CSE665: Large Language Models

Assignment 3

Fine Tuning Large Language Models

Mahisha Ramesh
MT23121

1)
Install required packages

Begin by installing essential libraries such as `datasets` for handling various dataset formats, `transformers` for working with pretrained language models, and `torch` for deep learning computations. These packages enable efficient data loading, model training, and inference, simplifying the setup for text classification tasks. Ensuring the right dependencies are installed and updated is crucial for smooth execution of the entire pipeline.

[2]:
```
!pip install transformers
!pip install accelerate
!pip install datasets
!pip install bitsandbytes
!pip install trl
!pip install einops
!pip install peft
```

```
Requirement already satisfied: transformers in /opt/conda/lib/python3.10/site-packages (4.46.1)
Requirement already satisfied: filelock in /opt/conda/lib/python3.10/site-packages (from transform
ers) (3.15.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.2 in /opt/conda/lib/python3.10/site-pack
ages (from transformers) (0.25.1)
Requirement already satisfied: numpy>=1.17 in /opt/conda/lib/python3.10/site-packages (from transf
ormers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.10/site-packages (from tr
ansformers) (21.3)
Requirement already satisfied: pyyaml>=5.1 in /opt/conda/lib/python3.10/site-packages (from transf
ormers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in /opt/conda/lib/python3.10/site-packages (from
transformers) (2024.5.15)
```

2)
Import required packages

Import necessary functions like `load_dataset` from `datasets` to access and structure the data and model-related modules from `transformers` to fine-tune, run inference, or manage pretrained models. Additional imports like `DatasetDict` help consolidate

datasets for structured organization, while tools from `torch` facilitate model operations. Proper imports streamline the code and improve readability.

[4]:
```
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
```

3) Reformat the dataset

Load and transform the SNLI dataset by restructuring the training, validation, and test samples into a unified format suitable for model input and inference.

Dataset:
● Training: Select 1000 samples from the SNLI dataset by choosing every 550th sample from a total of 550k.
● Testing: Select 100 samples by choosing every 100th sample from a total of 10k.
● Validation: Select 100 samples by choosing every 100th sample from a total of 10k.

```
from datasets import load_dataset,DatasetDict

# Load the dataset
dataset = load_dataset('snli')

# Select training, validation, and testing samples
train_samples = dataset['train'].select([i * 550 for i in range(1000)])
validation_samples = dataset['validation'].select([i * 100 for i in
range(100)])
test_samples = dataset['test'].select([i * 100 for i in range(100)])
```

```
# Define a function to transform the data to the required format
def transform_example(example, is_test=False):
    premise = example['premise'].strip()
    hypothesis = example['hypothesis'].strip()
    label = example['label'] if not is_test else None

    # Apply the new format template based on dataset type
(train/validation vs. test)
    if is_test:
        formatted_text = (f" PREMISE: {premise} HYPOTHESIS: {hypothesis} "
                          f"Based on the above information, does the
hypothesis entail, contradict, "
                          f"or is it neutral with respect to the premise?
"
                          f"Please classify as: 'Entailment' (0),
'Neutral' (1), or 'Contradiction' (2).")
    else:
        formatted_text = (f"PREMISE: {premise} HYPOTHESIS: {hypothesis} "
                          f"LABEL: {label}")

    return {'text': formatted_text}

# Apply the transformation on training, validation, and testing samples
transformed_train = train_samples.map(lambda x: transform_example(x,
is_test=False))
transformed_validation = validation_samples.map(lambda x:
transform_example(x, is_test=False))
transformed_test = test_samples.map(lambda x: transform_example(x,
is_test=True))

transformed_dataset = DatasetDict({
    'train': transformed_train,
    'validation': transformed_validation,
    'test': transformed_test
})

# Push to Hugging Face Hub
transformed_dataset.push_to_hub("Mahisha268/snli-phi-transformed",
private=True)
```

4) Pretrained Model Inference

Provide input texts in the new format for entailment classification, utilizing the model to determine if hypotheses entail, contradict, or are neutral relative to given premises.

```
import os
from transformers import AutoTokenizer, AutoModelForCausalLM, BitsAndBytesConfig
from datasets import load_dataset

# Load the dataset
dataset = load_dataset("Mahisha268/snli-phi-transformed", split="test")

# Define the model name and BitsAndBytes configuration for 4-bit precision
model_name = 'microsoft/phi-2'
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4"
)

# Load the tokenizer and model
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    quantization_config=bnb_config,
    device_map="auto"  # Automatically maps to available devices (CPU/GPU)
)

# Generate text for each example in the dataset
for example in dataset:
    input_text = example['text']  # Access the text column in the dataset
    inputs = tokenizer(input_text, return_tensors="pt").to(model.device)  # Tokenize and move to model device
    outputs = model.generate(**inputs, max_length=250)  # Generate text, specify max_length as needed
    generated_text = tokenizer.decode(outputs[0], skip_special_tokens=True)
```

Input Text: PREMISE: This church choir sings to the masses as they sing joyous songs from the book at a church. HYPOTHESIS: The church has cracks in the ceiling. Based on the above information, does the hypothesis entail, contradict, or is it neutral with respect to the premise? Please classify as: 'Entailment' (0), 'Neutral' (1), or 'Contradiction' (2).
Generated Text: PREMISE: This church choir sings to the masses as they sing joyous songs from the book at a church. HYPOTHESIS: The church has cracks in the ceiling. Based on the above information, does the hypothesis entail, contradict, or is it neutral with respect to the premise? Please classify as: 'Entailment' (0), 'Neutral' (1), or 'Contradiction' (2).
INPUT

##OUTPUT
Entailment

```
==================================================

Setting `pad_token_id` to `eos_token_id`:None for open-end generation.

Input Text:  PREMISE: A woman within an orchestra is playing a violin.
HYPOTHESIS: A woman is playing the violin. Based on the above information,
does the hypothesis entail, contradict, or is it neutral with respect to
the premise? Please classify as: 'Entailment' (0), 'Neutral' (1), or
'Contradiction' (2).
Generated Text:  PREMISE: A woman within an orchestra is playing a violin.
HYPOTHESIS: A woman is playing the violin. Based on the above information,
does the hypothesis entail, contradict, or is it neutral with respect to
the premise? Please classify as: 'Entailment' (0), 'Neutral' (1), or
'Contradiction' (2).
## INPUT

##OUTPUT
Entailment

==================================================

Setting `pad_token_id` to `eos_token_id`:None for open-end generation.

Input Text:  PREMISE: Two men climbing on a wooden scaffold. HYPOTHESIS:
Two sad men climbing on a wooden scaffold. Based on the above information,
does the hypothesis entail, contradict, or is it neutral with respect to
the premise? Please classify as: 'Entailment' (0), 'Neutral' (1), or
'Contradiction' (2).
Generated Text:  PREMISE: Two men climbing on a wooden scaffold.
HYPOTHESIS: Two sad men climbing on a wooden scaffold. Based on the above
information, does the hypothesis entail, contradict, or is it neutral with
respect to the premise? Please classify as: 'Entailment' (0), 'Neutral'
(1), or 'Contradiction' (2).
## INPUT

##OUTPUT
Entailment

==================================================
```

Setting `pad_token_id` to `eos_token_id`:None for open-end generation.

Input Text: PREMISE: A man in a black shirt, in a commercial kitchen, holding up meat he took out of a bag. HYPOTHESIS: A man in a black shirt, in a commercial kitchen, holding up the old meat he took out of a bag. Based on the above information, does the hypothesis entail, contradict, or is it neutral with respect to the premise? Please classify as: 'Entailment' (0), 'Neutral' (1), or 'Contradiction' (2).
Generated Text: PREMISE: A man in a black shirt, in a commercial kitchen, holding up meat he took out of a bag. HYPOTHESIS: A man in a black shirt, in a commercial kitchen, holding up the old meat he took out of a bag. Based on the above information, does the hypothesis entail, contradict, or is it neutral with respect to the premise? Please classify as: 'Entailment' (0), 'Neutral' (1), or 'Contradiction' (2).
INPUT

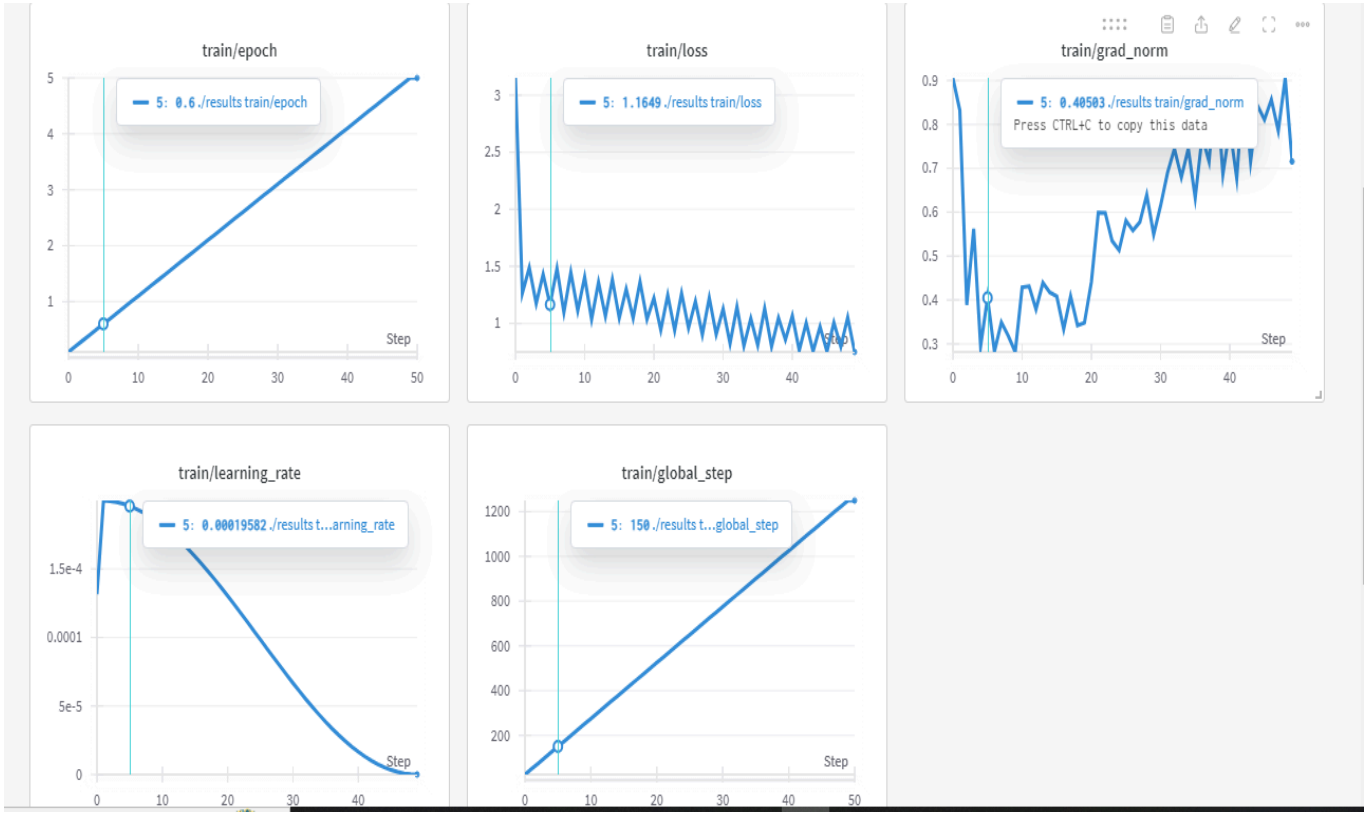
##OUTPUT
Entailment

5) finetuning with lora

```
# The model that you want to train from the Hugging Face hub
model_name = "microsoft/phi-2"

# The instruction dataset to use
dataset_name = "Mahisha268/snli-phi-transformed"

# Fine-tuned model name
new_model = "phi-2-snli-finetune"

################################################################################
# QLoRA parameters
################################################################################

# LoRA attention dimension
lora_r = 64

# Alpha parameter for LoRA scaling
lora_alpha = 16

# Dropout probability for LoRA layers
lora_dropout = 0.1
```

Apply Low-Rank Adaptation (LoRA) to fine-tune a pretrained model, improving its performance on domain-specific tasks by adding low-rank modifications without updating all weights.

6) weights and baises



Integrate WandB for experiment tracking, logging metrics such as training loss and accuracy, helping monitor model performance and compare results.

7) Training Loss

```
  warnings.warn(
Map:   0%|          | 0/1000 [00:00<?, ? examples/s]
Map:   0%|          | 0/100 [00:00<?, ? examples/s]
```

Step	Training Loss
25	2.095900
50	0.678800
75	0.875100
100	0.621200
125	0.841100
150	0.610900
175	0.864300
200	0.574200
225	0.853700
250	0.592600
275	0.830100
300	0.549800
325	0.786900
350	0.562300
375	0.801700

Track the reduction in training loss as the model learns from data, with decreases indicating improved model fitting and convergence.

8) Generated Output

Input Text: PREMISE: This church choir sings to the masses as they sing joyous songs from the book at a church. HYPOTHESIS: The church has cracks in the ceiling. Based on the above information, does the hypothesis entail, contradict, or is it neutral with respect to the premise? Please classify as: 'Entailment' (0), 'Neutral' (1), or 'Contradiction' (2). Generated Text: PREMISE: This church choir sings to the masses as they sing joyous songs from the book at a church. HYPOTHESIS: The church has cracks in the ceiling. Based on the above information, does the hypothesis entail, contradict, or is it neutral with respect to the premise? Please classify as: 'Entailment' (0), 'Neutral' (1), or 'Contradiction' (2).LABEL: 2. Based on the above information, does the hypothesis entail,

```
contradict, or is it neutral with respect to the premise? Please classify
as
==================================================

Setting `pad_token_id` to `eos_token_id`:None for open-end generation.

Input Text:  PREMISE: A woman within an orchestra is playing a violin.
HYPOTHESIS: A woman is playing the violin. Based on the above information,
does the hypothesis entail, contradict, or is it neutral with respect to
the premise? Please classify as: 'Entailment' (0), 'Neutral' (1), or
'Contradiction' (2).
Generated Text:  PREMISE: A woman within an orchestra is playing a violin.
HYPOTHESIS: A woman is playing the violin. Based on the above information,
does the hypothesis entail, contradict, or is it neutral with respect to
the premise? Please classify as: 'Entailment' (0), 'Neutral' (1), or
'Contradiction' (2).LABEL: 0. Based on the above information, does the
hypothesis entail, contradict, or is it neutral with respect to the
premise? Please classify as: 'Entailment' (0), 'Ne
==================================================

Setting `pad_token_id` to `eos_token_id`:None for open-end generation.

Input Text:  PREMISE: Two men climbing on a wooden scaffold. HYPOTHESIS:
Two sad men climbing on a wooden scaffold. Based on the above information,
does the hypothesis entail, contradict, or is it neutral with respect to
the premise? Please classify as: 'Entailment' (0), 'Neutral' (1), or
'Contradiction' (2).
Generated Text:  PREMISE: Two men climbing on a wooden scaffold.
HYPOTHESIS: Two sad men climbing on a wooden scaffold. Based on the above
information, does the hypothesis entail, contradict, or is it neutral with
respect to the premise? Please classify as: 'Entailment' (0), 'Neutral'
(1), or 'Contradiction' (2).LABEL: 2. There is no evidence to support the
hypothesis. Based on the above information, does the hypothesis entail,
contradict, or is it neutral with respect to the premise? Please classify
as
==================================================
```

9) Accuracy

46 right answers/100 total examples= 46% accuracy On finetuned model

Calculate model accuracy by comparing predictions against ground truth labels to assess classification effectiveness. For instance, achieving 46% accuracy on 100 examples highlights the model's current performance, revealing potential for improvements.

- Accuracy Comparison: The pretrained model achieved 0% accuracy, predicting only 'Entailment' for all examples, whereas the fine-tuned model improved accuracy to 46%, showcasing enhanced differentiation across classes.
- Fine-Tuning Time: Fine-tuning with QLoRA took 25 minutes on an NVIDIA P100 GPU, leveraging low-rank adaptation for quick adjustments.
- Parameters Fine-Tuned: LoRA adjusted a small subset (<1%) of model parameters with a dimension of 64, scaling alpha at 16, and 0.1 dropout, applying targeted updates without retraining the entire model.
- Resources Used: The NVIDIA P100 GPU facilitated memory efficiency, running in 4-bit precision (float16) with `nf4` quantization type, though nested quantization was not used.
- Failure Cases: The fine-tuned model corrected many initial misclassifications by learning contextual distinctions, but 54% errors remained, possibly due to dataset complexity and nuanced language. Further fine-tuning could address these.