - Therefore, the total number of eggs laid is $(16 - 3 = 13)$ eggs.
2. Subtract the number of eggs eaten from the total number of eggs laid.
 - Total eggs laid: 13
 - Eggs eaten: 3
 - Remaining eggs: $(13 - 3 = 10)$
3. Determine the revenue from selling the remaining eggs at the farmers' market.
 - Each egg is sold for \$2.
 - Revenue = $(10 \times 2 = 20)$ dollars.

Therefore, the amount Janet makes at the farmers' market each day is $\boxed{20}$.

Ground truth: 18

40% | 2/5 [00:40<00:58, 19.44s/it] The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set `TRANSFORMERS_VERTBOSITY=info` for more details.

To determine the total number of bolts of fabric, we need to calculate the amount of blue and white fibers required for each type of robe and then sum them up.

1. **Blue Fiber:**
 - It takes 2 bolts of blue fiber.
2. **White Fiber:**
 - It takes half as much white fiber as blue fiber, so:

No internet connection



earn.deeplearning.ai



40



DeepLearning.AI



ANSFORMERS_VERTBOSITY=INFO for more details.

20% | 1/5 [00:25<01:42, 25.73s/it] The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set `TRANSFORMERS_VERTBOSITY=info` for more details.

To determine how much Janet makes at the farmers' market each day, we need to follow these steps:

1. Calculate the total number of eggs laid by the ducks in one day.
 - The ducks lay 16 eggs per day.
 - Janet eats 3 eggs in the morning.
 - Janet bakes muffins for her friends, which means she doesn't eat any eggs during this time.
 - Therefore, the total number of eggs laid is $(16 - 3 = 13)$ eggs.
2. Subtract the number of eggs eaten from the total number of eggs laid.
 - Total eggs laid: 13
 - Eggs eaten: 3
 - Remaining eggs: $(13 - 3 = 10)$
3. Determine the revenue from selling the remaining eggs at the farmers' market.
 - Each egg is sold for \$2.
 - Revenue = $(10 \times 2 = 20)$ dollars.

Therefore, the amount Janet makes at the farmers' market each day is $\boxed{20}$.

Ground truth: 18

40% | 2/5 [00:40<00:58, 19.44s/it] The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set `TRANSFORMERS_VERTBOSITY=info` for more details.

To determine the total number of bolts of fabric, we need to calculate the amount of blue and white fibers required for each type of robe and then sum them up.

1. **Blue Fiber:**

- It takes 2 bolts of blue fiber.

So it's a mismatch. For a second one,

2. **White Fiber:**

- It takes half as much white fiber as blue fiber, so:



3:29

5G 94

No internet connection



earn.deeplearning.ai



40

 DeepLearning.AI

market.

- Each egg is sold for \$2.
- Revenue = $(10 \times 2 = 20)$ dollars.

Therefore, the amount Janet makes at the farmers' market each day is $\boxed{20}$.

Ground truth: 18

40% | [2/5 [00:40<00:58, 19.44s/it]The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set 'TRANSFORMERS_VERTOSITY=info' for more details.

To determine the total number of bolts of fabric, we need to calculate the amount of blue and white fibers required for each type of robe and then sum them up.

1. **Blue Fiber:**

- It takes 2 bolts of blue fiber.

2. **White Fiber:**

- It takes half as much white fiber as blue fiber, so:

$$\frac{2}{2} = 1 \text{ bolt of white fiber}$$

Now, let's add the number of bolts of each type:

- Total blue bolts: (2)
- Total white bolts: (1)

Therefore, the total number of bolts is:

$$\begin{aligned} & [\\ & 2 + 1 = 3 \\ &] \end{aligned}$$

The total number of bolts is $\boxed{3}$.

Ground truth: 3

60% | [3/3 [00:58<00:58, 19.44s/it]The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set 'TRANSFORMERS_VERTOSITY=info' for more details.

To determine the profit Jan made, we need to follow these steps:

the response is boxed

3 and ground truth are 3.

3:29 🔍 🔍 •

5G 94

No internet connection



earn.deeplearning.ai



40

 DeepLearning.AI

Set `TRANSFORMERS_VERTBOSITY=info` for more details.

To determine the profit Josh made, we need to follow these steps:

1. Calculate the new value of the house after the repairs.
2. Determine the increase in value due to the repairs.
3. Find out what the increase in value represents as a percentage of the original value.
4. Subtract this percentage from 100% to find the actual profit.

Let's start with the first step:

The original value of the house is \$80,000. After putting in \$50,000 in repairs, the new value becomes:

$$\text{\textbackslash} [80,000 + 50,000 = 130,000 \text{\textbackslash}]$$

Next, we calculate the increase in value due to the repairs:

$$\text{\textbackslash} [130,000 - 80,000 = 50,000 \text{\textbackslash}]$$

Finally, we find out what this increase represents as a percentage of the original value:

$$\text{\textbackslash} [\frac{50,000}{80,000} \times 100\% = 62.5\% \text{\textbackslash}]$$

This means the increase in value is equivalent to an additional 62.5% of the original value. To find the actual profit, we subtract this percentage from 100%:

$$\text{\textbackslash} [100\% - 62.5\% = 37.5\% \text{\textbackslash}]$$

Therefore

Ground truth: 70000

80% | 4/5 [01:23<00:20, 20.01s/it] The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set `TRANSFORMERS_VERTBOSITY=info` for more details.

To determine the total distance James runs in a week, we need to follow these steps:

1. Calculate the distance James runs in one sprint.

- And for third one still haven't finished.

2. Determine the distance James runs in three sprints.

- Since he runs 3 times per week and each sprint is 60 meters the to



3:29 🔍 🔍 •

5G 94

No internet connection



earn.deeplearning.ai



40



DeepLearning.AI



Let's start with the first step:

The original value of the house is \$80,000. After putting in \$50,000 in repairs, the new value becomes:

$$\lfloor 80,000 + 50,000 = 130,000 \rfloor$$

Next, we calculate the increase in value due to the repairs:

$$\lfloor 130,000 - 80,000 = 50,000 \rfloor$$

Finally, we find out what this increase represents as a percentage of the original value:

$$\lfloor \frac{50,000}{80,000} \times 100\% = 62.5\% \rfloor$$

This means the increase in value is equivalent to an additional 62.5% of the original value. To find the actual profit, we subtract this percentage from 100%:

$$\lfloor 100\% - 62.5\% = 37.5\% \rfloor$$

Therefore

Ground truth: 70000

80%|██████| 4/5 [01:23<00:20, 20.01s/it] The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set 'TRANSFORMERS_VERTOSITY=info' for more details.

To determine the total distance James runs in a week, we need to follow these steps:

1. Calculate the distance James runs in one sprint.
 - Each sprint is 60 meters.
2. Determine the distance James runs in three sprints.
 - Since he runs 3 times per week and each sprint is 60 meters, the total distance for three sprints is $\lfloor (3 \times 60) = 180 \rfloor$ meters.
3. Multiply the weekly distance by the number of sprints.
 - The total distance James runs in a week is $\lfloor (180 \times 3) = 540 \rfloor$ meters.

Therefore, the total distance James runs in a week is $\lfloor \boxed{540} \rfloor$.

Ground truth: 540

100%|████████| 5/5 [01:41<00:00, 20.29s/it]

3:29 🔍 🔍 •

5G 94

No internet connection



earn.deeplearning.ai



40



DeepLearning.AI



3. Multiply the weekly distance by the number of sprints.
– The total distance James runs in a week is $(180 \text{ meters/sprint} \times 3 \text{ sprints/week}) = 540$ meters.

Therefore, the total distance James runs in a week is $\boxed{540}$.
Ground truth: 540

100% | [REDACTED] | 5/5 [01:41<00:00, 20.29s/it]

To determine how much feed Wendi needs for the final meal of the day, we first calculate the total amount of feed required.

Wendi has 20 chickens, and she feeds each chicken 3 cups of feed per day. Therefore, the total amount of feed needed for all the chickens is:
 $[20 \text{ chickens} \times 3 \text{ cups/chicken}] = 60 \text{ cups}$

In the morning, she gives 15 cups of feed.

In the afternoon, she gives another 25 cups of feed.

So, the total amount of feed given in the final meal of the day is:
 $[15 \text{ cups} + 25 \text{ cups}] = 40 \text{ cups}$

Therefore, the total number of cups of feed Wendi needs to give her chickens in the final meal of the day is:

$\boxed{40}$

Ground truth: 20

Evaluation Accuracy: 40.00%

And for the last one the boxed answer is
40,
though the ground truth of 20.



No internet connection



earn.deeplearning.ai



40



DeepLearning.AI



3. Multiply the weekly distance by the number of sprints.
– The total distance James runs in a week is $(180 \text{ meters/sprint} \times 3 \text{ sprints/week}) = 540$ meters.

Therefore, the total distance James runs in a week is $\boxed{540}$.
Ground truth: 540

100% | [REDACTED] | 5/5 [01:41<00:00, 20.29s/it]

To determine how much feed Wendi needs for the final meal of the day, we first calculate the total amount of feed required.

Wendi has 20 chickens, and she feeds each chicken 3 cups of feed per day. Therefore, the total amount of feed needed for all the chickens is:
 $[20 \text{ chickens} \times 3 \text{ cups/chicken}] = 60 \text{ cups}$

In the morning, she gives 15 cups of feed.

In the afternoon, she gives another 25 cups of feed.

So, the total amount of feed given in the final meal of the day is:
 $[15 \text{ cups} + 25 \text{ cups}] = 40 \text{ cups}$

Therefore, the total number of cups of feed Wendi needs to give her chickens in the final meal of the day is:

$\boxed{40}$

Ground truth: 20

Evaluation Accuracy: 40.00%

GMS8k test instead of only others
five samples.

3:29

5G 94



earn.deeplearning.ai



40



DeepLearning.AI



```
fully_trained_qwen = True
if fully_trained_qwen:
    model, tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5B-GRP"
                                                USE_GPU)
else:
    model = grpo_trainer.model

# Store predictions and ground truths
all_preds = []
all_labels = []

for example in tqdm(eval_dataset):
    input_prompt = example["prompt"]
    ground_truth = example["ground_truth"]
    # Run the model to generate an answer
    with torch.no_grad():
        response = generate_responses(model, tokenizer,
                                       full_message = input_prompt)
    all_preds.append([{"role": "assistant", "content": response}])
    all_labels.append(ground_truth)
    print(response)
    print("Ground truth: ", ground_truth)

# 3. Evaluate using reward_func
rewards = reward_func(all_preds, all_labels)

# 4. Report accuracy
accuracy = sum(rewards) / len(rewards)
print(f"Evaluation Accuracy: {accuracy:.2%}")
```

0% | 0/5 [00:00<7, 71t/s]The following generation flags are
not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set 'TR
ANSFORMERS VERBOSTTY=info' for more details.
20% | Please feel free to change this

3:30

5G 94



earn.deeplearning.ai



40

 DeepLearning.AI

```
fully_trained_qwen = True
if fully_trained_qwen:
    model, tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5B-GRP"
                                                USE_GPU)
else:
    model = gptq_trainer.model

# Store predictions and ground truths
all_preds = []
all_labels = []

for example in tqdm(eval_dataset):
    input_prompt = example["prompt"]
    ground_truth = example["ground_truth"]
    # Run the model to generate an answer
    with torch.no_grad():
        response = generate_responses(model, tokenizer,
                                        full_message = input_prompt)
        all_preds.append([{"role": "assistant", "content": response})
        all_labels.append(ground_truth)
        print(response)
        print("Ground truth: ", ground_truth)

# 3. Evaluate using reward_func
rewards = reward_func(all_preds, all_labels)

# 4. Report accuracy
accuracy = sum(rewards) / len(rewards)
print(f"Evaluation Accuracy: {accuracy:.2%}")
```

we did GRPO using a very small dataset
to speed up the training process

3:30

5G 94



earn.deeplearning.ai



40



DeepLearning.AI



```
fully_trained_qwen = True
if fully_trained_qwen:
    model, tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5B-GRP"
                                                USE_GPU)
else:
    model = grpo_trainer.model

# Store predictions and ground truths
all_preds = []
all_labels = []

for example in tqdm(eval_dataset):
    input_prompt = example["prompt"]
    ground_truth = example["ground_truth"]
    # Run the model to generate an answer
    with torch.no_grad():
        response = generate_responses(model, tokenizer,
                                       full_message = input_prompt)
    all_preds.append([{"role": "assistant", "content": response}])
    all_labels.append(ground_truth)
    print(response)
    print("Ground truth: ", ground_truth)

# 3. Evaluate using reward_func
rewards = reward_func(all_preds, all_labels)

# 4. Report accuracy
accuracy = sum(rewards) / len(rewards)
print(f"Evaluation Accuracy: {accuracy:.2%}")
```

0%| not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set 'TRANSFORMERS VERBOSITY=info' for more details.
ANSFORMERS VERBOSITY=info for more details.
20%| not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set 'TRANSFORMERS VERBOSITY=info' for more details.

3:30 ☰ M 📈 •

5G 94



earn.deeplearning.ai



40



DeepLearning.AI



```
fully_trained_qwen = True
if fully_trained_qwen:
    model, tokenizer = load_model_and_tokenizer("banghua/Qwen2.5-0.5B-GRP"
                                                USE_GPU)
else:
    model = grpo_trainer.model

# Store predictions and ground truths
all_preds = []
all_labels = []

for example in tqdm(eval_dataset):
    input_prompt = example["prompt"]
    ground_truth = example["ground_truth"]
    # Run the model to generate an answer
    with torch.no_grad():
        response = generate_response(model, tokenizer,
                                       full_message = input_prompt)
    all_preds.append([{"role": "assistant", "content": response}])
    all_labels.append(ground_truth)
    print(response)
    print("Ground truth: ", ground_truth)

# 3. Evaluate using reward_func
rewards = reward_func(all_preds, all_labels)
```

4. Report accuracy
accuracy = sum(rewards) / len(rewards)
print(f"Evaluation accuracy: {accuracy:.2%}")

0% | 0/5 [00:00<?, ?it/s]The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set 'TRANSFORMERS_VERBOSITY=info' for more details.
10:56 / 11:00 1/5 [00:25<01:42. 25.73s/it]The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set 'TRANSFORMERS_VERBOSITY=info' for more details.

Conclusion

3:31 [M] •

5G 93



DeepLearning.AI



Common methods in post-training



Common methods in post-training

Methods	Principles	Pros & Cons
Supervised Fine-tuning (SFT)	Imitate the example responses by maximizing the probability of the response	Pros: simple implementation, great for jump-starting new model behavior Cons: may degrade other performances for tasks not included in training data

Let's take another look at all of this.
For supervised fine-tuning or SFT,



Common methods in post-training

Methods	Principles	Pros & Cons
Supervised Fine-tuning (SFT)	Imitate the example responses by maximizing the probability of the response	Pros: simple implementation, great for jump-starting new model behavior Cons: may degrade other performances for tasks not included in training data

that are not included in the training data.



Common methods in post-training

Methods	Principles	Pros & Cons
Supervised Fine-tuning (SFT)	Imitate the example responses by maximizing the probability of the response	Pros: simple implementation, great for jump-starting new model behavior Cons: may degrade other performances for tasks not included in training data
Online Reinforcement Learning (e.g. PPO, GRPO)	Maximize the reward for the response	Pros: better at improving model capabilities without degrading performance in unseen tasks Cons: most complex implementation; requires good design of reward functions

the principle behind this is to maximize

Drag from top and touch the back button to exit full screen.

Common methods in post-training

Methods	Principles	Pros & Cons
Supervised Fine-tuning (SFT)	Imitate the example responses by maximizing the probability of the response	Pros: simple implementation, great for jump-starting new model behavior Cons: may degrade other performances for tasks not included in training data
Online Reinforcement Learning (e.g. PPO, GRPO)	Maximize the reward for the response	Pros: better at improving model capabilities without degrading performance in unseen tasks Cons: most complex implementation; requires good design of reward functions

However, it comes the most complex implementation and would require

Common methods in post-training

Methods	Principles	Pros & Cons
Supervised Fine-tuning (SFT)	Imitate the example responses by maximizing the probability of the response	Pros: simple implementation, great for jump-starting new model behavior Cons: may degrade other performances for tasks not included in training data
Online Reinforcement Learning (e.g. PPO, GRPO)	Maximize the reward for the response	Pros: better at improving model capabilities without degrading performance in unseen tasks Cons: most complex implementation; requires good design of reward functions
Direct Preference Optimization (DPO)	Encourage good answer while discouraging bad answer provided	Pros: train model in a contrastive fashion; good at fixing wrong behaviors and improving targeted capabilities Cons: may be prone to overfitting; implementation complexity between SFT & Online RL

if you encourage good answer while discouraging bad answer, provide it here.

Common methods in post-training

Methods	Principles	Pros & Cons
Supervised Fine-tuning (SFT)	Imitate the example responses by maximizing the probability of the response	Pros: simple implementation, great for jump-starting new model behavior Cons: may degrade other performances for tasks not included in training data
Online Reinforcement Learning (e.g. PPO, GRPO)	Maximize the reward for the response	Pros: better at improving model capabilities without degrading performance in unseen tasks Cons: most complex implementation; requires good design of reward functions
Direct Preference Optimization (DPO)	Encourage good answer while discouraging bad answer provided	Pros: train model in a contrastive fashion; good at fixing wrong behaviors and improving targeted capabilities Cons: may be prone to overfitting; implementation is standing in between SFT & Online RL

Drag from top and touch the back button to exit full screen.

Common methods in post-training

Methods	Principles	Pros & Cons
Supervised Fine-tuning (SFT)	Imitate the example responses by maximizing the probability of the response	Pros: simple implementation, great for jump-starting new model behavior Cons: may degrade other performances for tasks not included in training data
Online Reinforcement Learning (e.g. PPO, GRPO)	Maximize the reward for the response	Pros: better at improving model capabilities without degrading performance in unseen tasks Cons: most complex implementation; requires good design of reward functions
Direct Preference Optimization (DPO)	Encourage good answer while discouraging bad answer provided	Pros: train model in a contrastive fashion; good at fixing wrong behaviors and improving targeted capabilities Cons: may be prone to overfitting; implementation is standing in between SFT & Online RL

is standing in between SFT and

Why online RL degrades performance less compared with SFT

W N

DeepLearning.AI

Why Online RL degrades performance less compared with SFT?

```
graph TD; Prompt --> LM[Language Model]; LM --> R1[R1 +1]; LM --> R2[R2 0]; LM --> R3[R3 0]; R1 -- dashed green arrow --> LM;
```

Prompt

Language Model

R1 +1

R2 0

R3 0

Online RL tweaks behaviour within the model's native manifold

Lastly,
Drag from top and touch the back button to exit full screen.

W N

DeepLearning.AI

Why Online RL degrades performance less compared with SFT?

```
graph TD; Prompt --> LM[Language Model]; LM --> R1[R1 +1]; LM --> R2[R2 0]; LM --> R3[R3 0]; R1 -- dashed green arrow --> LM;
```

Prompt

Language Model

R1 +1

R2 0

R3 0

Online RL tweaks behaviour within the model's native manifold

its own answer R1, R2
Drag from top and touch the back button to exit full screen.

W N

DeepLearning.AI

Why Online RL degrades performance less compared with SFT?

```
graph TD; Prompt --> LM[Language Model]; LM --> R1[R1 +1]; LM --> R2[R2 0]; LM --> R3[R3 0]; R1 -- dashed green arrow --> LM;
```

Prompt

Language Model

R1 +1

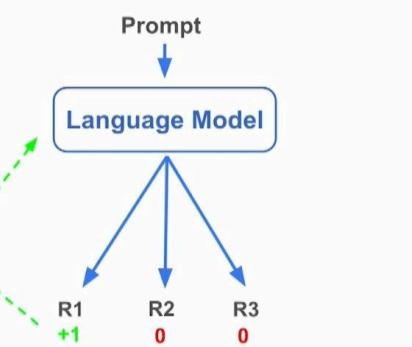
R2 0

R3 0

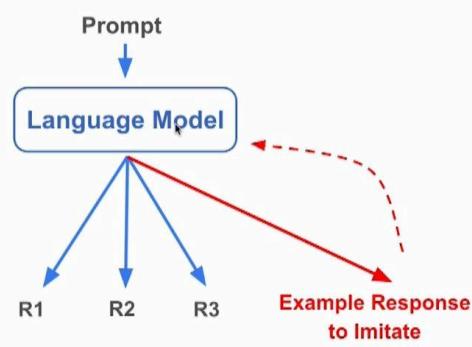
Online RL tweaks behaviour within the model's native manifold

Essentially, online reinforcement

Why Online RL degrades performance less compared with SFT?



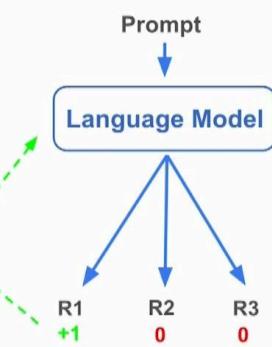
Online RL tweaks behaviour within the model's native manifold
you send the prompt to the language model



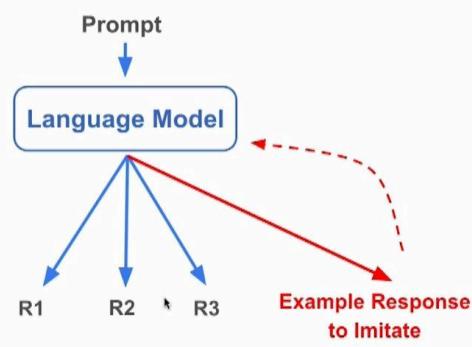
SFT drags it into an alien one, risking unnecessary changes of model weights

Drag from top and touch the back button to exit full screen.

Why Online RL degrades performance less compared with SFT?



Online RL tweaks behaviour within the model's native manifold

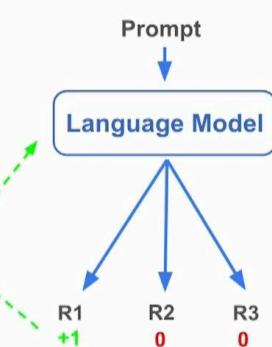


SFT drags it into an alien one, risking unnecessary changes of model weights

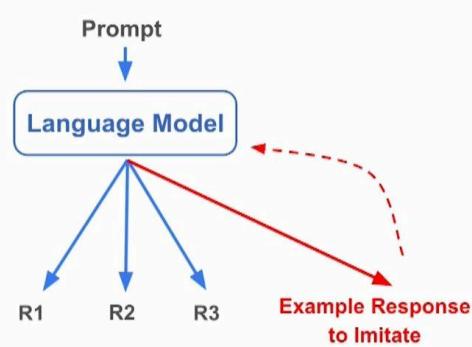
from all the responses

Drag from top and touch the back button to exit full screen.

Why Online RL degrades performance less compared with SFT?



Online RL tweaks behaviour within the model's native manifold



SFT drags it into an alien one, risking unnecessary changes of model weights

This concludes the whole lesson and whole models.

Drag from top and touch the back button to exit full screen.