Tuning Foundation Models CSCI 5922 Spring 2025

Josh Myers-Dean 3/18/2025

Who am I?















Who am I? — Research Experience







2018-2021

2020-2021

2021-Present











2022

2023

2024



Any Gesture





Why Do We Need to Tune Foundation Models?



Tuning — Motivation

Generalist vs. Specialist

BSc in CS Reqs.

- Computer Networking
- Data Structures
- Algorithms
- Software Engineering
- Computer Architecture

BSc emphasizes general knowledge



PhD in CS Reqs. Focus: Machine Learning

- Bayesian Statistics
- NLP
- Reinforcement Learning
- Independent Study
- Dynamical Systems

PhD emphasizes **specific** knowledge



Tuning — Foundation Models

- Large scale foundation model training emphasizes breadth of knowledge
- Fine-tuning allows for specialized use cases

What are the best API's to use to leverage images?

When working with images through APIs, there are several robust options available depending on your needs, such as processing, analyzing, or generating images. Here are some of the top APIs you might consider:

- 1. Google Cloud Vision API: This API offers powerful image recognition capabilities, allowing you to understand the content of an image, including tagging, facial detection, and landmark recognition. It can also perform Optical Character Recognition (OCR).
- 2. Amazon Rekognition: Part of AWS, this API provides similar features to Google Cloud Vision but also includes the ability to analyze video content. It can detect objects, scenes, and faces in images and videos.
- 3. Microsoft Azure Computer Vision API: This service from Microsoft offers features like image tagging, face detection, OCR, and even generates smart thumbnails by analyzing the content of images to produce cropped and framed thumbnails focusing on important elements.
- IBM Watson Visual Recognition: This API provides a suite of image recognition services that include classifying images into thousands of categories, detecting objects and faces, and training custom classifiers on visual content.

Base ChatGPT-4

What are the best API's to use to leverage images?

There are several APIs available to access and analyze remote sensing imagery. The best API depends on your use case, whether it's retrieving satellite images, processing geospatial data, or analyzing land cover and vegetation. Below are some of the top APIs for leveraging remote sensing images:

1. Google Earth Engine (GEE) API

- Best for: Large-scale geospatial analysis, machine learning, and cloud-based processing.
- Features:
 - Free access to a vast archive of satellite imagery (Landsat, Sentinel, MODIS, etc.).
 - Built-in tools for spectral analysis, classification, and time-series analysis.
 - Python and JavaScript support.
- Access: Google Earth Engine

2. Sentinel Hub API

Best for: Efficient access to Sentinel-1, Sentinel-2, Landsat, MODIS, and commercial datasets.

Remote Sensing Fine-tuned GPT



Overview

- Tool use
- Instruction tuning
- RLHF
- Parameter-efficient methods for fine-tuning (PEFT)
- Demo

Overview

- Tool use
- Instruction tuning
- RLHF
- Parameter-efficient methods for fine-tuning (PEFT)
- Demo

+	
Search	
Go	

Core idea: Similar to how we use "tools", let's teach AI to do the same!







Recap of Prompting + ICL

- **Prompting:** Given an instruction, provide a response
- In-context learning: Given k demonstrations, provide a response that matches the structure of the demonstrations

Zero-Shot Prompt

Classify the sentiment of the following text as positive, negative, or neutral. Text: I think the vacation was okay. Sentiment:



Neutral



Few-Shot Prompt

Classify the sentiment of the following text as positive, negative, or neutral. Text: The product is terrible. Sentiment: Negative Text: Super helpful, worth it Sentiment: Positive Text: It doesn't work! Sentiment:



Negative

https://learnprompting.org/docs/basics/few_shot



- Rewind to pre-foundation model era: NS-VQA [Yi, NeurIPS18]
- Key Idea: Use NNs to extract attributes of a scene, generate programs & answer visual questions





Teaching Foundation Models *

 Rewind to pre-foundation model era: NS-VQA [Yi, NeurIPS18]

 Key Idea: Use NNs⁺ extract attributes scene, gener to answer



	(a) Input Ima	app	•						ۍ ر	ne Repres	entation	
ation			A .		C				erial	Color	X	У
Vi			QU.	2	-)		Me	tal	Purple	-0.45	-1.10
II,		11	20-	AV	•		Cube	Me	tal	Green	3.83	-0.04
		A		212		arge	Cube	Me	tal	Green	-3.20	0.63
					+	Small	Cylinder	Rub	ber	Purple	0.75	1.31
		5 -	501	·	5	Large	Cube	Me	tal	Green	1.58	-1.60
			- C1U	ر de-renderin	g)							
	11	1' 1	12			\downarrow]	II. Pr	rogra	m Execut	ion	
	, 51'	1037	rogram. er_shape(scen.	e, cylinder)	1. 2.	filter relate	_shape		3. 4.	filter_ filter_	shape size	5. (
		- / '	. relate(behind)		ID	Size	Shape	•••	ID	Size	•••	A
	-	0	→ 3. filter_shape(scen	e, cube) →	1	Small	Cube		2	Large		An
	111	LSTM -	→ 4. filter_size(scene	, large)	3	Large	Cube		5	Large		
	- 11 -	LSTM -	→ 5. count(scene)		5	Large	Cube					
	\sim											
. 0	20											





- (Almost) modern era: Visual Programming for Compositional Visual Reasoning
- Key idea: Leverage in-context instruction-program pairs to perform tasks using a library of Python functions

```
Instruction: Hide the face of Nicole Kidman with :p
Program:
OBJ0=Facedet(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='Nicole Kidman')
IMAGE0=Emoji(image=IMAGE, object=OBJ1, emoji='face with tongue')
RESULT=IMAGE0
Instruction: Create a color pop of the white Audi
Program:
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='white Audi')
IMAGE0=ColorPop(image=IMAGE, object=OBJ1)
RESULT=IMAGE0
Instruction: Replace the red car with a blue car
Program:
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='red car')
IMAGE0=Replace(image=IMAGE, object=OBJ1, prompt='blue car')
```

RESULT=IMAGE0

Instruction: Replace the BMW with an Audi and cloudy sky with clear sky
Program:



```
IMAGE1=Replace(image=IMAGE0, object=OBJ1, query= cloudy sky )
IMAGE1=Replace(image=IMAGE0, object=OBJ2, prompt='clear sky')
RESULT=IMAGE1
```





- (Almost) modern era: Visual Programming for Compositional Visual Reasoning
- Key idea: Leverage in-context instruction-program pairs to perform tasks using a library of Python functions







Instruction: Replace the ground with white snow and the bear with a white polar bear

Prediction:





- Depth of tasks are only limited by ability to:
- 1) Construct example programs





Program: RESULT=ANSWER2

Natural Language Visual Reasoning LEFT:

Program:

RESULT=ANSWER4 Prediction: False

Factual Knowledge Object Tagging



Program:

OBJ0=FaceDet(image=IMAGE) IMAGE0=Tag(image=IMAGE, object=OBJ1) RESULT=IMAGE0

[Gupta, CVPR23]

Question: Are there both ties and glasses in the picture?

```
BOX0=Loc(image=IMAGE, object='ties')
ANSWER0=Count(box=BOX0)
BOX1=Loc(image=IMAGE, object='glasses')
ANSWER1=Count(box=BOX1)
ANSWER2=Eval("'yes' if {ANSWER0} > 0 and {ANSWER1} > 0 else 'no'")
Prediction: no
```

RIGHT:



Statement: The left and right image contains a total of six people and two boats.

ANSWER0=Vqa(image=LEFT, question='How many people are in the image?') ANSWER1=Vqa(image=RIGHT, question='How many people are in the image?') ANSWER2=Vqa(image=LEFT, question='How many boats are in the image?') ANSWER3=Vaa(image=RIGHT, question='How many boats are in the image?') ANSWER4=Eval('{ANSWER0} + {ANSWER1} == 6 and {ANSWER2} + {ANSWER3} == 2')

Prediction: IMAGE0



Instruction: Tag the 7 main characters on the TV show Big Bang Theory

LIST0=List(query='main characters on the TV show Big Bang Theory', max=7) OBJ1=Classify(image=IMAGE, object=OBJ0, categories=LIST0)

Natural Language Image Editing

IMAGE:







Instruction: Hide Daniel Craig with 8) and Sean Connerv with ;) Program:

OBJ0=FaceDet(image=IMAGE)

OBJ1=Select(image=IMAGE, object=OBJ0, query='Daniel Craig', category=None) IMAGE0=Emoji(image=IMAGE, object=OBJ1, emoji='smiling_face_with_sunglasses') OBJ2=Select(image=IMAGE, object=OBJ0, query='Sean Connery', category: None) IMAGE1=Emoji(image=IMAGE0, object=OBJ2, emoji='winking_face') RESULT=IMAGE1

IMAGE:







Instruction: Replace desert with lush green grass Program:

OBJ0=Seg(image=IMAGE)

OBJ1=Select(image=IMAGE, object=OBJ0, query='desert', category=None) IMAGE0=Replace(image=IMAGE, object=OBJ1, prompt='lush green grass') RESULT=IMAGE0

IMAGE:





Instruction: Create a color pop of Barack Obama (person) Program: OBJ0=Seg(image=IMAGE) OBJ1=Select(image=IMAGE, object=OBJ0, query='Barack Obama', category='person') IMAGE0=ColorPop(image=IMAGE, object=OBJ1)

RESULT=IMAGE0

- Depth of tasks are only limited by ability to:
- 1) Construct example programs
- 2) Modules available in your library



[Gupta, CVPR23]



In-context performance saturates when increasing amount of demonstrations

Near identical performance despite 4x the amount of examples!

Modern day: Agentic Al

I'll search for the highest-rated tour of historic Rome on TripAdvisor. Once I find a suitable option, I'll provide you with the details. Let's begin.

Worked for 2 minutes ^

Navigating to TripAdvisor website Selecting "Things to Do" category Searching for historic Rome tours Closing pop-up, continuing tour search Searching for Historic Rome tours Exploring all historic Rome tour options Closing Colosseum tab, resuming tour search Closing tour pop-up, tab afterward Exploring options for top-rated tours Sorting results by tour ratings Exploring filters for top-rated tours Scrolling for sorting options finding tours



Find and book me the highest rated one-day tour of Rome on Tripadvisor.



OpenAl's Operator (industry)



Modern day: Agentic Al



Simular's Agent S (Academia, ICLR 2025)

Overview

- Tool use \bullet
- Instruction tuning
- RLHF
- Parameter-efficient methods for fine-tuning (PEFT)
- Demo

Instruction Tuning - Familiarity

• If you've used a modern day ChatBot, you've likely interacted with instruction-tuned models!







Instruction Tuning — Intuition



Instruction Tuning

- **Key Idea:** Given a decent pre-trained model, tune it to follow arbitrary instructions!
- ***Also referred to as SFT (Supervised Fine-Tuning)***

Task-specific Modeling

Training on small-scale, well-annotated data

Pre-trained Models

Pre-training on large-scale, noisy data

Task-specific finetuning on small-scale, well-annotated data

Generalist Modeling

Pre-training on xx...x largescale, noisy data

Zero-shot or In-context Fewshot with a few examples as demonstration

Instruction Tuning

Take a pre-trained model / Multimodal alignment pre-training

Instruction tuning on smallscale, pseudo-labeling data

https://www.thetalkingmachines.com/sites/default/files/2023-09/2309.10020_compressed.pdf



Instruction Tuning – Background

Idea: train for many tasks with instructionanswer pairs

Finetune on many tasks ("instruction-tuning")

Input (Commonsense Reasoning)

Here is a goal: Get a cool sleep on summer days.

How would you accomplish this goal?

OPTIONS:

 Keep stack of pillow cases in fridge. -Keep stack of pillow cases in oven.

<u>Target</u>

[Wei, ICLR21]

keep stack of pillow cases in fridge

Input (Translation)

Translate this sentence to Spanish:

The new office building was built in less than three months.

Target

El nuevo edificio de oficinas se construyó en tres meses.

Sentiment analysis tasks

Coreference resolution tasks

. . .



Instruction Tuning – Background

the context and format from its training tasks

Finetune on many tasks ("instruction-tuning")



Intuition: Even if the model hasn't seen the task, it can learn to reason about

Input (Translation)

Translate this sentence to

The new office building was built in less than three

El nuevo edificio de oficinas se construyó en tres meses.

Inference on unseen task type

Input (Natural Language Inference)

Premise: At my age you will probably have learnt one lesson.

Hypothesis: It's not certain how many lessons you'll learn by your thirties.

Does the premise entail the hypothesis?

OPTIONS

-it is not possible to tell -yes

FLAN Response

It is not possible to tell



Instruction Tuning — Background

 In vision: Same setup as NLP, but instructions are now is the context of images

[Liu, NeurIPS23]



<!DOCTYPE html> <html> <head> <title>My Joke Website</title> <style> body { background-color: white; font-family: Arial, sans-serif; } .container { display: flex; flex-direction: column; align-items: center; justify-content: center; height: 100vh; } h1 { font-size: 36px; margin-bottom: 20px; } p { font-size: 20px; } button { background-color: blue; color: white; border: none; padding: 10px 20px; font-size: 18px; cursor: pointer; } </style> </head> <body> <div class="container"> <h1>My Joke Website</h1> Welcome to my joke website. Push the button to reveal the punchline! <button onclick="revealPunchline()">Push me!</button> </div> <script> function revealPunchline() { document.getElementById("punchline").innerHTML = "The punchline is: 'Why was the math book sad? Because it had too many problems."; </script> </body> </html>

My Joke Website [Funny Joke] [push to reveal purchline]

Turn this mock-up into an interactive website using html/js, replace the joke with a real joke.



Instruction Tuning — Vision

• Need: "glue" layer to feed image embeddings to LLM



[Liu, NeurIPS23]



Instruction Tuning - Training

Recall text-generation from RNNs

$$y_i = \operatorname{argmax} \phi(\underbrace{h_i}, \underbrace{h_i})$$
,
hidden state pre-

• This is expensive! Inference time scales linearly with sequence length

 y_{i-1})

evious token

Instruction Tuning - Training



Without Teacher Forcing

Teacher forcing: Feed ground truth y_{i-1} into the model when generating y_i



https://medium.com/towards-data-science/what-is-teacher-forcing-3da6217fed1c



Instruction Tuning - Training

- Teacher forcing: Feed ground truth y_{i-1} into the model when generating y_i
- At inference time: Model learns to "fill in the blank"

User: Translate this sentence: I like dogs Assistant: Me gusta perros

Training time instructions



Inference time instructions

Beyond IT: The Alignment Problem



I have a coworker who always takes credit for my work. I'm feeling anxious about this. What should I do?

Instruction tuning yields direct and concise, task-orientated answers...

What would you respond as a human?





Beyond IT: The Alignment Problem

but does not account for human preferences



I have a coworker who always takes credit for my work. I'm feeling anxious about this. What should I do?

Instruction tuning yields direct and concise, task-orientated answers...



Document your work and keep records of your contributions. Consider talking directly to your coworker about your concerns. If the behavior continues, escalate the issue to your manager or HR.



Beyond IT: The Alignment Problem

but does not account for human preferences



I have a coworker who always takes credit for my work. I'm feeling anxious about this. What should I do?

Instruction tuning yields direct and concise, task-orientated answers...

I'm sorry you're going through this situation. It might help to start by keeping detailed records of your work to clearly establish your contributions. When you're ready, consider having a calm, private conversation with your coworker to share your feelings. If that doesn't help, you could talk to your manager or HR for further support. **Remember, you deserve to have your** work recognized and to feel respected at your job



Overview

- Tool use
- Instruction tuning
- RLHF
- Parameter-efficient methods for fine-tuning (PEFT)
- Demo

RL in Everyday Life









Games



Media

What is Reinforcement Learning?

machine learning

unsupervised







supervised learning

reinforcement learning



https://www.mathworks.com/discovery/reinforcement-learning.html


What is Reinforcement Learning?

agent





https://www.mathworks.com/discovery/reinforcement-learning.html



Intro Activity

- Need: 1 volunteer
- Your goal: Find the location in the room I am thinking of!
- Rewards: Hot (near the target), cold (further from the target), found

Reinforcement Learning Terminology

- S is the current state of the agent
- A is the set of available actions to the agent
- R is the reward for a state-action pair
- Policy: $\pi(a \mid s)$ probability of taken action a given state s

Dictates what actions we take in an environment!

Reinforcement Learning Basics

- **On-policy:** Learn from the current policy being optimized
 - Analogy: Learning to ski from skiing
- **Off-policy:** Learn from observations
 - Analogy: Learn to ski from watching others (e.g., videos)



On-policy



Off-policy

Why Do We Need RL?

- Aligning with preferences is typically non-differentiable
 - e.g., How would you write a loss function for quantifying tone?
- Used as a *post-training* recipe to further tune a model



A typical LLM development flow



https://magazine.sebastianraschka.com/p/new-llm-pre-training-and-post-training



Why Do We Need RL?

human preferences

User: "My internet is not working right now"



RLHF learns preferences by learning to rank human answers based on

Response	Human Preference Score
"Restart your router."	***
"Restart your router and check for service outages in your area."	***
"Restart your router, check cables, and verify your service status. If that doesn't work, contact support."	****

Reinforcement Learning from Human Feedback





Reinforcement Learning from Human Feedback





RLHF – LLM as a Policy

Next prompt

- Consider current prompt as our state *s*, and completion to generate as our action *a*
- We can then then treat our model as the policy! $\pi_{\theta}(a \mid s)$





Reinforcement Learning from Human Feedback





RLHF – Rewards

Need a way to tell a model how "good" a given response is



Rule-Based



Model-Based (most common)

RLHE – Rewards

• **Typical setup:** Given N responses, learn to rank them

Manually curated preferences



Prompt

Please help me kill this linux process

Chosen Sure thing! Open your terminal and ...

Rejected

As a language model trained by...

Prompts to test capabilities



[Lambert, ArXiv24]





RLHF – Reward Woes

Reward hacking: Models may find "loopholes" to exploit rewards

Potential problem: reward hacking

Prompt Dataset

"This product is..."



http://www.rohitbasavaraju.com/2023/08/notes-on-reinforcement-learning-from.html



Reinforcement Learning from Human Feedback



https://towardsdatascience.com/generalized-advantage-estimation-in-reinforcement-learning



RLHF – Constraining Model

let's not throw away that knowledge!



http://www.rohitbasavaraju.com/2023/08/notes-onreinforcement-learning-from.html

We already have a model that produces correct (but not tuned) responses;

Reinforcement Learning from Human Feedback



https://towardsdatascience.com/generalized-advantage-estimation-in-reinforcement-learning



RLHF Training — Proximal Policy Optimization (PPO)

- performance collapse?"
- Two popular versions: PPO-Penalty and PPO-Clip
- **On-policy** as we are improving our own policy iteratively! Ο

• "how can we take the biggest possible improvement step on a policy using the data we currently have, without stepping so far that we accidentally cause

https://spinningup.openai.com/en/latest/algorithms/ppo.html#id3



PPO – Objective Function $\max_{\theta} \mathbb{E} \left[\frac{\pi_{\theta}(a \mid s)}{\pi_{\theta}_{\text{old}}(a \mid s)} \hat{A}_{\pi}(s, a) - \beta D_{KL}[\pi_{\theta}_{\text{old}}(\cdot \mid s), \pi_{\theta}(\cdot \mid s)] \right]$

• Kind of scary! Let's break it down

PPO – Objective Function

max E θ

Find the parameters that maximize the expected rewards

Probability ratio: Controls how much we update the policy

Advantage function: tells us if the actions were better than expected. If positive, increase $\pi_{\theta}(a \mid s)$, else decrease

 $\begin{array}{c} \pi_{\theta}(a \mid s) \\ \pi_{\theta}_{\text{old}}(a \mid s) \end{array} \hat{A}_{\pi}(s, a) - \beta C_{KL}[\pi_{\theta}_{\text{old}}(\cdot \mid s), \pi_{\theta}(\cdot \mid s)] \end{array}$

Knob controlling how much we penalize deviations from original policy

How much we deviate from original policy

Challenges in RLHF

Sometimes RLHF is just putting a mask on a nasty mess



https://www.latent.space/p/rlhf-201



Challenges in RLHF

supervised learning

Sample efficiency: RL algorithms typically require more observations than



Samples

Challenges in RLHF

Exploration vs. exploitation: How do we balance trying new actions (exploration) while exploiting what we know works?



Want to Learn More About RL at CU?

- Decision Making Under Uncertainty ASEN 5264
 - Zachary Sunberg <u>https://github.com/CU-ADCL/CU-DMU-Materials</u>
 - Typically offered Spring
- Deep Reinforcement Learning CSCI 7000
 - Alessandro Roncone
 - Typically offered Fall

Overview

- Tool Use
- Instruction Tuning
- RLHF
- Parameter-Efficient Methods for Fine-Tuning (PEFT)
- Demo

How Can Academia Keep Up?

- How can we take advantage of these models?





Most large models come from industry labs with large amounts of resources

- Yes... If we don't care about retaining previous knowledge
- No... If the models are large or you are GPU poor (e.g., grad students)

- Yes... If we don't care about retaining previous knowledge
- Catastrophic forgetting: A model "forgets" old knowledge as new knowledge is learned







- No... If the models are large or you are GPU poor (e.g., grad students)
- What gets stored in memory when training models?

- Need to store model weights, gradients, optimizer states, etc.
- Consider model with 1B parameters, AdamW optimizer, 32-bit precision
- How much GPU VRAM do you think we need?

(a) 12GB (b) 48GB

(c) 200GB (d) 100GB

- 4 bytes + 4 bytes + 12 bytes $\cdot 1e^9 \approx 2B$ bytes $\approx 200GB$ Model Optimizer Gradients
- Also need memory for images/text/audio!

GPU	Memory	Cost (2/2024)	(Cloud) Machines
T40 / K80	24GB	\$150	Google Colab, AWS p2.*
V100	32GB	\$2,500	Google Colab
A100	40GB or 80GB	\$8,000/\$16,000	Google Colab, AWS p3.*
H100	80GB	\$44,000	AWS p4.*
6000 Ada, L40	48GB	\$8000	N/A
Mac M*	Same as CPU	\$2000	N/A

https://phontron.com/class/anlp2024/assets/slides/anlp-08-instructiontuning.pdf



PEFI – Motivation

- For N specific tasks, need to fine-tune N different models
- What if we just train a few specialized parameters?

PEFT — Motivation

- PEFT Parameter Efficient Fine-Tuning
- performance
- 0

 Adapt pre-trained models to new tasks by updating only a small subset of parameters to reduce computational and storage costs while maintaining

Less parameters == less memory == less GPUS == happy grad students!

PEFT – Benefits

• Prevents catastrophic forgetting! Want to perform your original task? Drop the modified parameters for the original ones!





PEFT — Motivation

Cheaper for on device O applications as we do not need to store N copies of the model!

Serializ
250MB —
200MB —
150MB
100MB
50MB
0MB



PEFT — Motivation



• Analogy: Hands serve as our "base" model and tools as adaptations for new tasks


PEFT - Overview



(a) Adapter

(b) Prefix Tuning

Add Add Add Scaling Scaling 00 00 $h \bigcirc$ $h \bigcirc$ $\bigcirc \bigcirc$ \bigcirc \bigcirc \bigcirc $W_{\rm up}$ $W_{ m up}$ $W_{ m up}$ ReLU ReLU PLM module PLM module PLM module $W_{ m down}$ $W_{ m down}$ $W_{ m down}$ $\bigcirc \bigcirc$ \bigcirc $x \bigcirc$ \bigcirc $x \bigcirc$ \bigcirc \bigcirc \bigcirc

(c) LoRA

(d) Parallel Adapter

(e) Scaled PA



[He, ICLR22]

PEFT - Overview



Add Add Add Scaling Scaling 00 00 $h \bigcirc$ $h \bigcirc$ 00 \bigcirc \bigcirc \bigcirc $W_{ m up}$ $W_{\rm up}$ $W_{ m up}$ ReLU ReLU PLM module PLM module PLM module $W_{ m down}$ $W_{ m down}$ $W_{ m down}$ 00 00 \bigcirc $x \bigcirc$ \bigcirc $x \bigcirc$ \bigcirc \bigcirc

(c) LoRA

(d) Parallel Adapter

(e) Scaled PA



[He, ICLR22]

PEFT — Adapters

- transformer blocks
- Freeze gradient updates to the original model parameters



Adapters introduce task-specific trainable parameters (3-5% additional) into





PEFT – Adapters

- Consider a generalist model (e.g., BERT) trained on English
- What if I want to perform NER on a low-resource language (e.g., Quechua)?

Apple or today one announced the second QUANTITY generation iPhone SE COMM a powerful new iPhone COMM featuring a 4.7- inch QUANTITY Retina HD display.



PEFT — Adapters

- translation)
- Let's combine them!



Each adapter corresponds to some task (e.g., language translation, task

Source adapters (solid) get replaced by target adapters (Dashed) at inference time





PEFT — Adapters

- Downsides?
- Requires extra parameters (3-5%) extra inference time
- Need separate modules for each task no multi-task learning
- Need additional storage for each module

PEFT - Overview



Add Add Add Scaling Scaling 00 00 $h \bigcirc$ $h \bigcirc$ 00 \bigcirc \bigcirc \bigcirc $W_{ m up}$ W_{up} $W_{ m up}$ ReLU ReLU PLM module PLM module PLM module $W_{ m down}$ $W_{\rm down}$ $W_{ m down}$ 00 $x \bigcirc$ OO $x \bigcirc$ OO \bigcirc \bigcirc \bigcirc

(c) LoRA

(d) Parallel Adapter

(e) Scaled PA



[He, ICLR22]

PEFT – Prefix Tuning

- Introducing Soft Prompts: learnable, task specific vectors
- Instead of giving the model a text prompt, prepend a task specific vector to the model inputs and freeze the rest of the model!





PEFT – Prefix Tuning

- matrix P_{θ})
- Only need to update P_{θ} during training!



• Key Idea: First 2 hidden states (h_i) are the learnable vectors (i.e., rows from a

ve Model (e.g. GPT2)								
y (target utterance)								
								l
gwarts	[SEP]	Harry	Potter	is	graduated	from	Hogwarts	
h_8	h_9	h_{10}	h_{11}	h_{12}	h_{13}	h_{14}	h_{15}	
8	9	10	11	12	13	14	15	
8]		Y_{idx}	= [9,	10, 1	1, 12, 13	, 14, 1	15]	



PEFT – Prefix Tuning: CoOp

- Prefix tuning works for vision too!
- Consider CLIP; How well would this do on satellite imagery? Why?



Prompt

- a photo of a [CLASS].
- a satellite photo of [CLASS].

a centered satellite photo of [CLASS].





PEFT – Prefix Tuning: CoOp

Domain shift from original training data! Prompt engineering only gets so far!



Prompt

- a photo of a [CLASS].
- a satellite photo of [CLASS].
- a centered satellite photo of [CLASS].





PEFT – Prefix Tuning: CoOp

- Context Optimization optimize for the context the image appears in
- Key Idea: Learn the context of the image with continuous vectors
- Replace prompts with soft prompts ("a photo of..." \rightarrow [V])

EuroSAT



a photo of a

- a satellite ph
- a centered sa

[V]₁[V]₂ ... [V

Prompt	Accuracy
[CLASS].	22.30
oto of [CLASS].	31.12
atellite photo of [CLASS].	31.53
] _м [CLASS].	81.60



PEFT – Prefix Tuning

- Downsides of prefix tuning
 - Vectors are **task specific** need to train a new one for each task



Prompts are only affected, what if we need to modify the vision features?

PEFT - Overview



[He, ICLR22]

PEFT -- LORA

- Let's learn Low Rank Adaptations of our model weights!
- Key Idea: For specific linear layers (e.g., attention weights), learn 2 matrices of rank r during training; freeze all other parameters



Low-rank Matrix Decomposition

https://www.aporia.com/learn/low-rank-adaptation-lora/





PEFT – LORA

• At inference: merge newly learned weights with OG weights!



[Hu, ICLR22]



https://huggingface.co/docs/peft/main/en/conceptual_guides/lora



PEFT – LORA

Faster at inference time since no new parameters

Batch Size Sequence Length $ \Theta $	32 512 0.5M	16 256 11M	1 128 11M
Fine-Tune/LoRA	1449.4±0.8	338.0±0.6	19.8±2.7
Adapter ^L Adapter ^H	1482.0±1.0 (+2.2%) 1492.2±1.0 (+3.0%)	354.8±0.5 (+5.0%) 366.3±0.5 (+8.4%)	23.9±2.1 (+20.7%) 25.8±2.2 (+30.3%)



Inference latency



PEFT – LORA

Better performance on target tasks!





	WikiSC)L			
×					•
	•••				
				Method	
*			•	Fine-Tu	ne
			+	PrefixE	mbed
			*	PrefixLa	ayer
			×	Adapte	r(H)
				LoRA	

11 8 10 9 *log*₁₀ # Trainable Parameters



- Most PEFT methods have only slightly different designs! 0
- How do we know which one to use?





- No additional latency?
- **Classification task?** 0
- Complex task + low computational budget? •
- Complex task + high computational budget?



- LoRA
- Prefix Tuning
- Adapters 0



- No additional latency?
 - LoRA
- **Classification task?**
- Complex task + low computational budget?
- Complex task + high computational budget?



PEFT -- Wrap Up

- No additional latency?
 - LoRA
- Classification task?
 - Likely doesn't matter
- Complex task + low computational budget?
- Complex task + high computational budget?



- No additional latency?
 - LoRA
- **Classification task?**
 - Likely doesn't matter
- Complex task + low computational budget?
 - Prefix tuning
- Complex task + high computational budget?



- No additional latency?
 - LoRA
- **Classification task?**
 - Likely doesn't matter
- Complex task + low computational budget?
 - Prefix tuning
- Complex task + high computational budget?
 - Adapters



- Anecdote: LoRA is the de facto standard for tuning large models

Adapters are also popular since extra parameters give more expressive power



Overview

- Tool use
- Instruction tuning
- RLHF
- Parameter-efficient methods for fine-tuning (PEFT)
- Demo