IMDB Sentiment Analysis

Low-Rank Adaptation (LoRA Fine-Tuning)

CS614 – GenAI with LLM
Yue Jun Yuan

Objective

- Investigate if PEFT with LORA can significantly improve the performance of a lightweight pretrained model for a text classification task
- Obtain baseline performance of a pretrained LLM on classifying text reviews
- Perform PEFT with LoRA and observe any improvements in performance

Dataset

```
DatasetDict({
    train: Dataset({
        features: ['text', 'label'],
        num_rows: 25000
    })
```

Example “text”:

Going into the movie with the right expectations, I somewhat liked this movie.
Like most reviewers who have seen this movie, I fully agree that the plot was razor thin,
clichéd, and I could predict every plot twist from the very beginning of the movie.
But, the dancing sequences were VERY well done, and I really enjoyed the fusion of classical
and hip-hop dance (both which I enjoy watching). The music/soundtrack of the movie was also
very good, which made the "drama" scenes more bearable. The leads (Jenna Dewan and Channing Tatum)
were OK as actors, but their dancing throughout the movie was impressive and mesmerizing.<br /><br />
All in all, a movie worth watching if you like to watch good dance sequences, and this movie is
MUCH MUCH MUCH better than "You Got Served" in terms of the plot and drama.
Then again, that doesn't say much, does it? =P

Dataset

```
DatasetDict({
    train: Dataset({
        features: ['text', 'label'],
        num_rows: 25000
    })
```

Example “text”:

Going into the movie with the right expectations, I somewhat liked this movie.
Like most reviewers who have seen this movie, I fully agree that the plot was razor thin,
clichéd, and I could predict every plot twist from the very beginning of the movie.
But, the dancing sequences were VERY well done, and I really enjoyed the fusion of classical
and hip-hop dance (both which I enjoy watching). The music/soundtrack of the movie was also
very good, which made the "drama" scenes more bearable. The leads (Jenna Dewan and Channing Tatum)
were OK as actors, but their dancing throughout the movie was impressive and mesmerizing.<br /><br />
All in all, a movie worth watching if you like to watch good dance sequences, and this movie is
MUCH MUCH MUCH better than "You Got Served" in terms of the plot and drama.
Then again, that doesn't say much, does it? =P

“label”: 0

Not easy to predict

Dataset / Training Preparation

1. Instantiate “bert-base-uncased” model
   - ❖ Total parameters: 109,483,778

2. Tokenize dataset (based on vocabulary of “bert-base-uncased”
   - ❖ Adds the 'input_ids', 'token_type_ids', 'attention_mask’ columns

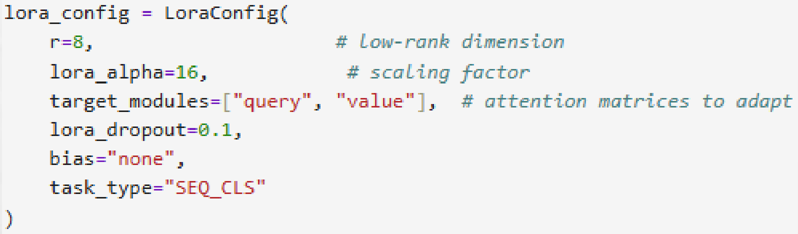
3. Define train and test splits
   - ❖ 5000 records for train and 2000 records for test

4. Define HF Trainer class

5. Generate baseline pre-LoRA results – trainer.evaluate()

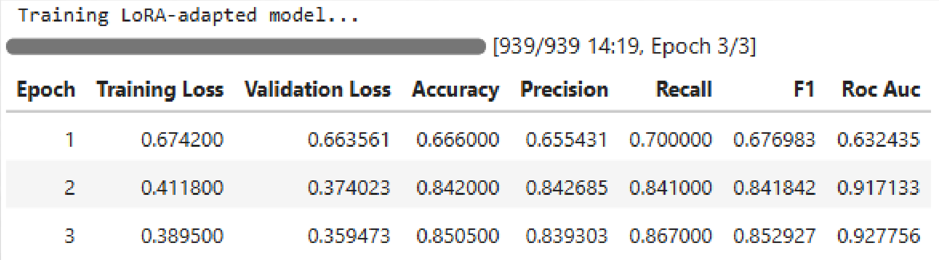
Dataset / Training Preparation

1. **Define LoRA configuration**



```
lora_config = LoraConfig(
    r=8,                       # Low-rank dimension
    lora_alpha=16,              # scaling factor
    target_modules=["query", "value"],  # attention matrices to adapt
    lora_dropout=0.1,
    bias="none",
    task_type="SEQ_CLS"
)
```

2. **trainer.train()**



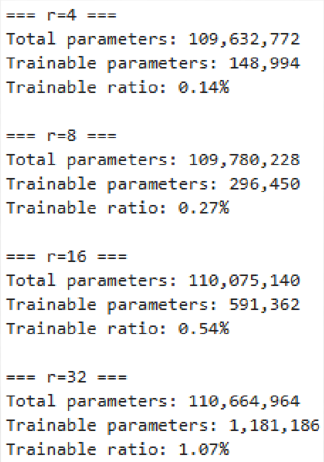
```
Training LoRA-adapted model...
[939/939 14:19, Epoch 3/3]
```

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1	Roc Auc
1	0.674200	0.663561	0.666000	0.655431	0.700000	0.676983	0.632435
2	0.411800	0.374023	0.842000	0.842685	0.841000	0.841842	0.917133
3	0.389500	0.359473	0.850500	0.839303	0.867000	0.852927	0.927756

3. **trainer.evaluate()**

W'=W+ α/r BA

- **r: rank of low-rank matrices A and B**
  - **Smaller r: less adaptation capacity, faster training**
  - **Larger r: more expressive, higher compute cost**
- **lora_alpha α : scaling factor**
  - **Higher α : stronger influence for lora adapters**
- **target_modules: specify which modules to target and insert low rank adapters – typically W^q and W^v**



```
=== r=4 ===
Total parameters: 109,632,772
Trainable parameters: 148,994
Trainable ratio: 0.14%

=== r=8 ===
Total parameters: 109,780,228
Trainable parameters: 296,450
Trainable ratio: 0.27%

=== r=16 ===
Total parameters: 110,075,140
Trainable parameters: 591,362
Trainable ratio: 0.54%

=== r=32 ===
Total parameters: 110,664,964
Trainable parameters: 1,181,186
Trainable ratio: 1.07%
```

Evaluation Comparison

Metrics	Baseline (Pre-LoRA)	Post LoRA fine-tuning
Eval_loss	0.726	0.359
Eval_accuracy	0.492	0.851
Eval_precision	0.432	0.839
Eval_recall	0.051	0.867
Eval_F1	0.091	0.853
Eval_ROC_AUC	0.344	0.928