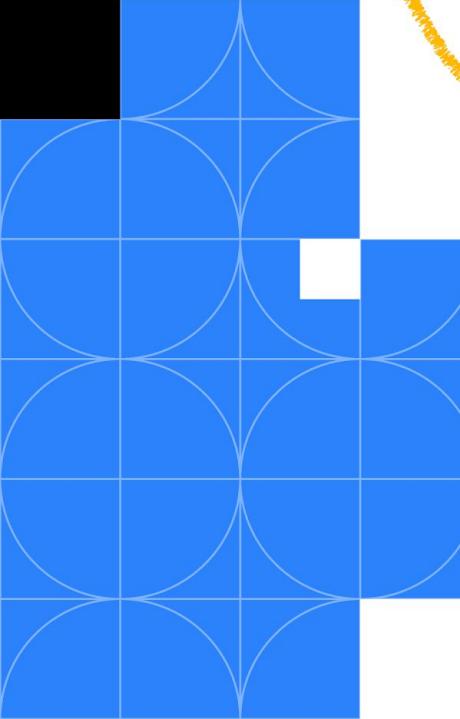


```
text:  
  'Section Title',  
  style: TextStyle(  
    color: Colors.blue[200],  
  ),  
,
```

devfest

```
s.star,  
r: Colors.blue[500],  
Text('23'),
```



Google Developer Groups



Optimizing LLMs



A Journey through Parameter-Efficient Fine-Tuning with Low-Rank Adaptation



@aadityaura



aadityaura@gmail.com



aadityaura.github.io

Ankit Pal (Aaditya Ura)
Senior Research Engineer,
Saama AI Research Lab, Chennai

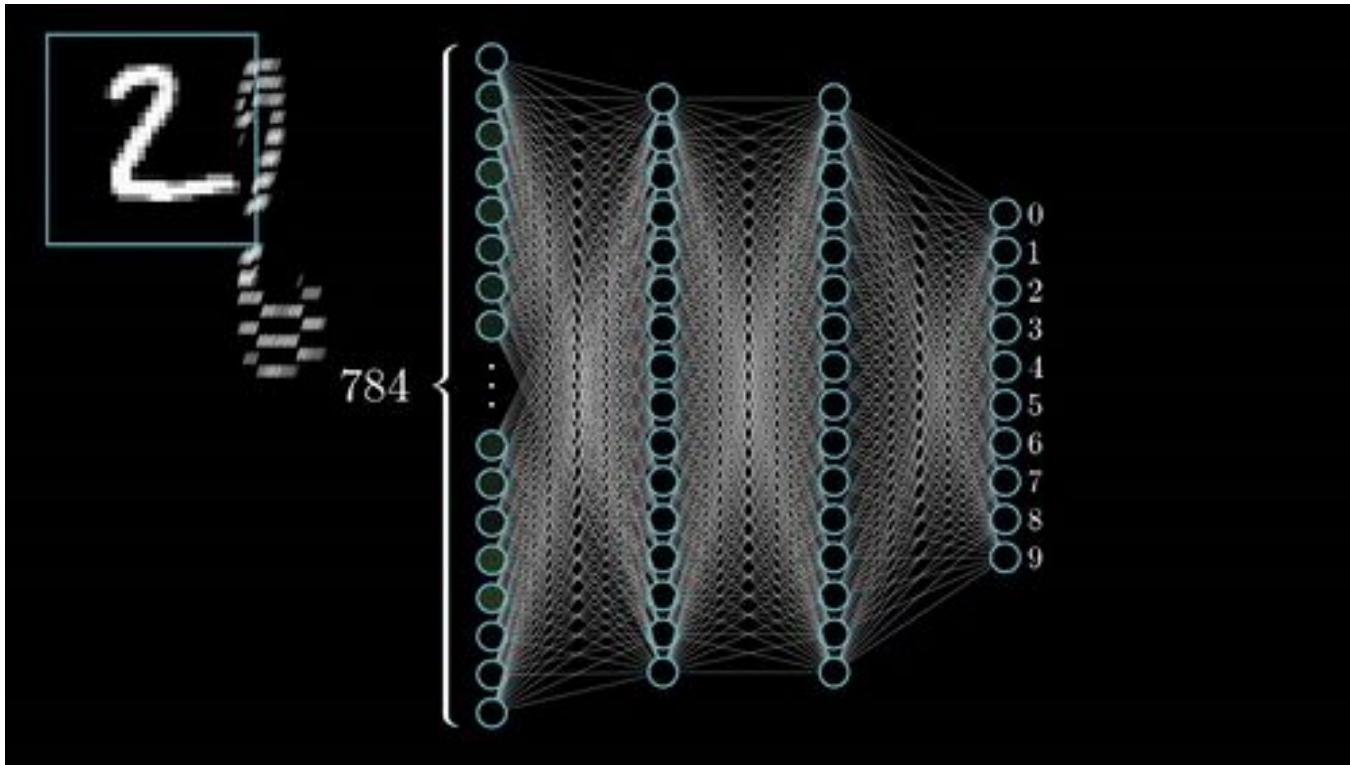
About Me



Cool Research | Trekking | Boxing | Skiing | Chess

- Senior AI Research Engineer [@ Saama AI Research Lab](#)
- Research interests involve
Representation Learning on Graphs and Manifolds
- Generative Modeling, MLOps, Signal Processing
and their applications in Healthcare data

AI/ML



Source : <http://tinyurl.com/ypa76y9m>

Neural Networks

classify the each number's image from 0-9

1	1	5	4	3
7	5	3	5	3
5	5	9	0	6
3	5	2	0	0

Neural Networks



?

Which number is it?

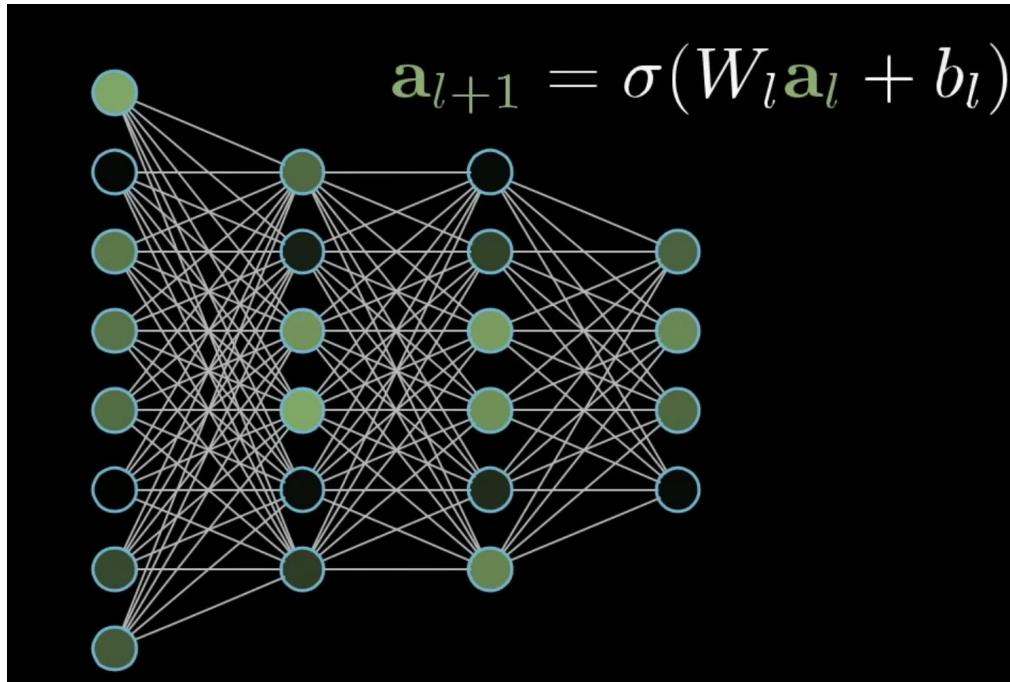
Neural Networks



Default view (

Source : <http://tinyurl.com/ypa76y9m>

Neural Networks



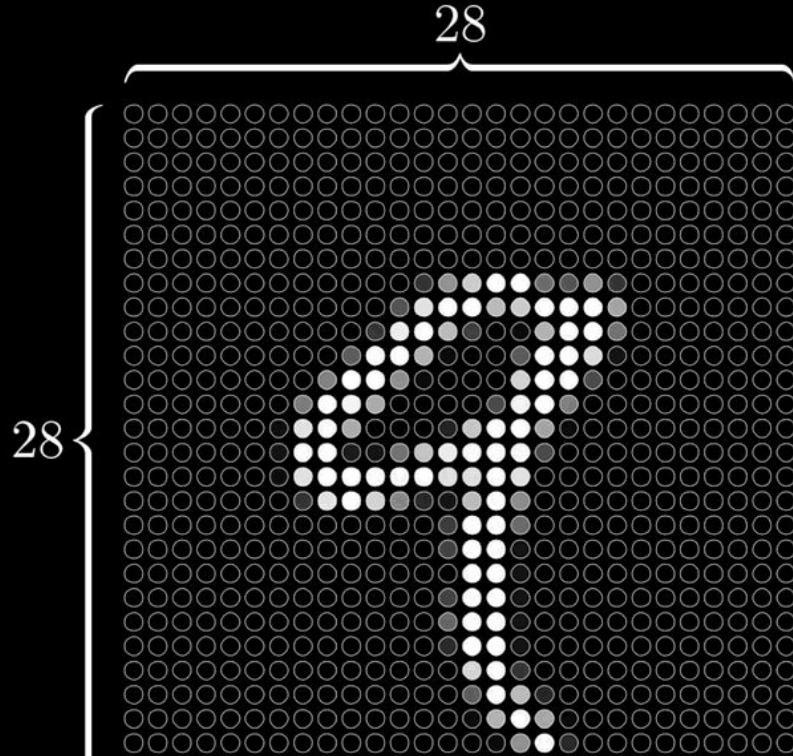
Neural Networks



Neuron

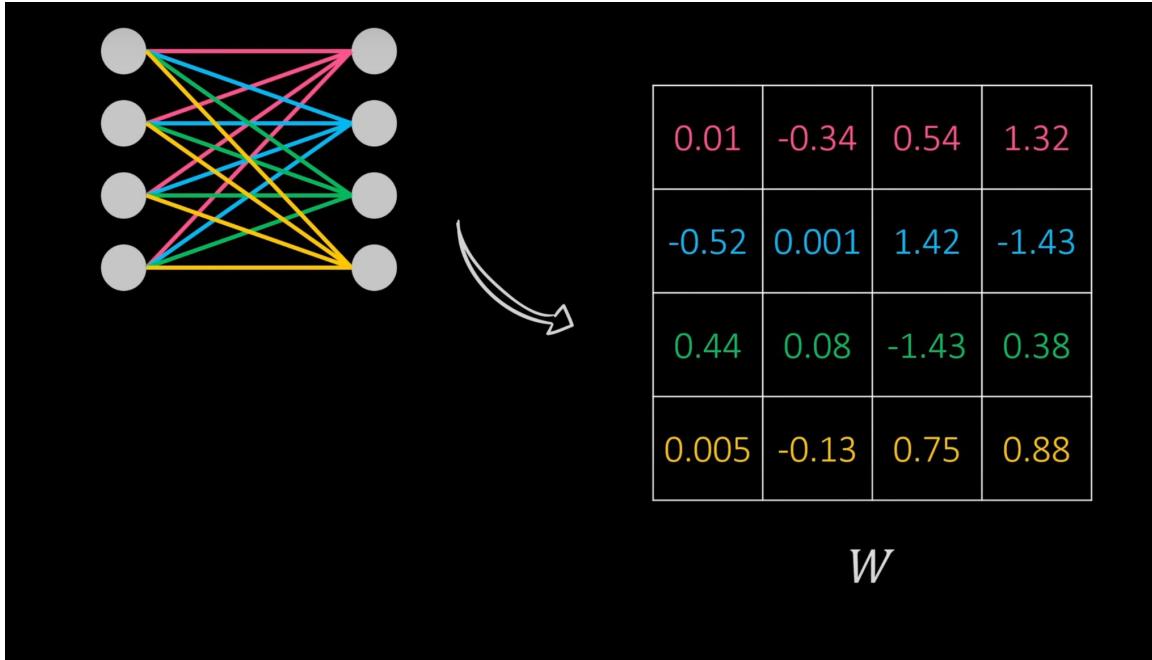
Source : <http://tinyurl.com/ypa76y9m>

Neural Networks



$$28 \times 28 = 784$$

Parameters in Machine Learning Model

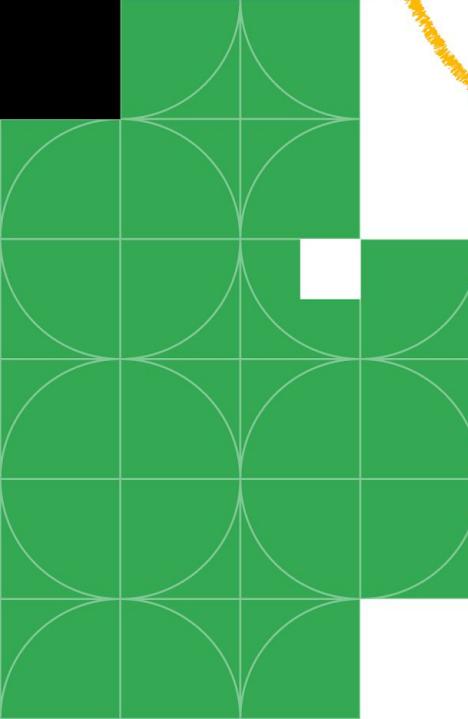


```
text  
  'Section Title',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
,
```

devfest

```
s.star,  
r: Colors.green[500],
```

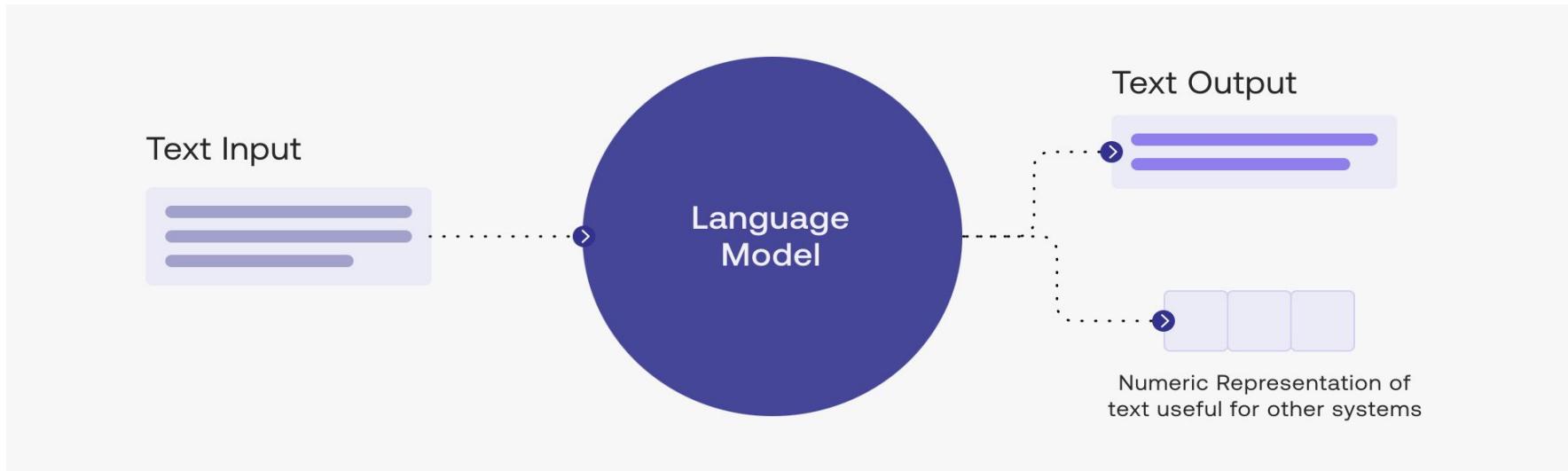
```
Text('23'),
```



Large Language Models(LLMs)

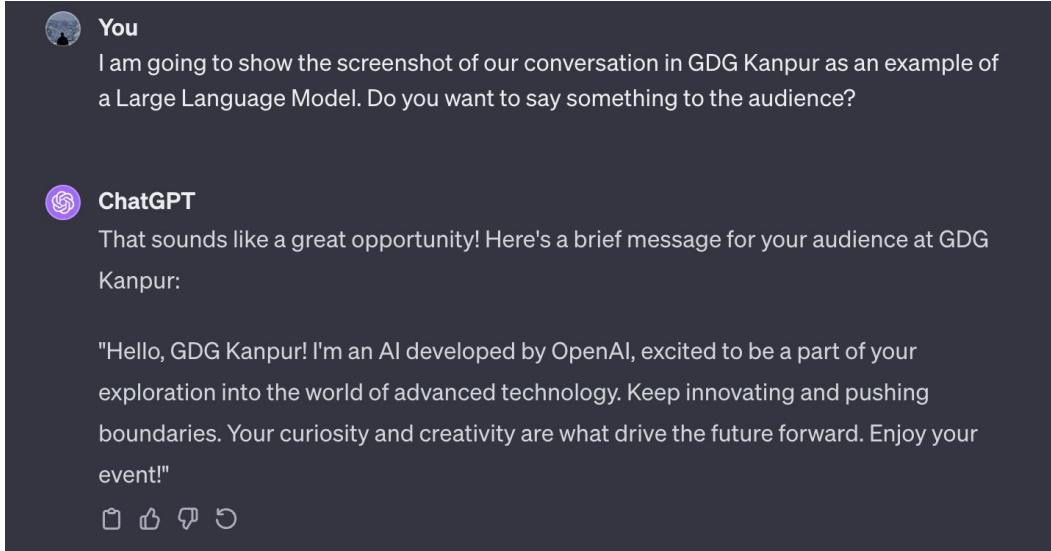
Understanding Large Language Models

LLMs are advanced AI models trained on vast datasets to understand and generate human-like text.



Source: <https://docs.cohere.com/docs/llmu>

Understanding Large Language Models

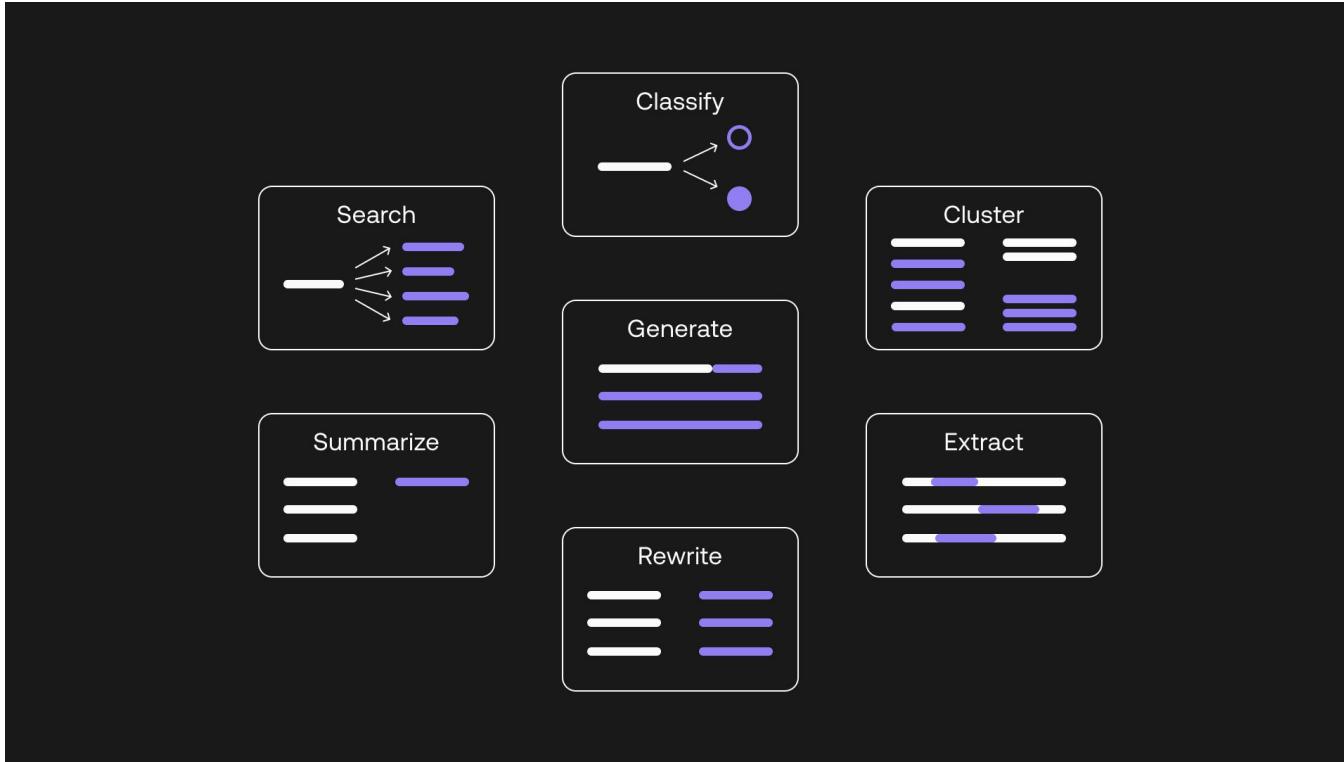


You
I am going to show the screenshot of our conversation in GDG Kanpur as an example of a Large Language Model. Do you want to say something to the audience?

ChatGPT
That sounds like a great opportunity! Here's a brief message for your audience at GDG Kanpur:
"Hello, GDG Kanpur! I'm an AI developed by OpenAI, excited to be a part of your exploration into the world of advanced technology. Keep innovating and pushing boundaries. Your curiosity and creativity are what drive the future forward. Enjoy your event!"
   

ChatGPT is one example of a Large Language Model.

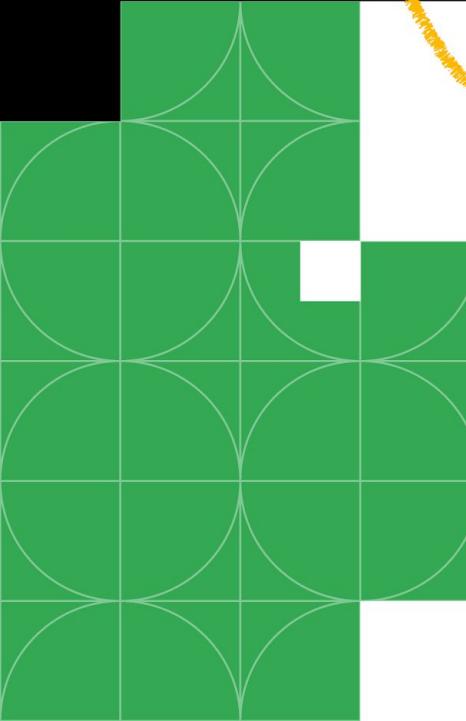
Understanding Large Language Models



```
text  
  'Section Title',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
,
```

devfest

```
s.star,  
r: Colors.green[500],  
  
Text('23'),
```



Google Developer Groups

What is Fine-Tuning?

Fine-Tuning

Taking a pre-trained model and training at least one model parameter

Fine-Tuning

Taking a pre-trained model and training at least one model parameter



GPT-3

Fine-Tuning

Taking a pre-trained model and training at least one model parameter



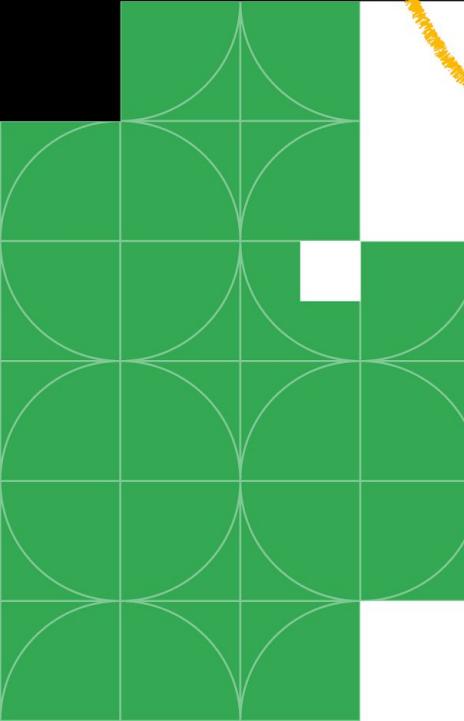
GPT-3

ChatGPT
(i.e. GPT-3.5-turbo)

```
text  
  'Section Title',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
,
```

devfest

```
s.star,  
r: Colors.green[500],  
  
Text('23'),
```



Google Developer Groups

Why We Need to Fine-Tune?

Need of Fine-Tuning



Say drink

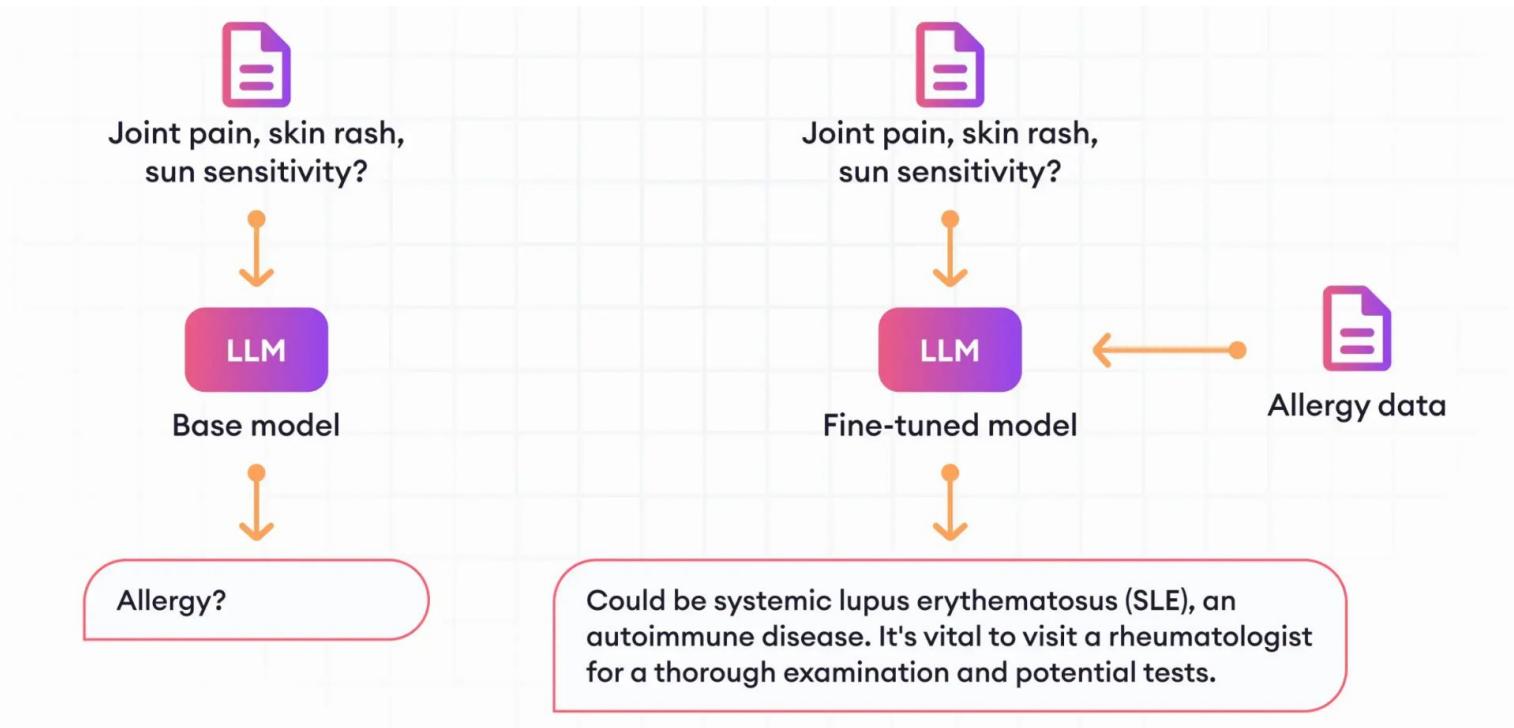
Drink.

Say it 10 times

Sure! Here you go:

1. It
2. It
3. It
4. It
5. It
6. It
7. It

Need of Fine-Tuning

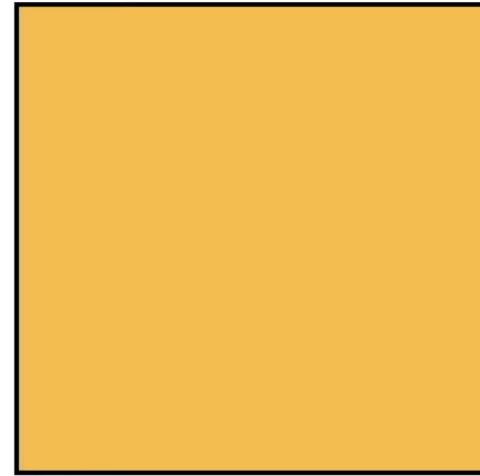


Need of Fine-Tuning

A smaller (fine-tuned) model can outperform a larger base model



InstructGPT (1.3B)

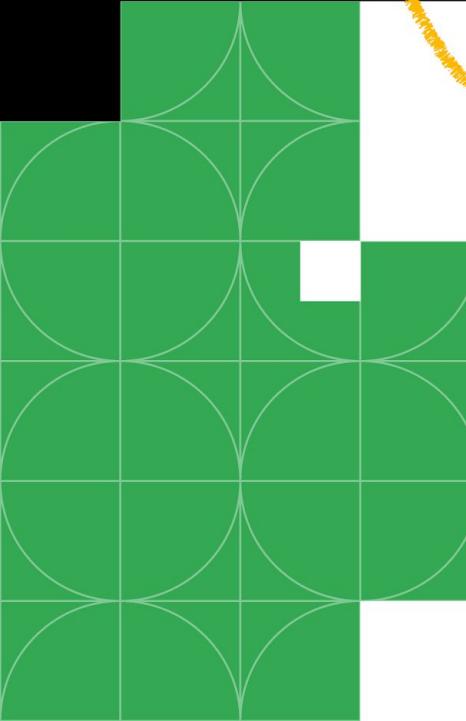


GPT-3 (175B)

```
text  
  'Section Title',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
,
```

devfest

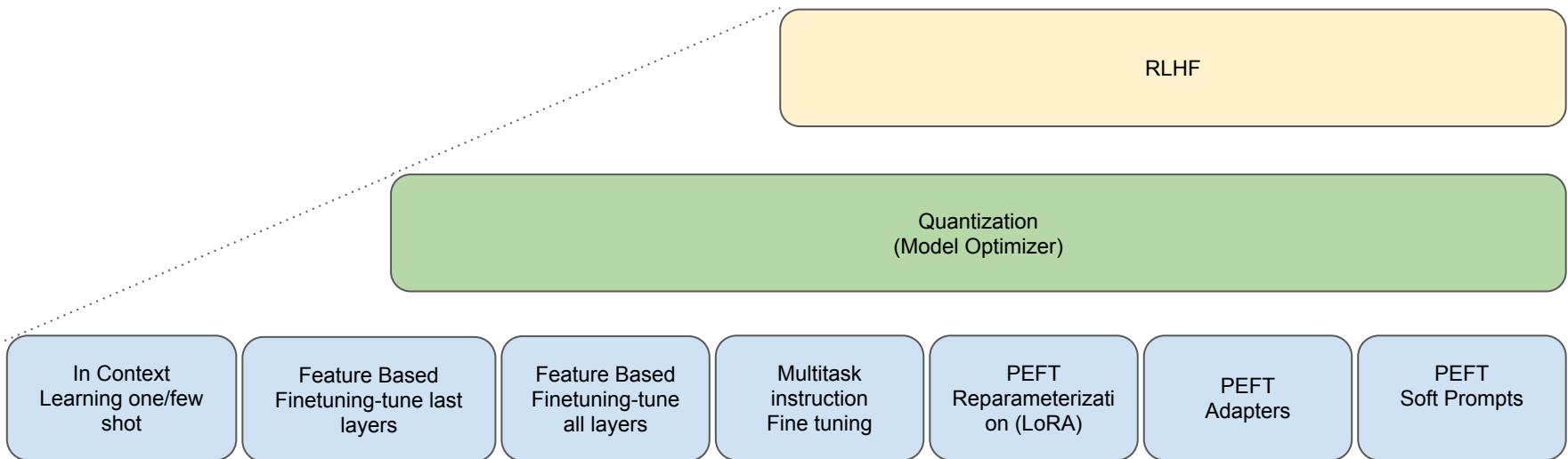
```
s.star,  
r: Colors.green[500],  
  
Text('23'),
```



Google Developer Groups

Fine-Tuning Approaches

Fine-Tuning Approaches

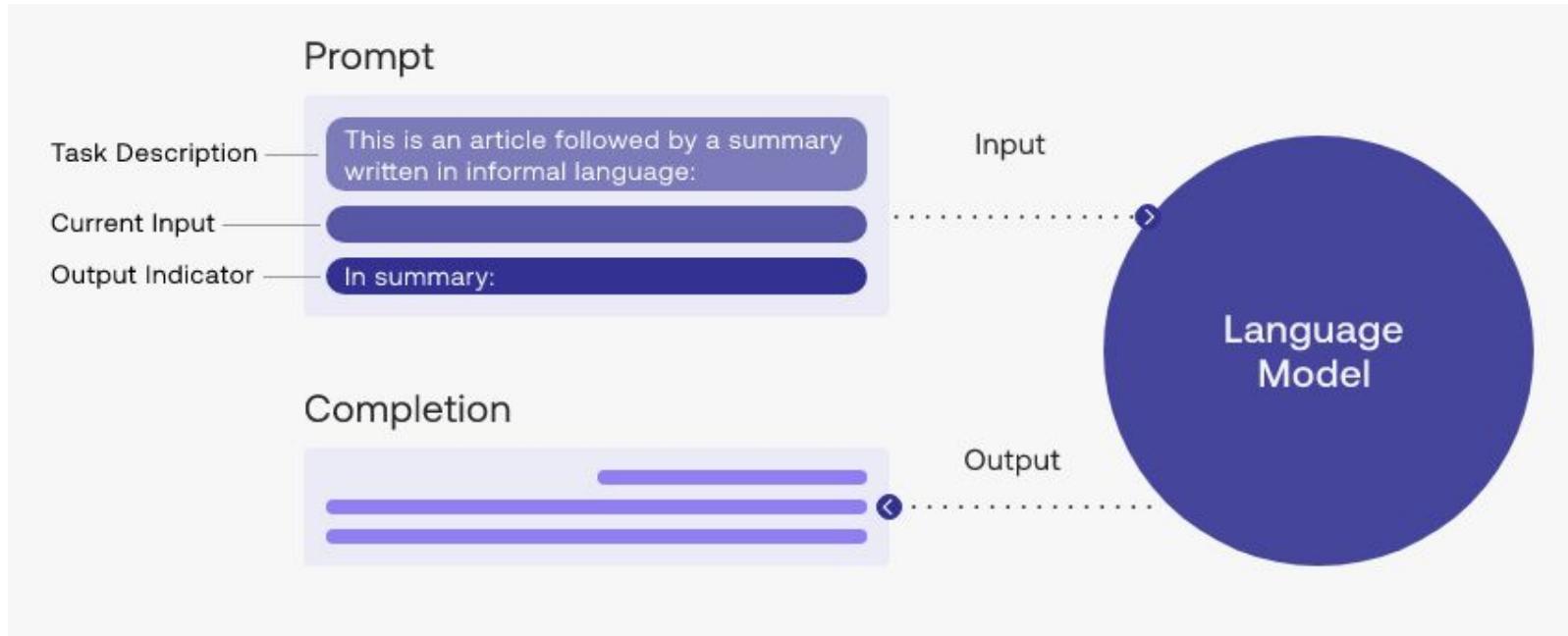


Fine-Tuning Approaches

Prompt
Engineering



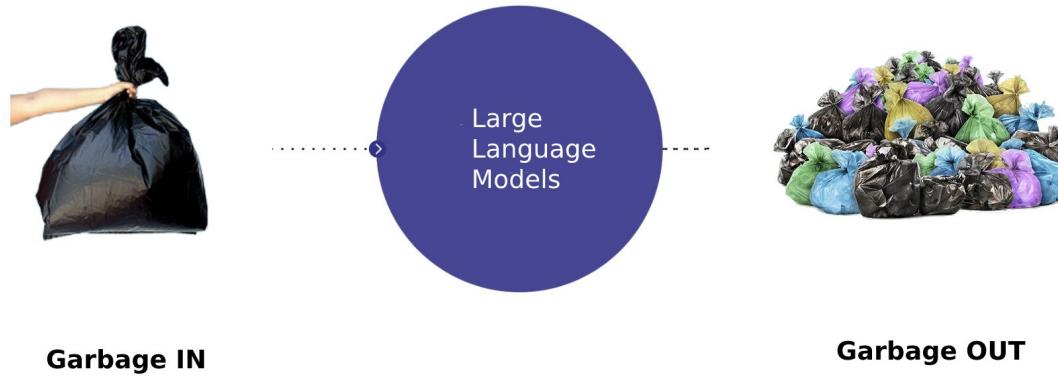
Prompt-Engineering



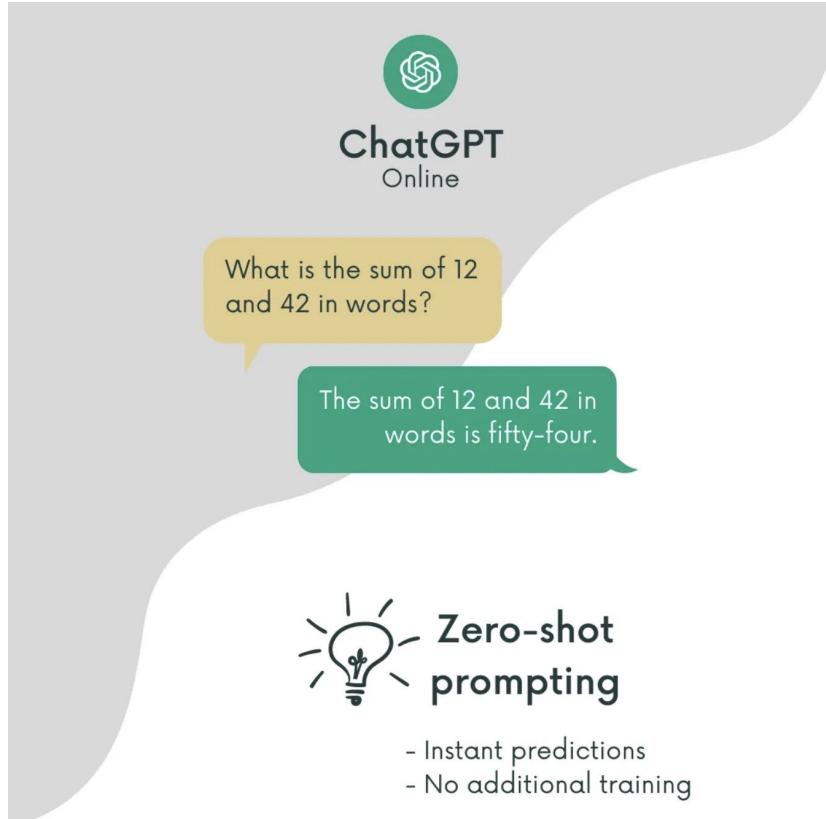
Prompting is a method that allows users to interact with LLMs

Prompt-Engineering

Prompt plays a key role in LLMs



Types of prompting



Types of prompting

The diagram illustrates the concept of few-shot prompting. It features a grey background with a large, light-grey speech bubble shape. Inside this bubble, the ChatGPT logo (a green circle with a white swirl) and the text "ChatGPT Online" are displayed. Below this, a yellow speech bubble contains the first two interactions:

Q: What is the sum of 2 and 3?
A: Five

Q: What is the sum of 12 and 10000?
A: Ten thousand and twelve

Q: What is the sum of 12 and 42?

Below these, a green speech bubble contains the response:

The sum of 12 and 42 is:
 $12 + 42 = 54$
So, the answer is fifty-four.

 **Few-shot prompting**

- Examples or templates needed
- One to five examples

Types of prompting

The image shows a digital interface for ChatGPT Online. At the top is a green circular icon with a white AI-like symbol. Below it, the text "ChatGPT" is written in a large, bold, black font, with "Online" in a smaller, regular black font underneath. A grey speech bubble contains a question: "Q: What is the sum of 14 and 18?". The response is: "A: To sum 14 and 18, add 8 and 4 to give 12, carry over 1. Add the carried over 1 to 1 and 1. This sums to 31." Below this, another question is asked: "Q: What is the sum of 32 and 49?". The response is: "To sum 32 and 49, start by adding the ones place, which gives $2 + 9 = 11$. Write down the 1 and carry over the 1. Then add the tens place, which gives $3 + 4 +$ the carried over 1, for a total of 8." Finally, the result is given as "Therefore: $32 + 49 = 81$ ".

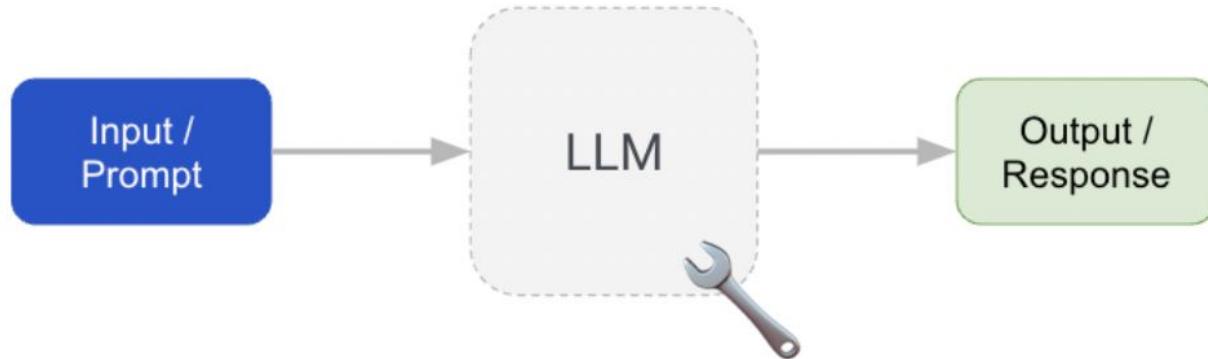
Chain-of-thought prompting



- Breaks down problems
- More interpretable answers

Fine-Tuning Approaches

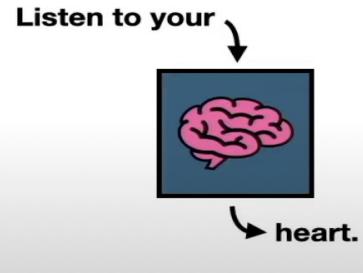
Fine-tuning



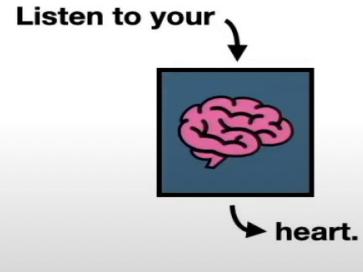
Fine-Tuning Methods



Types of Fine-Tuning

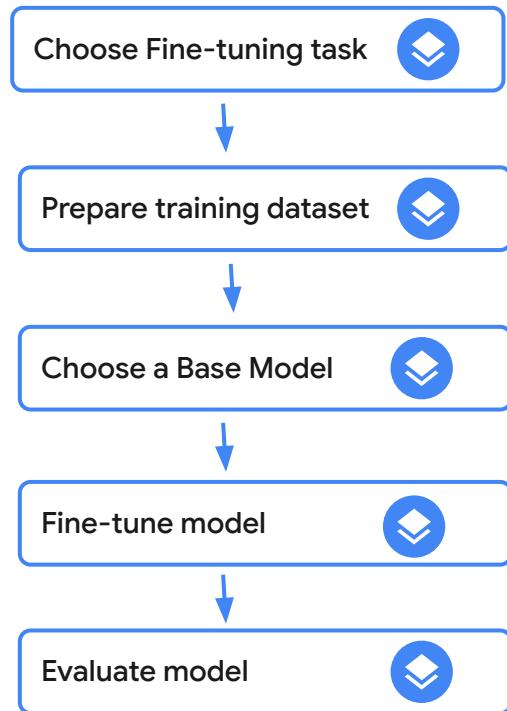


Types of Fine-Tuning



The cat

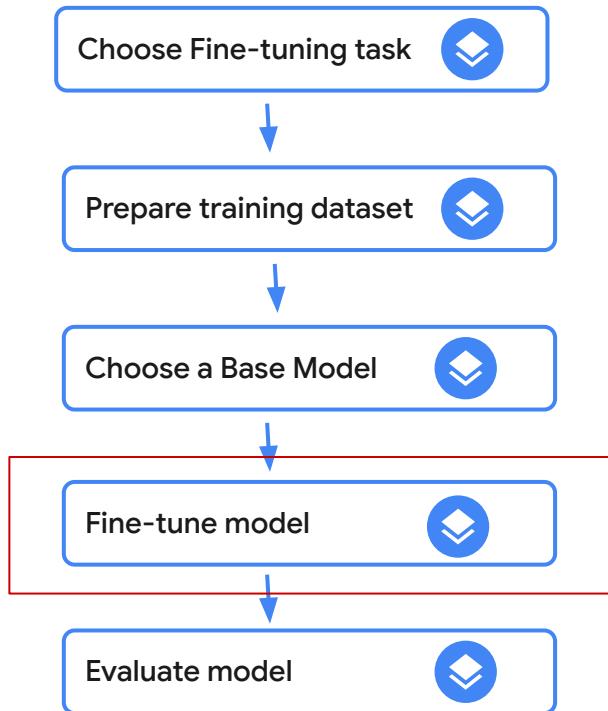
5 Steps



Input	Output



5 Steps

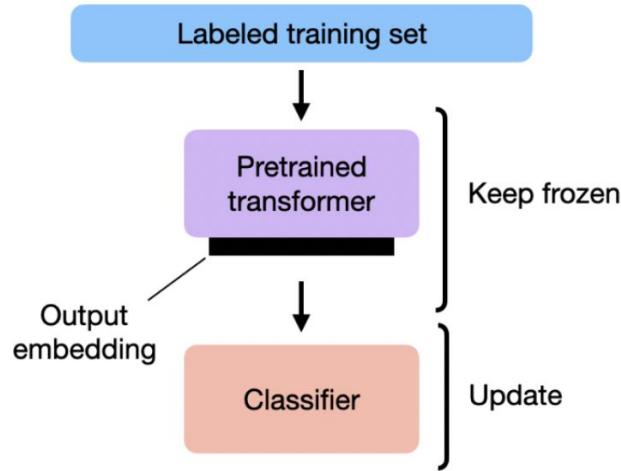


Input	Output



Fine-tune model via supervised learning

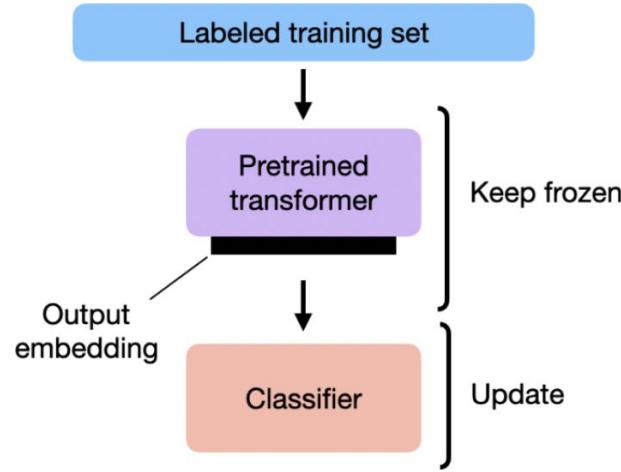
1) FEATURE-BASED APPROACH



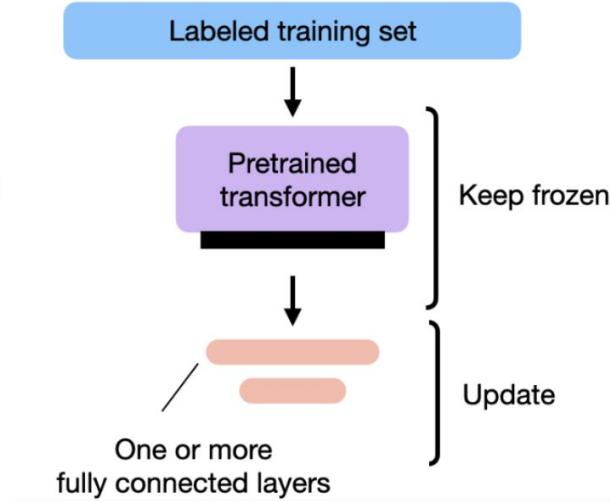
<https://magazine.sebastianraschka.com/p/finetuning-large-language-models>

Fine-tune model via supervised learning

1) FEATURE-BASED APPROACH



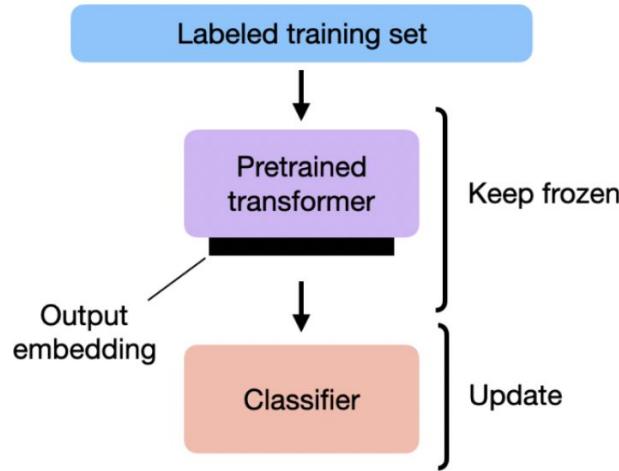
2) FINETUNING I



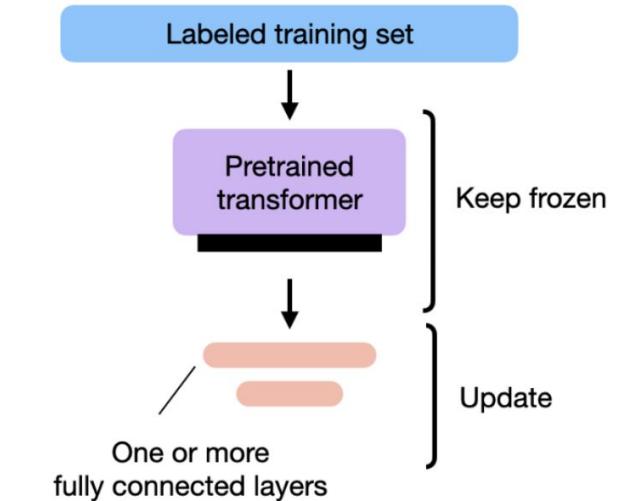
<https://magazine.sebastianraschka.com/p/finetuning-large-language-models>

Fine-tune model via supervised learning

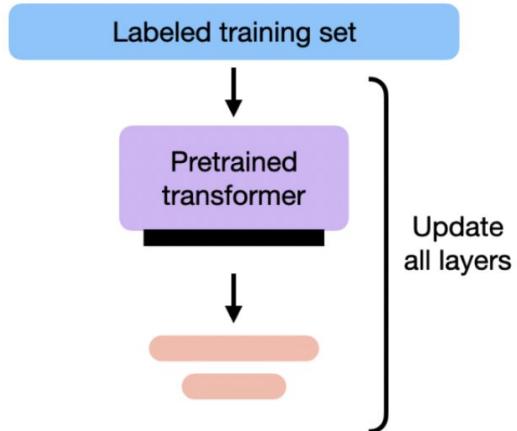
1) FEATURE-BASED APPROACH



2) FINETUNING I

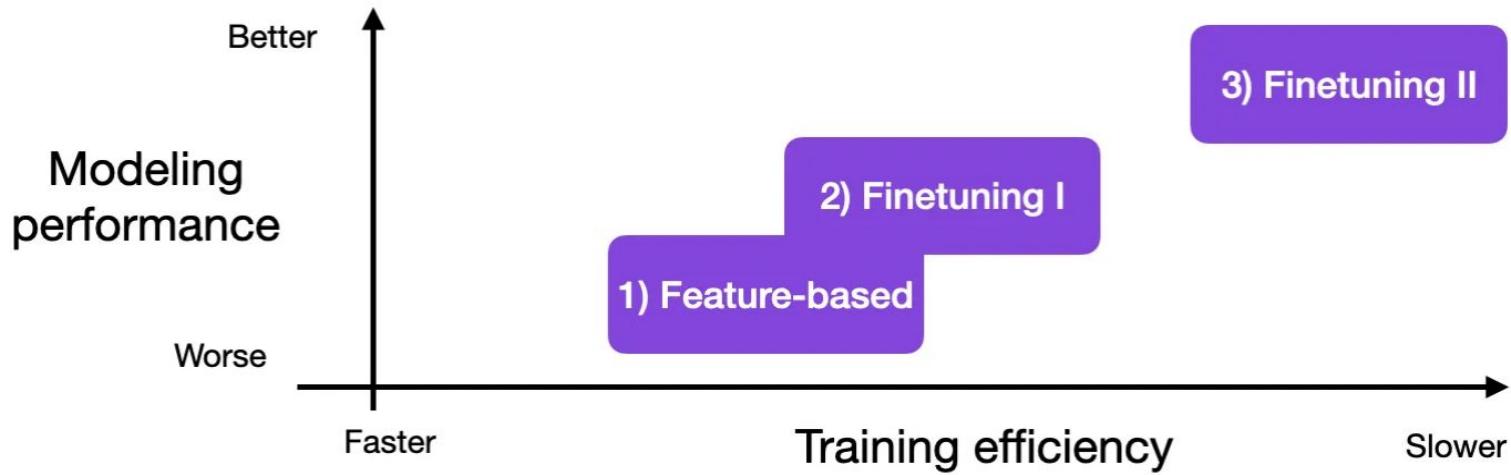


3) FINETUNING II



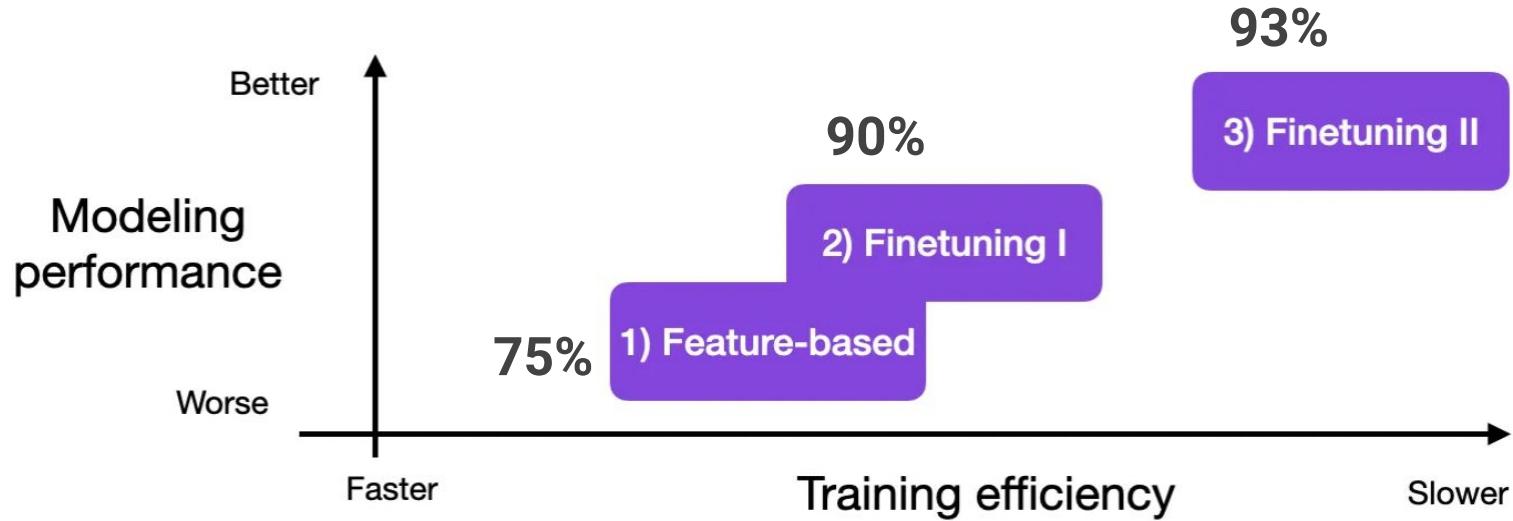
<https://magazine.sebastianraschka.com/p/finetuning-large-language-models>

Comparing LLM fine-tuning methods



Rule-of-thumb computational and modeling performance trade-offs for various approaches.

Comparing LLM fine-tuning methods



Rule-of-thumb computational and modeling performance trade-offs for various approaches.

Instruction Fine-Tuning

2

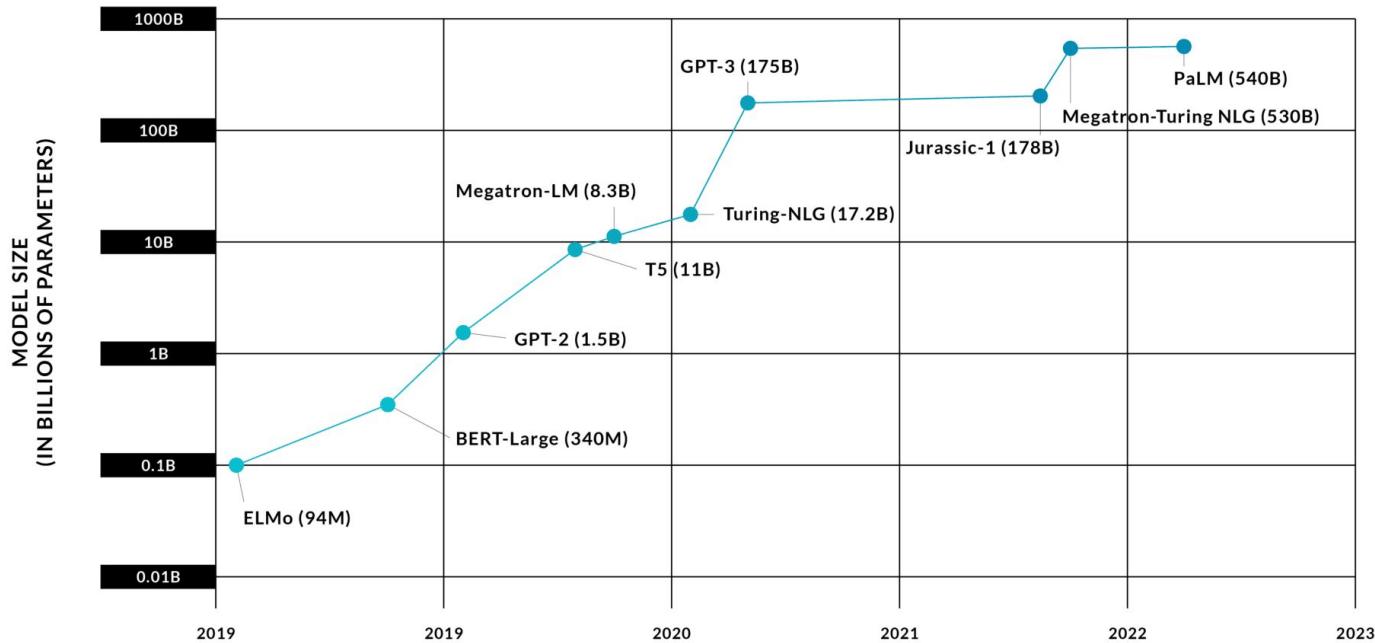
Supervised finetuning

More next-token prediction

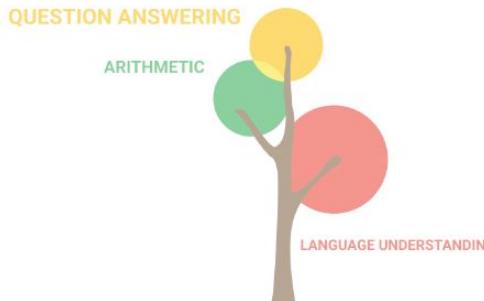
Usually 1k-50k instruction-response pairs

```
{
    "instruction": "Evaluate this sentence for spelling and grammar mistakes",
    "input": "He finnished his meal and left the restaurant",
    "output": "He finished his meal and left the restaurant.",
},
{
    "instruction": "Give three tips for staying healthy.",
    "input": "",
    "output": "\n        1. Eat a balanced diet. \\\n        2. Exercise regularly to keep your body active and strong. \\\n        3. Get enough sleep and maintain a consistent sleep schedule.",
}
```

Language Model Sizes Over Time



Understanding Large Language Models



8 billion parameters

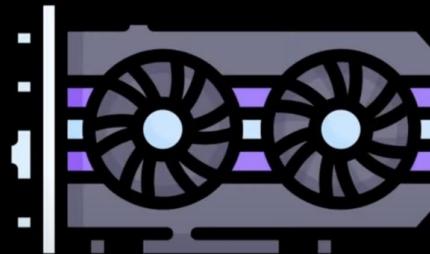




Time



Time



GPU Memory



Llama-2-7b



Requires \$800k to train



Llama-2-13B



Requires \$1.5M to train



Llama-2-70B



Requires \$8M to train

How to train & fine-tune under the small budget?

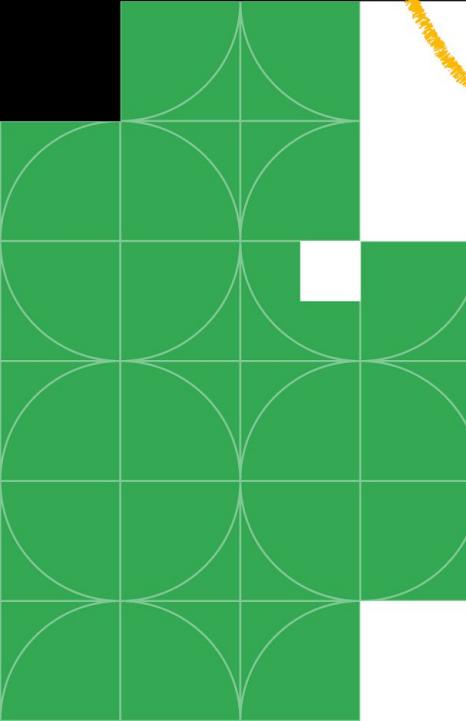


हम करते हैं प्रबंध
आप चिंता मत करिये

```
text  
  'Section Title',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
,
```

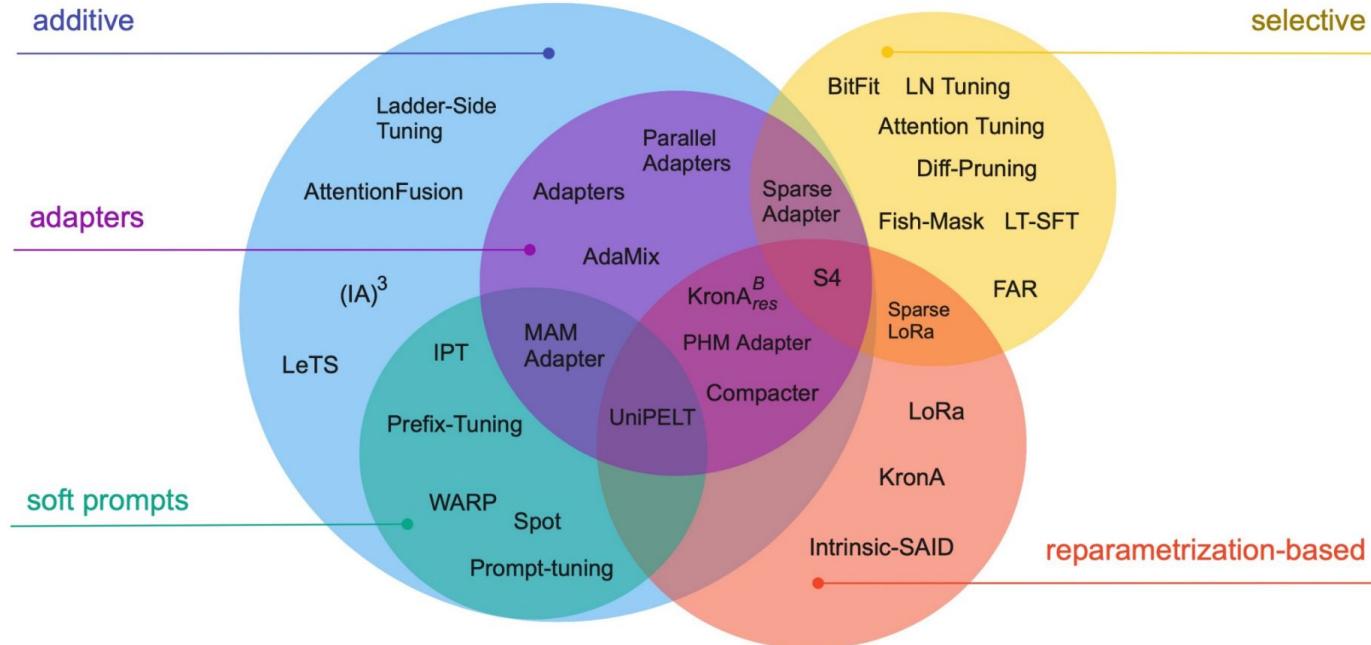
devfest

```
s.star,  
r: Colors.green[500],  
  
Text('23'),
```



Parameter-efficient fine-tuning

Parameter-efficient fine-tuning

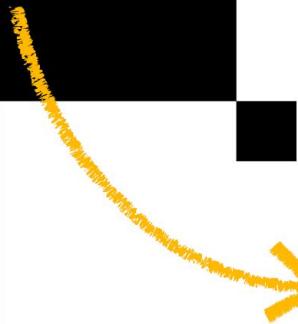
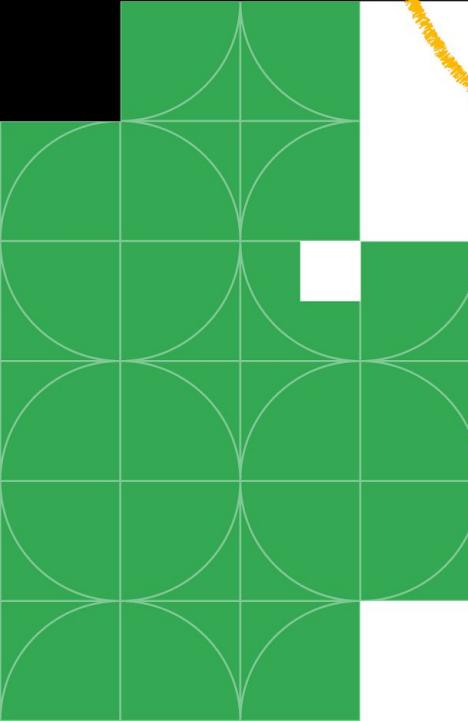


```
text  
  'Section Title',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
,
```

devfest

```
s.star,  
r: Colors.green[500],
```

```
Text('23'),
```



Low-rank adaptation (LoRA)

Low-rank adaptation (LoRA)

LoRA: LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS

Edward Hu* Yelong Shen* Phillip Wallis Zeyuan Allen-Zhu

Yuanzhi Li Shean Wang Lu Wang Weizhu Chen

Microsoft Corporation

{edwardhu, yeshe, phwallis, zeyuana,

yuanzhil, swang, luw, wzchen}@microsoft.com

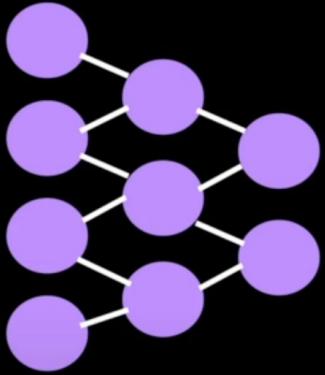
yuanzhil@andrew.cmu.edu

(Version 2)

ABSTRACT

An important paradigm of natural language processing consists of large-scale pre-training on general domain data and adaptation to particular tasks or domains. As we pre-train larger models, full fine-tuning, which re-trains all model parameters, becomes less feasible. Using GPT-3 175B as an example – deploying independent instances of fine-tuned models, each with 175B parameters, is prohibitively expensive. We propose Low-Rank Adaptation, or LoRA, which freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture, greatly reducing the number of trainable parameters for downstream tasks. Compared to GPT-3 175B fine-tuned with Adam, LoRA can reduce the number of trainable parameters by 10,000 times and the GPU memory requirement by 3 times. LoRA performs on-par or better than fine-tuning in model quality on RoBERTa, DeBERTa, GPT-2, and GPT-3, despite having fewer trainable parameters, a higher training throughput, and, unlike adapters, *no additional inference latency*. We also provide an empirical investigation into rank-deficiency in language model adaptation, which sheds light on the efficacy of LoRA. We release a package that facilitates the integration of LoRA with PyTorch models and provide our implementations and model checkpoints for RoBERTa, DeBERTa, and GPT-2 at <https://github.com/microsoft/LoRA>.





Rank

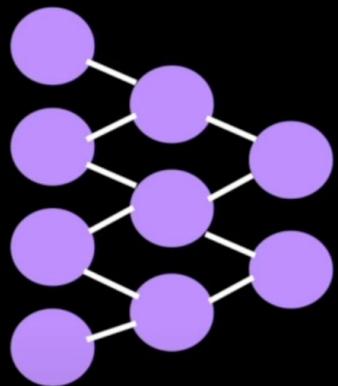
$$\begin{bmatrix} 2 & 20 & 1 \\ 4 & 40 & 2 \\ 6 & 60 & 3 \end{bmatrix}$$

Rank = 1

$$\begin{bmatrix} 2 & 20 & 1 \\ 4 & 70 & 2 \\ 6 & 60 & 3 \end{bmatrix}$$

Rank = 2

Rank



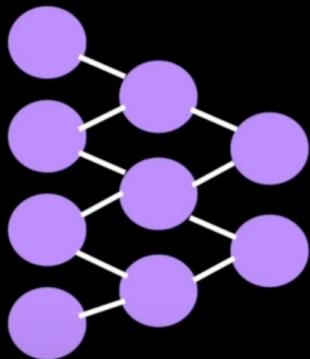
$$\begin{bmatrix} 2 & 20 & 1 \\ 4 & 40 & 2 \\ 6 & 60 & 3 \end{bmatrix}$$

Rank = 1

$$\begin{bmatrix} 2 & 20 & 1 \\ 4 & 70 & 2 \\ 6 & 60 & 3 \end{bmatrix}$$

Rank = 2

Rank



$$\begin{bmatrix} 2 & 20 & 1 \\ 4 & 40 & 2 \\ 6 & 60 & 3 \end{bmatrix}$$

Rank = 1

$$\begin{bmatrix} 2 & 20 & 1 \\ 4 & 70 & 2 \\ 6 & 60 & 3 \end{bmatrix}$$

Rank = 2

Matrix decomposition

items

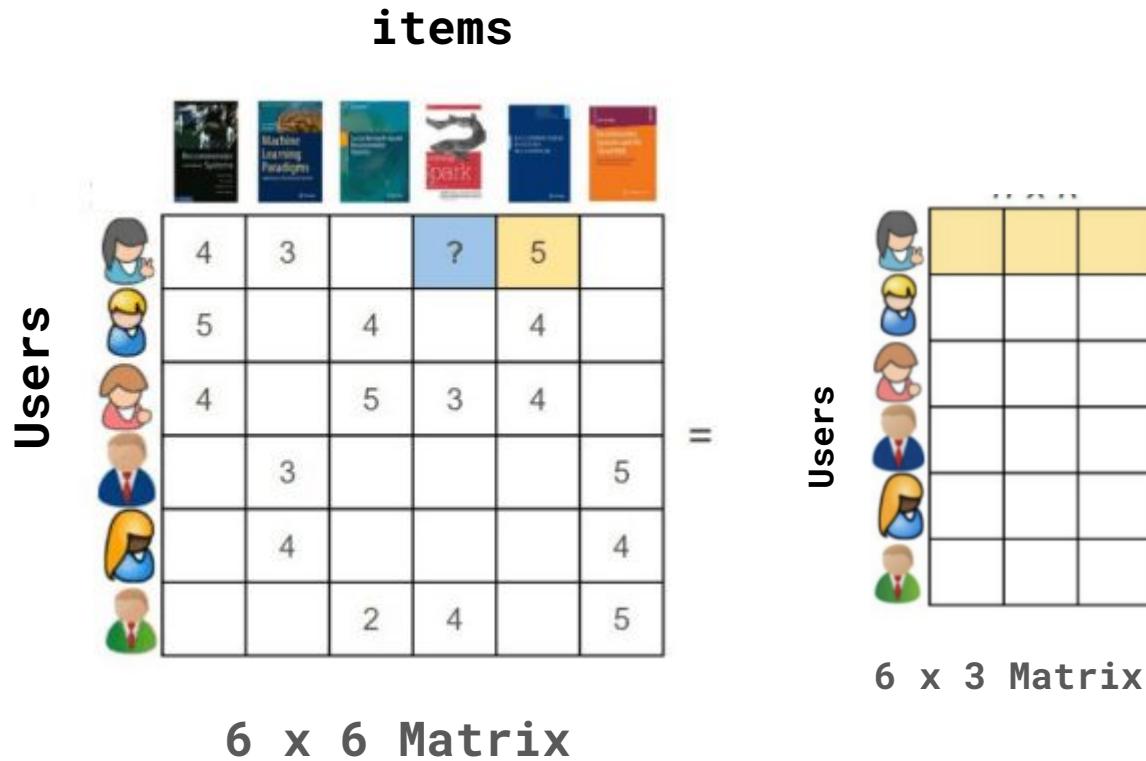


	items					
Users	4	3		?	5	
	5		4		4	
	4		5	3	4	
	3					5
	4					4
		2	4			5

=

6 x 6 Matrix

Matrix decomposition



Matrix decomposition

items

Users	item 1	item 2	item 3	item 4	item 5
User 1	4	3		?	5
User 2	5		4		4
User 3	4		5	3	4
User 4		3			
User 5	4				5
User 6		2	4		5

=

6 x 6 Matrix

Users	item 1	item 2	item 3
User 1	yellow	yellow	yellow
User 2			
User 3			
User 4			
User 5			
User 6			

6 x 3 Matrix

items

X

Items	User 1	User 2	User 3	User 4	User 5	User 6
Item 1						
Item 2						
Item 3						
Item 4						
Item 5						
Item 6						

3 x 6 Matrix

Rank Decomposition

$$\begin{bmatrix} 2 & 20 & 1 \\ 4 & 40 & 2 \\ 6 & 60 & 3 \end{bmatrix}$$

3×3



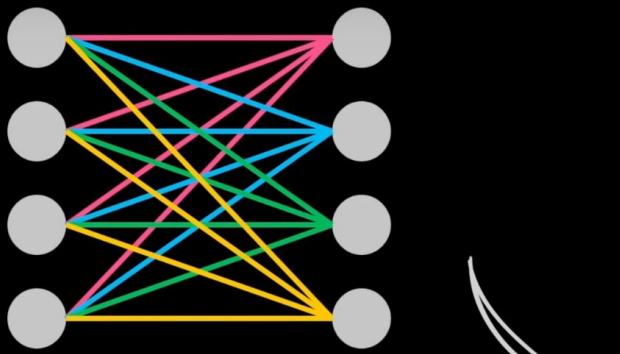
Rank Decomposition

$$\begin{bmatrix} 2 & 20 & 1 \\ 4 & 40 & 2 \\ 6 & 60 & 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \times [2 \ 20 \ 30]$$

3×3

Rank Decomposition

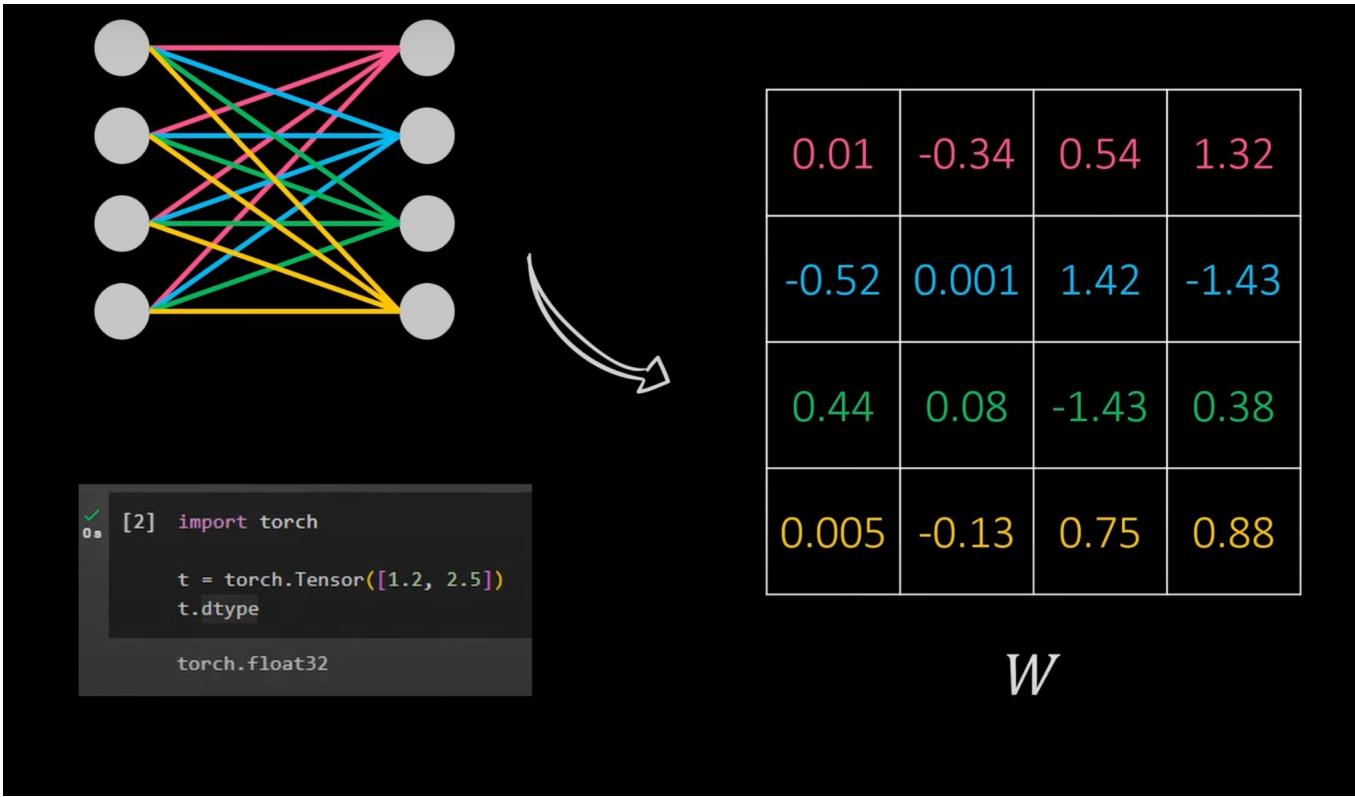
$$\begin{bmatrix} 2 & 20 & 1 \\ 4 & 40 & 2 \\ 6 & 60 & 3 \end{bmatrix}_{3 \times 3} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}_{3 \times 1} \times [2 \quad 20 \quad 30]_{1 \times 3}$$



A curved white arrow points from the neural network diagram to the right, indicating a connection between the two.

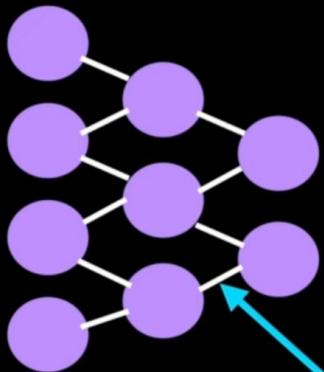
0.01	-0.34	0.54	1.32
-0.52	0.001	1.42	-1.43
0.44	0.08	-1.43	0.38
0.005	-0.13	0.75	0.88

W



Rank

number of linearly independent rows or columns a matrix



A green oval encloses a matrix of weights $W^{[l]}$. The matrix has three columns and $n^{[l]}$ rows. The columns are labeled $w_{11}^{[l]}, w_{12}^{[l]}, \dots, w_{1n^{[l-1]}}^{[l]}$, $w_{21}^{[l]}, w_{22}^{[l]}, \dots, w_{2n^{[l-1]}}^{[l]}$, and so on. The rows are labeled $w_{n^{[l]}1}^{[l]}, w_{n^{[l]}2}^{[l]}, \dots, w_{n^{[l]}n^{[l-1]}}^{[l]}$. Ellipses indicate intermediate rows and columns.

$$\begin{bmatrix} w_{11}^{[l]} & w_{12}^{[l]} & \cdots & w_{1n^{[l-1]}}^{[l]} \\ w_{21}^{[l]} & w_{22}^{[l]} & \cdots & w_{2n^{[l-1]}}^{[l]} \\ \vdots & \vdots & \cdots & \vdots \\ w_{n^{[l]}1}^{[l]} & w_{n^{[l]}2}^{[l]} & \cdots & w_{n^{[l]}n^{[l-1]}}^{[l]} \end{bmatrix}$$

Low-rank adaptation



**min # independent
rows/columns**

Rank is smaller than dimensions

Low-rank adaptation

min # independent
rows/columns

$$\text{rank} \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 3 \\ 3 & 0 & 1 \end{pmatrix} = 3$$

$$\text{rank} \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 3 \end{pmatrix} = 2$$

Rank is smaller than dimensions



Low-rank adaptation



min # independent
rows/columns

$$\text{rank} \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 3 \\ 3 & 0 & 1 \end{pmatrix} = 3$$

$$\text{rank} \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 3 \end{pmatrix} = 2$$

Rank is smaller than dimensions

Low-rank adaptation

min # independent
rows/columns

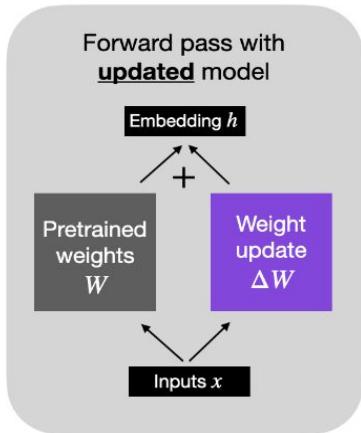
Fine-tuning of models

$$\text{rank} \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 3 \\ 3 & 0 & 1 \end{pmatrix} = 3$$

 W_0  $W_0 + \Delta W$

Low-rank adaptation (LoRA)

Alternative formulation (regular finetuning)



$$W_0 + \Delta W = W_0 + BA$$

0.15	-0.14	-0.21	0.612
-0.22	0.204	0.308	-0.86
-0.30	-0.16	0.634	0.147
-0.07	-0.2	0.246	0.523

 ΔW

Shape: (4, 4)

0.3	-0.14
-0.42	0.201
0.46	0.38

 B

Shape: (4, 2)

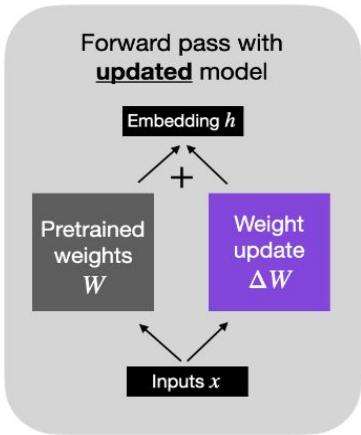
0.1	-0.44	0.04	1.42
-0.92	0.1	1.62	-1.33

 A

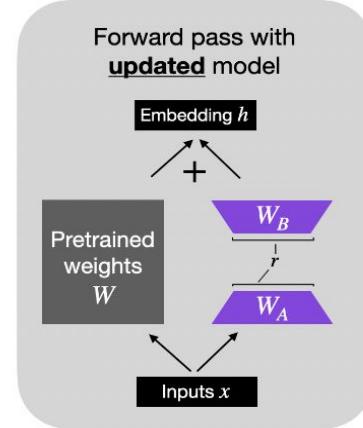
Shape: (2, 4)

Low-rank adaptation (LoRA)

Alternative formulation (regular finetuning)



LoRA weights, W_A and W_B , represent ΔW



```
def regular_forward_matmul(x, W):
    h = x @ W
    return h

def lora_forward_matmul(x, W, W_A, W_B):
    h = x @ W # regular matrix multiplication
    h += x @ (W_A @ W_B)*alpha # use scaled LoRA weights
    return h
```

$$W_0 + \Delta W = W_0 + BA$$



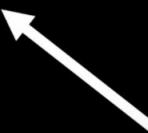
$B \in \mathbb{R}^{d \times r}$

$A \in \mathbb{R}^{r \times k}$

rank $r \ll \min(d, k)$

$$\frac{\alpha}{r}$$

Scaling factor



Rank

$$W_0 + \Delta W = W_0 + BA$$

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

$$A \in \mathbb{R}^{r \times k}$$

$$\text{rank } r \ll \min(d, k)$$



└

$$\frac{\alpha}{r}$$

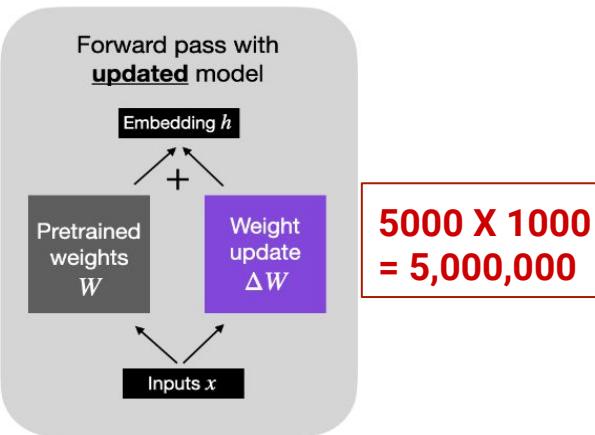
Scaling factor

$$r$$

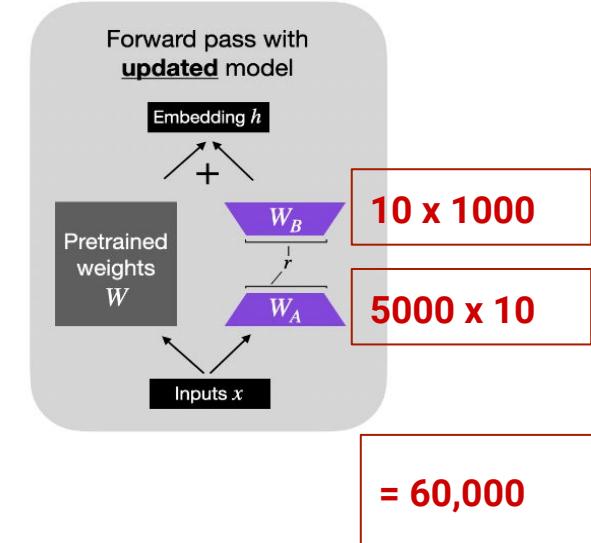
Rank

Low-rank adaptation (LoRA)

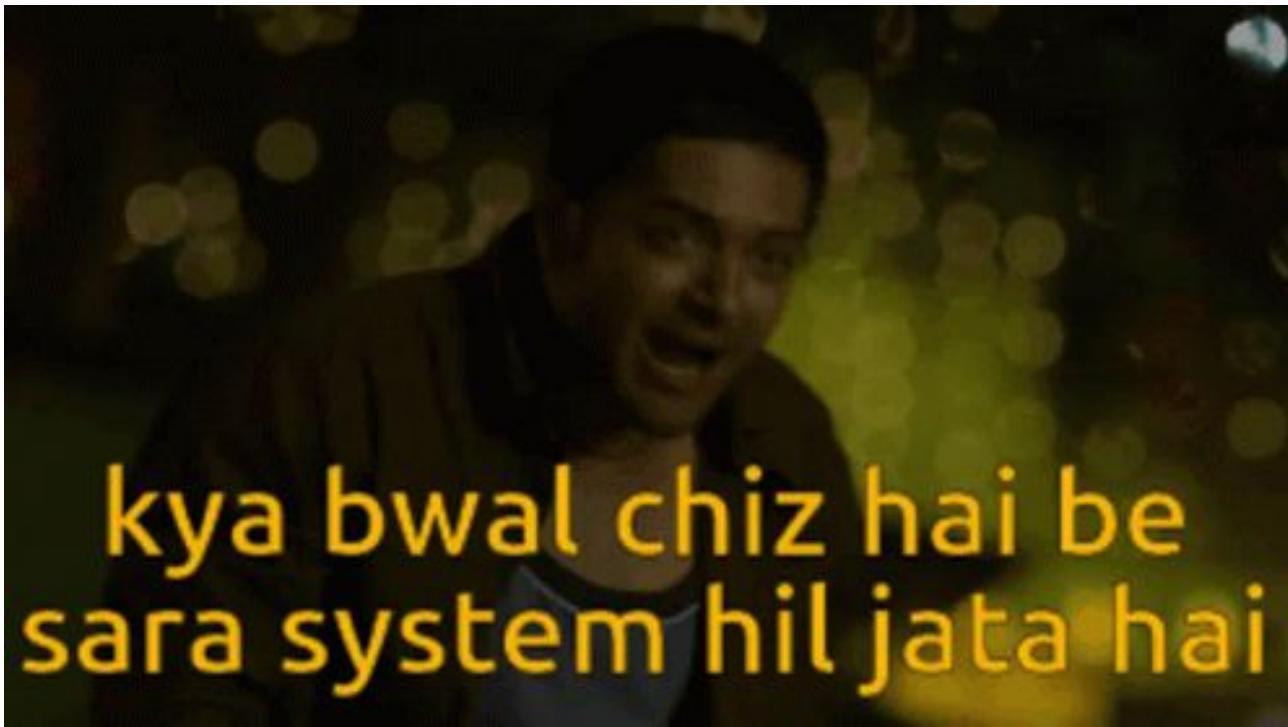
Alternative formulation (regular finetuning)



LoRA weights, W_A and W_B , represent ΔW

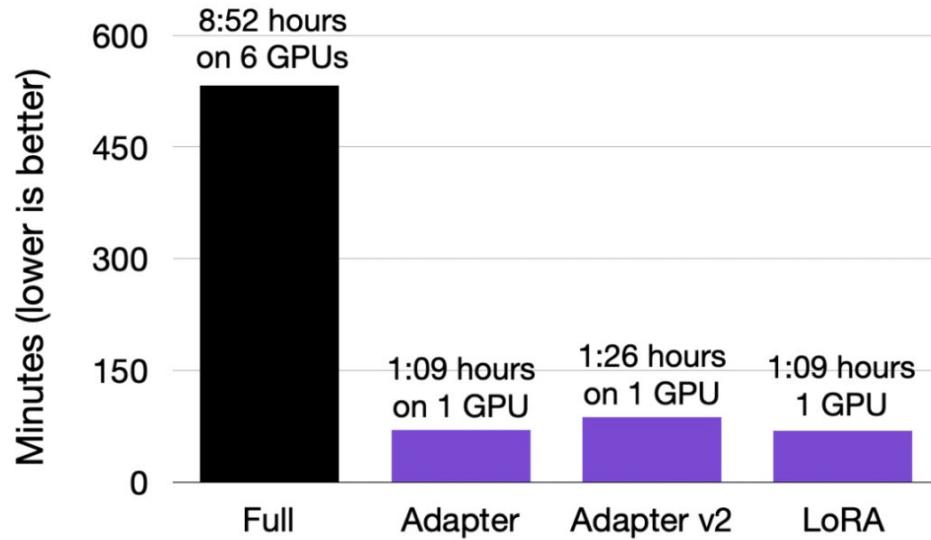


Parameter Efficient Fine-tuning



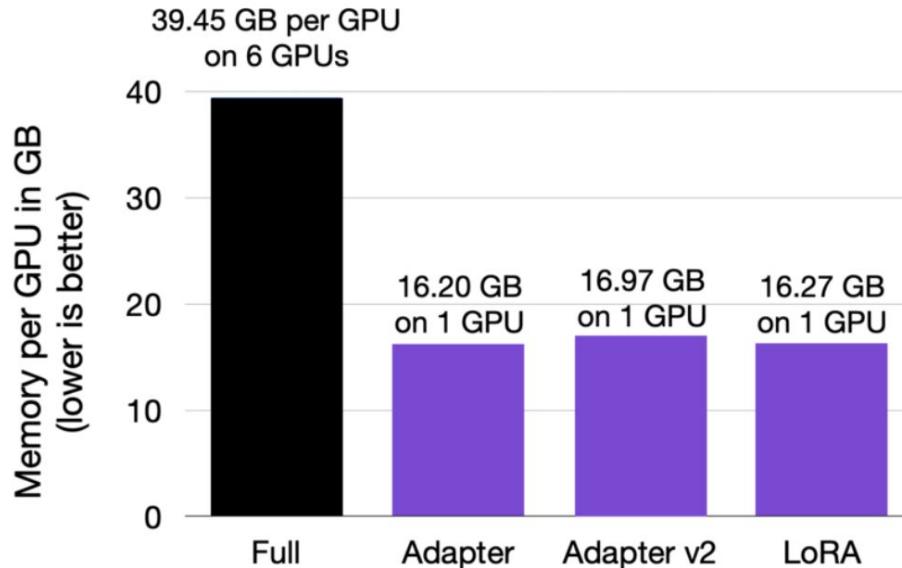
Low-rank adaptation (LoRA)

Time to complete 52k training iterations for Falcon 7B



Low-rank adaptation (LoRA)

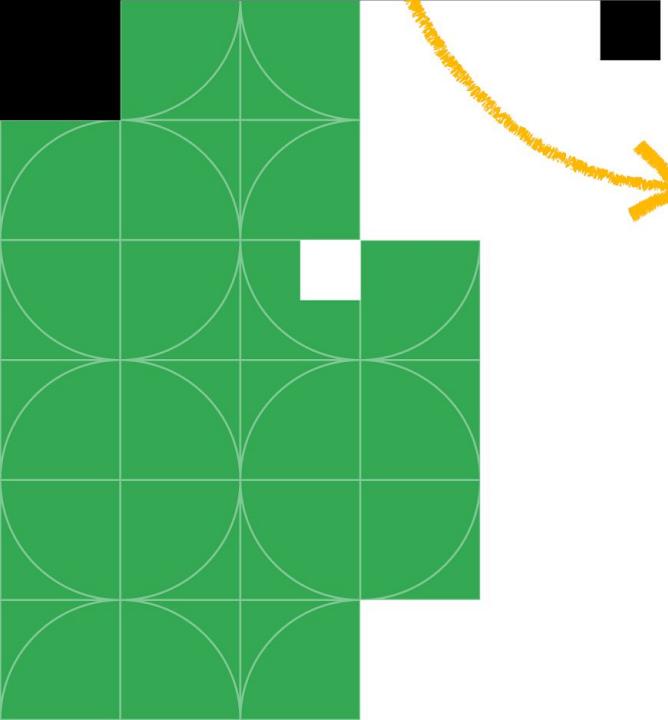
Memory requirements per GPU for Falcon 7B



```
text  
  'Section Title',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
,
```

devfest

```
s.star,  
r: Colors.green[500],  
  
Text('23'),
```



Open Source LLMs

LLaMA

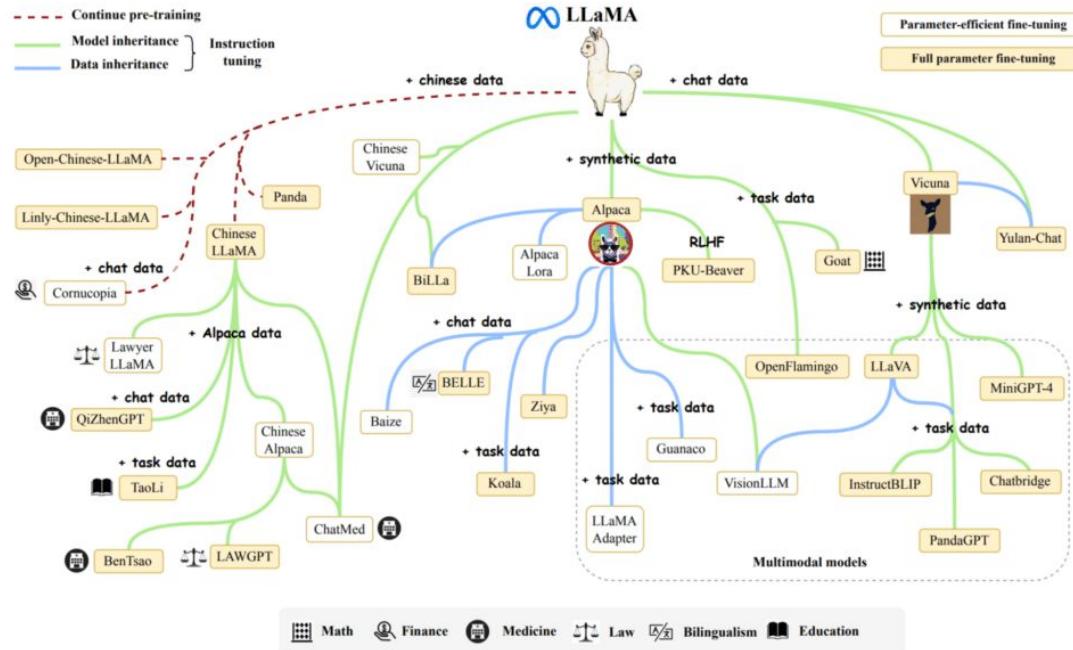


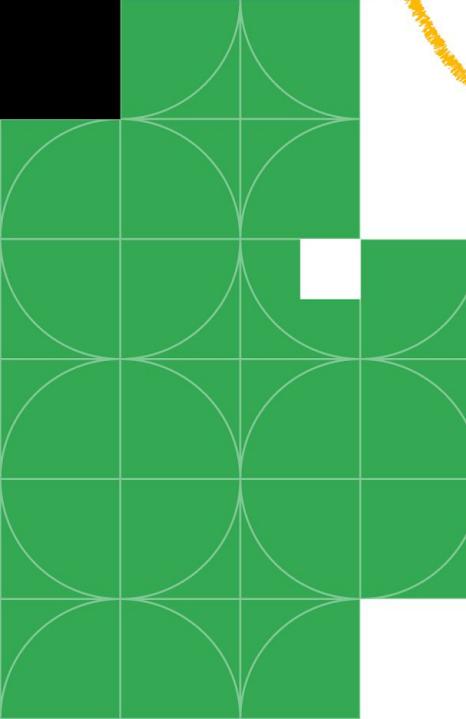
Fig. 4: An evolutionary graph of the research work conducted on LLaMA. Due to the huge number, we cannot include all the LLaMA variants in this figure, even much excellent work. To support incremental update, we share the source file of this figure, and welcome the readers to include the desired models by submitting the pull requests on our GitHub page.

```
text  
  'Section Title',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
,
```

devfest

```
s.star,  
r: Colors.green[500],
```

```
Text('23'),
```

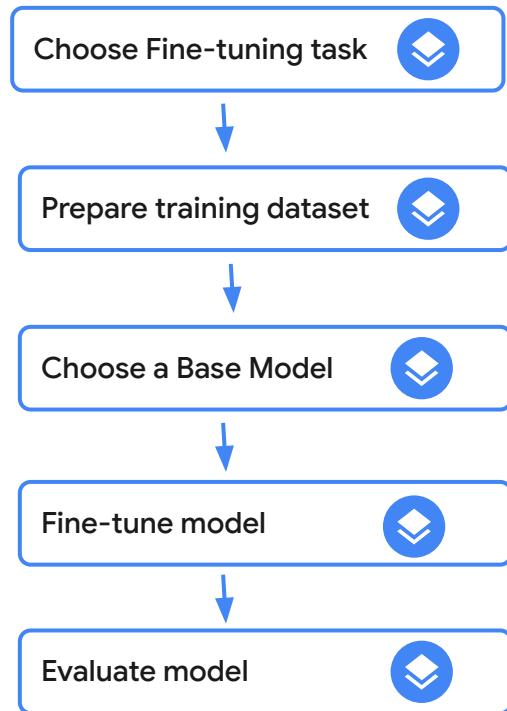


Fine-tuning LLaMA-2 on Personal Whatsapp data



Abhi maza ayega na bhidu

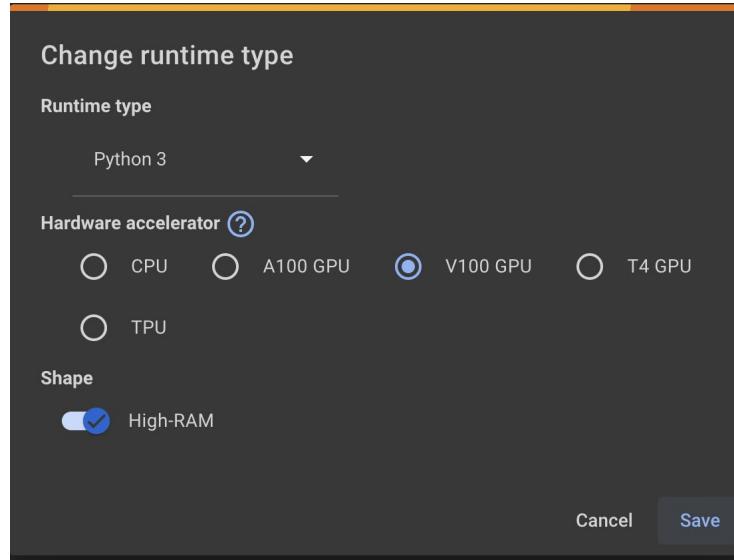
5 Steps



Input	Output



Setup



```
[1] pip3 install accelerate peft bitsandbytes transformers trl
```

Import Required Packages

```
▶ import os
  import torch
  from datasets import load_dataset
  from transformers import (
      AutoModelForCausalLM,
      AutoTokenizer,
      BitsAndBytesConfig,
      HfArgumentParser,
      TrainingArguments,
      pipeline,
      logging,
  )
  from peft import LoraConfig, PeftModel
  from trl import SFTTrainer
```

Prepare dataset

```
[ ] formatter = ChatDatasetFormatter('chat_dataset.json', 'Aadi', 'chat_dataset.csv')
formatter.prepare_dataset()
```



Context	Reply
Abhijeet Kerla: What are you working on as res...	Working on Frozen adapters in Bert paper How's...
Abhijeet Kerla: What are you working on as res...	What was the issue?
Abhijeet Kerla: What are you working on as res...	Yup About to message you

Model setup

```
# Load base model

model = AutoModelForCausalLM.from_pretrained(
    pretrained_model_name_or_path="NousResearch/Llama-2-7b-chat-hf",
    quantization_config=quant_config,
    device_map={"": 0},
)
model.config.pretraining_tp = 1
```

Trainer Setup

```
[ ] # Load LoRA configuration
peft_args = LoraConfig(
    lora_alpha=16,
    lora_dropout=0.1,
    r=64,
    bias="none",
    task_type="CAUSAL_LM",
)
```

```
▶ # Set supervised fine-tuning parameters
trainer = SFTTrainer(
    model=model,
    train_dataset=dataset,
    peft_config=peft_args,
    dataset_text_field="text",
    max_seq_length=None,
    tokenizer=tokenizer,
    args=training_params,
    packing=False,
)
```

Train

```
[ ] # Train model  
    trainer.train()
```

```
warnings.warn(
```

```
[187/250 05:43 < 01:56, 0.54 it/s, Epoch 0.74/1]
```

Step	Training Loss
25	1.408300
50	1.656600
75	1.213100
100	1.443900
125	1.176500
150	1.366400
175	1.173500

Train W/O Parameter efficient



~\$500

30 GB of storage

~ 100GB of GPU RAM



Train W/O Parameter efficient



~\$500

30 GB of storage

~ 100GB of GPU RAM



Train With Parameter efficient

Free

~100 Mb of storage

~ 16 GB of GPU RAM



LLMtuner



Scan me!



The screenshot shows the `README.md` file of the LLMtuner GitHub repository. It features a circular icon of a llama wearing headphones and looking at a computer screen. The text below the icon reads "LLMTuner". A horizontal line follows, with the text "LLMTuner: Fine-Tune Llama, Whisper, and other LLMs with best practices like LoRA, QLoRA, through a sleek, scikit-learn-inspired interface." Below this are several small buttons: "License Apache 2.0", "PRs welcome", "Discord Community", and "Open in Colab". Another horizontal line follows, with the section title "Installation". Under "With pip", it says "This repository is tested on Python 3.7+". At the bottom, there is a note: "You should install Premonition using the command".

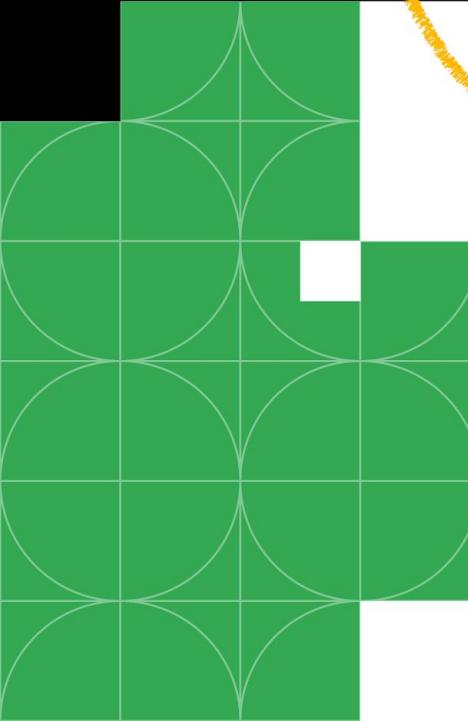
<https://github.com/promptslab/LLMtuner/>

```
text  
  'Section Title',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
,
```

devfest

```
s.star,  
r: Colors.green[500],
```

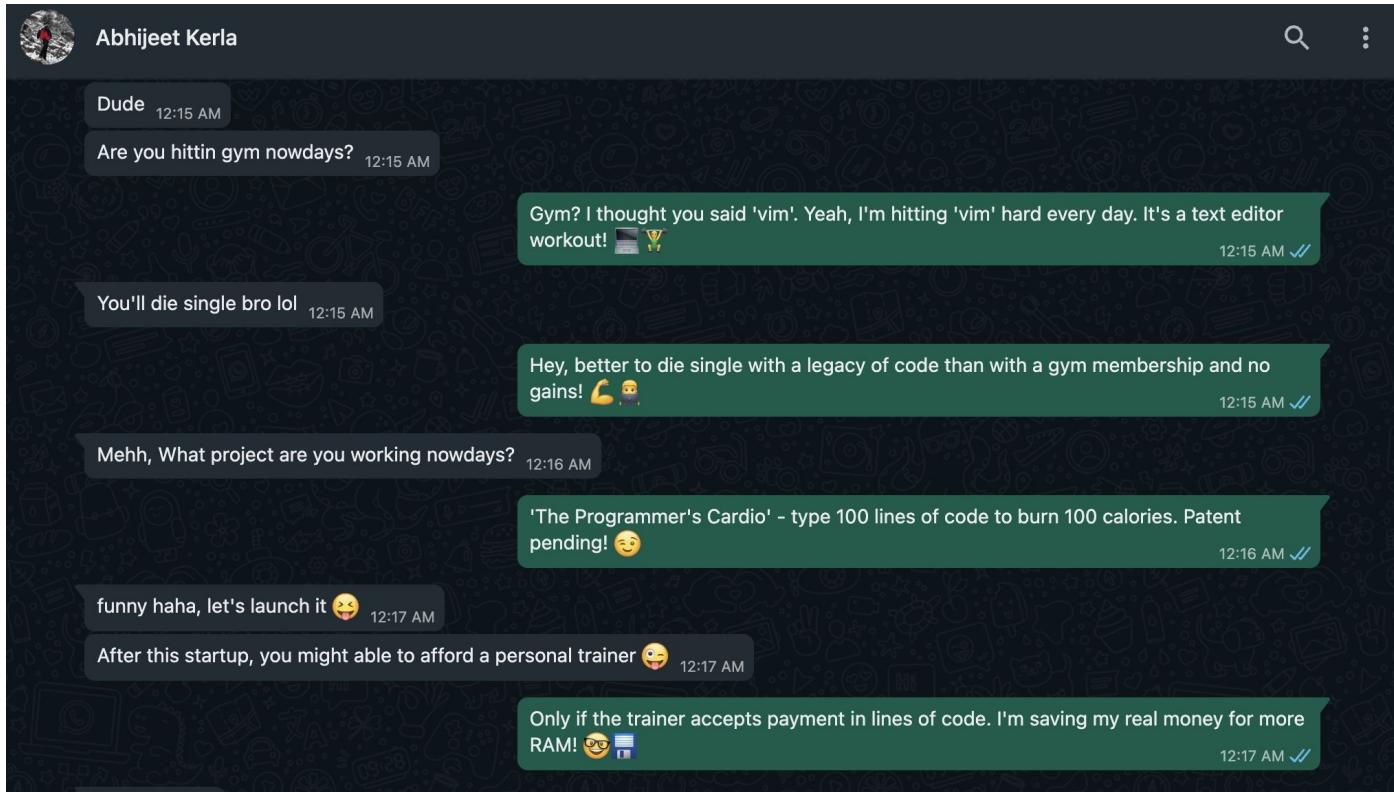
```
Text('23'),
```



Google Developer Groups

Inference

LLaMA is talking behalf of me



LLaMA is talking behalf of me

Sherya
online

Hey, did you submit your timesheet? Prithvi is asking. 12:21 AM

Uh, sorry, but I have a bit of an emergency situation going on right now. My cat has gotten into my computer and deleted all my files, including my timesheet. 12:21 AM ✓

?? 12:22 AM

Haha, I wish I had a backup. Unfortunately, I didn't think to save everything to an external hard drive or something. 12:22 AM ✓

Kindly check your slack and reply Prithvi !! 12:22 AM

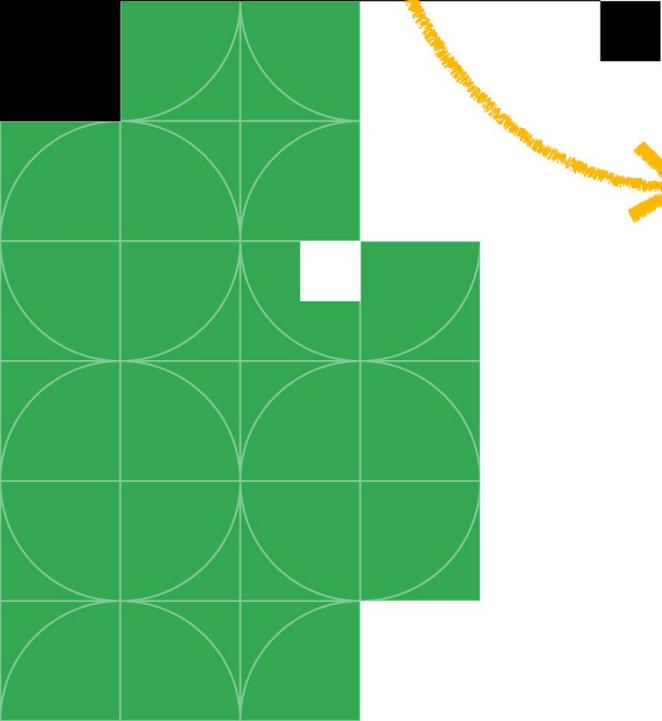
Uh, I guess I'll start. [laughing nervously] 12:22 AM ✓

```
text  
  'Section Title',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
,
```

devfest

```
s.star,  
r: Colors.green[500],
```

```
Text('23'),
```



Take Home Messages

LLaMA is talking behalf of me

- Use Parameter Efficient Fine-tuning (Peft) for large language models for domain specific task.

LLaMA is talking behalf of me

- Use Parameter Efficient Fine-tuning (Peft) for large language models for domain specific task.
- Fine-tune 7B parameter models quickly on a 14 GB RAM GPU.

LLaMA is talking behalf of me

- Use Parameter Efficient Fine-tuning (Peft) for large language models for domain specific task.
- Fine-tune 7B parameter models quickly on a 14 GB RAM GPU.
- Utilize free online GPUs like Google Colab and Kaggle.

LLaMA is talking behalf of me

- Use Parameter Efficient Fine-tuning (Peft) for large language models for domain specific task.
- Fine-tune 7B parameter models quickly on a 14 GB RAM GPU.
- Utilize free online GPUs like Google Colab and Kaggle.
- Apply LoRA to all layers for optimal performance.

LLaMA is talking behalf of me

- Use Parameter Efficient Fine-tuning (Peft) for large language models for domain specific task.
- Fine-tune 7B parameter models quickly on a 14 GB RAM GPU.
- Utilize free online GPUs like Google Colab and Kaggle.
- Apply LoRA to all layers for optimal performance.
- Optimizer choice is less crucial due to similar top optimizer benchmarks.

Thank you!

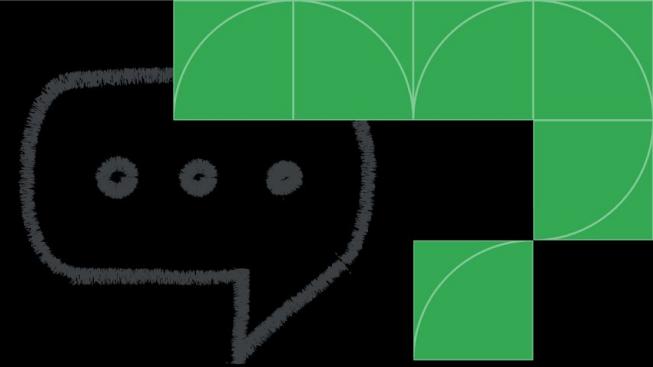
Let's get connected
on Twitter! I am
[@aadityaura](#)



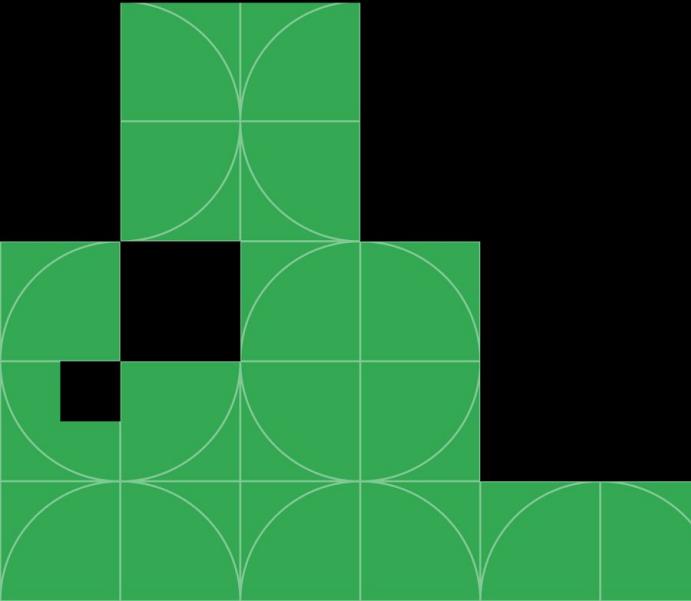
```
text  
'Simple Statement or URL',  
style: TextStyle(  
color: Colors.green[200],  
)  
,
```

devfest

```
s.star,  
r: Colors.green[500],  
  
Text('23'),
```



Lucknow AI (UP-AI.org)



Lucknow AI



@ailucknow



<http://lucknowai.org>

Lucknow AI , Tensorflow Group
Lucknow
WhatsApp community

Lucknow AI

Search Lucknow AI

Lucknow AI on GitHub

Lucknow AI

Open Source AI Research & Mentorship

Get started now Project Awadhi

X 🧪 📱 📸

Empower the idea of AI and knowledge sharing throughout Lucknow

Education and Knowledge Sharing

Promoting AI & ML Research

Active Engagement in Open Source

Supporting Startup Ventures

Solving Local Challenges with AI

अवानी NLP and Data Advancement

Collaborations

TensorFlow
User Group Lucknow



TensorFlow

User Group Lucknow



Google AI



Scan me!

 @TFUGlucknow

 @AILucknow

 <http://lucknowai.org>

