Fine-Tuning LLMs

Krish Thakker

What is Fine-Tuning?

Base Model

(davinci (GPT-3))

Prompt: Tell me how to fine-tune a model

Completion:

How can I control the complexity of a model?
How do I know when my model is done?
How do I test a model?
How do I define a task for my model?
How do I scale a model?
How do I make a probabilistic model?
How do I start using a model once it's complete?
How do I cope with missing data?
How do I make my modeling efforts sustainable?

Fine-tuned Model

(text-davinci-003)

Prompt: Tell me how to fine-tune a model

Completion:

Fine-tuning a model involves adjusting the parameters of a pre-trained model in order to make it better suited for a given task. There are generally three steps to fine-tuning a model:

1. Select a base model - Select a pre-trained deep learning model that has been trained on a large dataset.
2. Adjust parameters - Adjust parameters of the pre-trained model to better suit the desired task. This may include changing the number of layers, adjusting learning rate, adding regularization, or tweaking the optimizer.
3. Train the model - Train the new model on the desired dataset. The amount of data and the amount of training required will depend on the task and the model.

Terms To Know

LLM = Large Language Model (a subset of machine learning models)

PEFT = Parameter-Efficient Fine-Tuning

LoRA = Low-Rank Adaptation

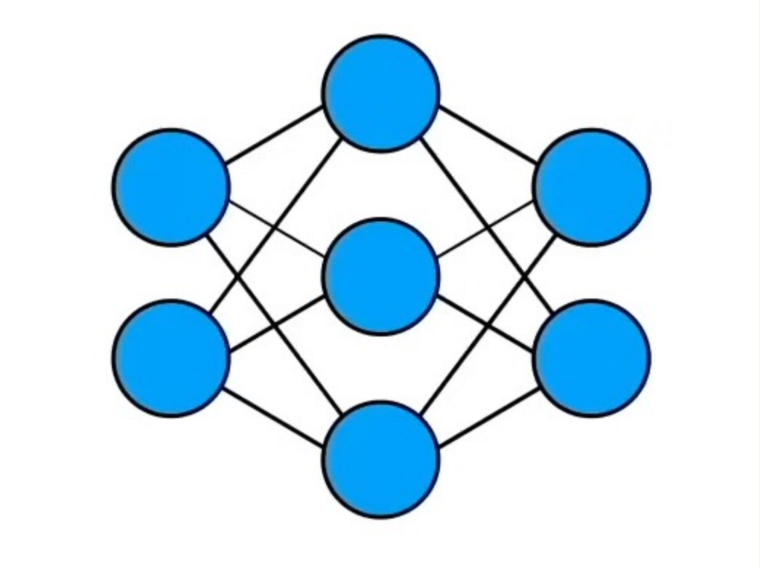
Hugging Face = library of machine learning models and datasets

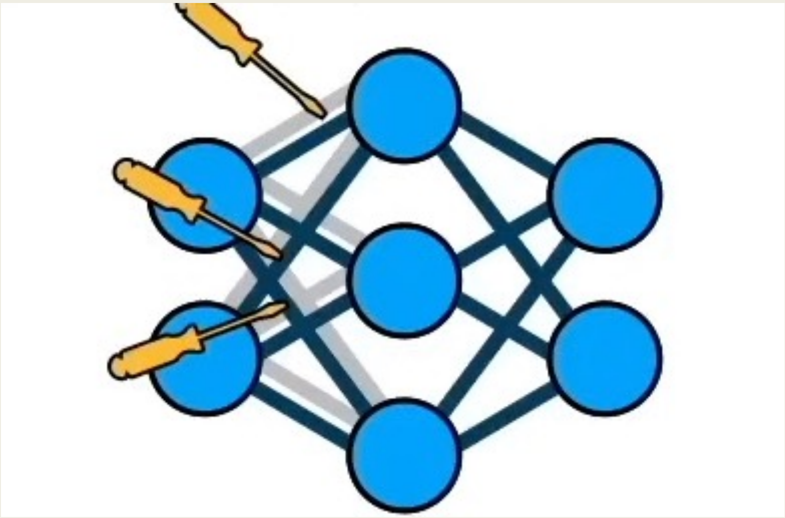
BERT = Bidirectional Encoder Representations from Transformers (an LLM used for natural language processing)

DistilBERT = smaller, faster cheaper version of BERT

High-Level Overview

- Load a dataset to train LLM
- Set metrics to judge accuracy
- Test untrained model
- Fine-tune model with PEFT/LoRA
- Test fine-tuned model



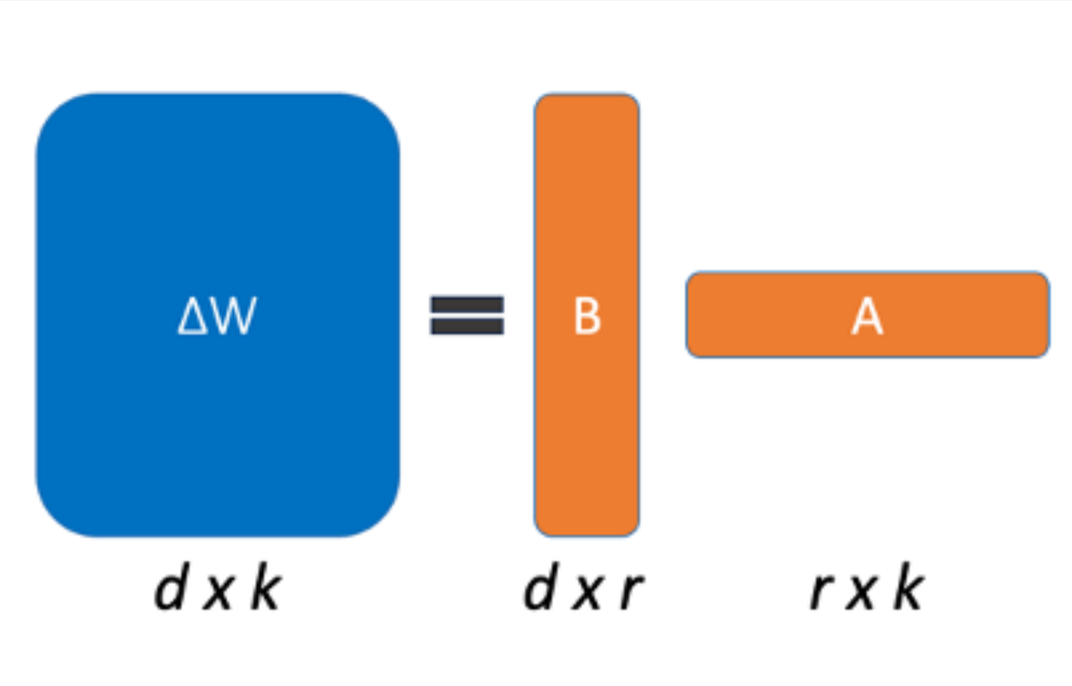


What is PEFT?

- Parameter-Efficient Fine-Tuning
- Freezes most of the parameters of a model
- Only fine-tunes a few of the parameters
- Optimizes time and cost of fine-tuning process

What is LoRA?

- Low-Rank Adaptation: A PEFT technique
- Adds new trainable parameters
- Let d = 1,000, k = 1000
- Ex: 1,000 x 1,000 parameter model = 1,000,000 parameters
- Set a low rank: r = 4.
- 4,000 + 4,000 = 8,000 < 1,000,000 parameters
- 0.8%



Install + Import

- Evaluate
- Datasets
- PEFT
- Transformers

Install Packages

```
[1]: !pip install evaluate datasets peft transformers
```

•••

Imports

```
[2]: from datasets import load_dataset, DatasetDict, Dataset

     from transformers import (
         AutoTokenizer,
         AutoConfig,
         AutoModelForSequenceClassification,
         DataCollatorWithPadding,
         TrainingArguments,
         Trainer)

     from peft import PeftModel, PeftConfig, get_peft_model, LoraConfig
     import evaluate
     import torch
     import numpy as np
```

Dataset

- Load imdb-truncated dataset
- From Hugging Face library
- Display training data

Dataset

```
[3]: # load imdb-truncated dataset
     dataset = load_dataset('shawhin/imdb-truncated')
```

```
Downloading readme: 100%  592/592 [00:00<00:00, 57.5kB/s]
Downloading data: 100%  836k/836k [00:00<00:00, 2.70MB/s]
Downloading data: 100%  853k/853k [00:00<00:00, 4.27MB/s]
Generating train split: 100%  1000/1000 [00:00<00:00, 3592.63 examples/s]
Generating validation split: 100%  1000/1000 [00:00<00:00, 67959.17 examples/s]
```

```
[4]: dataset
```

```
[4]: DatasetDict({
         train: Dataset({
             features: ['label', 'text'],
             num_rows: 1000
         })
         validation: Dataset({
             features: ['label', 'text'],
             num_rows: 1000
         })
     })
```

```
[5]: # display % of training data with label=1
     np.array(dataset['train']['label']).sum()/len(dataset['train']['label'])
```

```
[5]: 0.5
```

Datasets: shawhin / **imdb-truncated** like 4

Dataset card | Viewer | Files and versions | Community 3

Split (2)
train · 1k rows

Search this dataset

label (int64)	text (string · lengths)
	0 1 / 130 6.3k
1	. . . or type on a computer keyboard, they'd probably give this eponymous film a rating of "10." After all, no elephants are shown being killed during the movie; it is not even implied that any are hurt. To the contrary, the master of ELEPHANT WALK, John Wiley (Peter Finch), complains that he cannot shoot any of the pachyderms--no matter how menacing--without a permit from the…
1	During 1933 this film had many cuts taken from it because it was very over the top for the story content and the fact that Lily Powers,(Barbara Stanwyck) would do anything to obtain great wealth and power. Lily's father had forced his daughter into prostitution at the age of 14 and she grew up in a steel mill of a town with very poor people and her father ran a speakeasy…
0	Let me be clear. I've used IMDb for years. But only today I went through the trouble of registering on the site, just so I could give this movie the lowest possible rating. I've seen hundreds of films, some of them bad, a few awful. Never, though, have i seen such a contrast of pretense and incompetence, of high intentions and failure. Mira Sorvino is horribl…
1	Carlos Mencia was excellent this is hour special. He was working hard to show everybody he was the real deal. I know people have said he's stolen material in this special, but that is not true. Carlos brings comedy up front the way he wants it, not how anyone else wants it, that is why he is so good. People say he's not funny because he says Dee dee dee too much, and they…
1	I was initially dubious about this movie (merely because of the subject), but the richly drawn characters, the fabulous scenes of the buffalo hunt, and the dramatic conclusion make it well-worth watching. I initially had trouble distinguishing between the two buffalo hunters but as the movie progressed they increasingly distinguished themselves. I am still haunted by the fina…
0	No matter how well meaning his "message" is - this film is a terribly made trainwreck - awful acting, lame camera work - I do not know why Carr agreed to try and pull off a stutter - he is lousy at it. You watch the extras on the DVD and the way he has a camera follow him around - he just soaks it up - he loves being the center of attention. He is a bad actor - he reminds me…
1	When you typically watch a short film your always afraid that the person creating the film tries to throw too much into it. That's not the case with this one. A great story about a young girl who's had enough and other worldly forces trying to help make things right. Eric Etebari does a wonderful job of representing the spirit of twisted justice and helps to…
1	How to lose friends and Alienate people came out in 2008. It bombed at U.S. Box offices. It's an absolutely hilarious film with a great cast. Simon Pegg is great playing Sidney Young, who wrote the book "How to lose friends and Alienate people. I know it's not a true story. The only way I know that is because Sidney wants to go out with an actress named Sophie Maes. Sophie…
0	#1 Vampires vs. Humans #2 Military-reject roughneck squad as first responders to dangerous, unknown Vampire incursions. #3 Sexy female Vampire on the side of the "good guys". #4 Plenty of gore and action. There are four (4) major plot devices that may help you decide if you want to watch this movie. If you want all four, then the nex…
1	This movie is intelligent. That is, more than most other movies, it transcends the least common denominator - stupid people will probably not appreciate it. The story also relies heavily on dialogue. It has some parallels to Lost in Translation, although Before Sunrise is much brighter, somehow less abstract, and simply a lot better. The script, the characters and…
1	THE PERVERT'S GUIDE TO CINEMA (2007) **** If Loving Cinema Makes Me A Pervert, So Be It! If you are a true 'moviefreak' like me then I'm sure you can't get enough of films about film-making and I don't mean necessarily the dry documentary know and then. I mean a total discourse on the film viewing experience. Well if that's the case have I got a lulu of…
0	Previous comments encouraged me to check this out when it showed up on TCM, but it was a severe disappointment. Lupe Valdez is great, but doesn't get enough screen time. Frank Morgan and Eugene Palette play familiar but promising characters, but the script leaves them stranded. The movie revolves around the ego of Lee Tracy's character, who is at best a self-…
0	The wife of a stage producer in London hopes to fix up the American song-and-dance man starring in her husband's latest show with an acquaintance, an American girl who makes her living modeling fashions in society circles. Unfortunately, the couple has already met on their own, with the girl thinking the guy is actually the show producer married to her friend (the fact…
0	This had high intellectual pretensions.The main lead intends to give a "deep" "meaningful" rendering(with voice over for his frames of mind naturally) and he was certainly influenced by the

< Previous 1 2 3 ... 10 Next >

Metrics + Model

- DistilBERT is used
  - *Faster model for testing purposes*
- Define metric labels
- Generate model

Set Metrics and Load Model

```
[6]: # using DistilBERT, a smaller model
model_checkpoint = 'distilbert-base-uncased'

# define label maps
id2label = {0: "Negative", 1: "Positive"}
label2id = {"Negative":0, "Positive":1}

# generate classification model from model_checkpoint
model = AutoModelForSequenceClassification.from_pretrained(
    model_checkpoint, num_labels=2, id2label=id2label, label2id=label2id)
```

•••

```
[7]: # display architecture
model
```

•••

Accuracy

- Evaluates accuracy
- Allows model to fine-tune effectively

Accuracy Evaluation

[9]:
```python
# import accuracy evaluation metric
accuracy = evaluate.load("accuracy")
# define an evaluation function to pass into trainer later
def compute_metrics(p):
    predictions, labels = p
    predictions = np.argmax(predictions, axis=1)

    return {"accuracy": accuracy.compute(predictions=predictions, references=labels)}
```

Test Untrained Model

- List of movie ratings
- Predictions for +/- are unknown
- 60% accuracy

Apply Untrained Model to Text

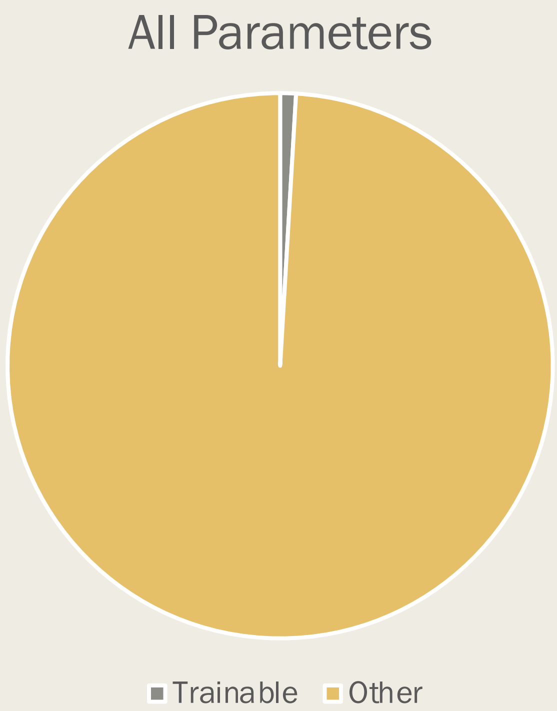
[10]:
```python
# define list of examples
text_list = ["It was good.", "Not a fan, don't recommed.", 
             "Better than the first one.", "This is not worth watching even once.", 
             "This one is a pass."]

print("Untrained model predictions:")
print("----------------------------")
for text in text_list:
    # tokenize text
    inputs = tokenizer.encode(text, return_tensors="pt")
    # compute logits
    logits = model(inputs).logits
    # convert logits to label
    predictions = torch.argmax(logits)

    print(text + " - " + id2label[predictions.tolist()])
```

```
Untrained model predictions:
----------------------------
It was good. - Negative
Not a fan, don't recommed. - Negative
Better than the first one. - Negative
This is not worth watching even once. - Negative
This one is a pass. - Negative
```

PEFT/LoRA Setup



Train Model with PEFT/LoRA

```
[11]: peft_config = LoraConfig(task_type="SEQ_CLS",
                               r=4,
                               lora_alpha=32,
                               lora_dropout=0.01,
                               target_modules = ['q_lin'])
```

```
[12]: peft_config
```

```
...
```

```
[13]: model = get_peft_model(model, peft_config)
      model.print_trainable_parameters()
```

```
trainable params: 628,994 || all params: 67,584,004 || trainable%: 0.9307
```

Fine-Tune

- Trains model using LoRA
- Accuracy metric is used to show progress
- Took 7:35 minutes

```
In [32]: # train model
         trainer.train()
```

[2500/2500 07:35, Epoch 10/10]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.503386	{'accuracy': 0.861}
2	0.455300	0.356936	{'accuracy': 0.885}
3	0.455300	0.593905	{'accuracy': 0.881}
4	0.201400	0.576699	{'accuracy': 0.901}
5	0.201400	0.642713	{'accuracy': 0.897}
6	0.064700	0.739768	{'accuracy': 0.891}
7	0.064700	0.876202	{'accuracy': 0.889}
8	0.014900	0.897519	{'accuracy': 0.89}
9	0.014900	0.887371	{'accuracy': 0.894}
10	0.008800	0.902315	{'accuracy': 0.893}

Test Fine-Tuned Model

- Fine-Tuned Model is used to display predictions
- 60% ➡️ 100% accuracy

Generate Prediction

```
In [38]: print("Trained model predictions:")
         print("--------------------------")
         for text in text_list:
             inputs = tokenizer.encode(text, return_tensors="pt")

             logits = model(inputs).logits
             predictions = torch.max(logits,1).indices

             print(text + " - " + id2label[predictions.tolist()[0]])
```

```
Trained model predictions:
--------------------------
It was good. - Positive
Not a fan, don't recommed. - Negative
Better than the first one. - Positive
This is not worth watching even once. - Negative
This one is a pass. - Negative
```

QUESTIONS?