# HW7

irene0613

June 26, 2024

## 1 Introduction

This assignment involves using LoRA (Low-Rank Adaptation) and other Parameter Efficient Fine-Tuning (PEFT) methods to fine-tune a DistilBERT model for sequence classification on the IMDB dataset. The objective is to experiment with different method configurations and observe their results.

Detailed documentation is provided in the README.md file and the Transformers_peft.ipynb notebook, along with screenshots of the training process conducted on Google Colab.

## 2 Training Method

### 2.1 Model and Tokenizer

We utilize **distilbert-base-cased** as our foundational model and ensure consistent initialization of the tokenizer from the model checkpoint.

### 2.2 Data Preparation

- Load the IMDB movie review dataset using the **datasets** library. The dataset contains 50,000 movie reviews labeled as positive or negative.
- Create a small-scale training and validation dataset from the IMDB dataset (ratio 4:1) and truncate each review to the first 50 tokens.
- Tokenize the newly created small-scale training and validation datasets in batches, processing 16 samples at a time.

### 2.3 Training Parameters

- Batch Size: During each iteration, the model calculates the loss for this batch of data and updates the weights based on this loss. The choice of batch size affects training speed and model performance.
- Learning Rate: Controls the step size of model weight updates. It is a crucial parameter that influences the convergence speed and effectiveness during the training process.
- Evaluation Strategy: Ensures that the model does not overfit or underfit during training and provides immediate feedback to adjust training parameters. In this assignment, evaluation is done per epoch—after training one epoch, an evaluation is performed.
- Number of Epochs: Controls the number of times the model trains on the training dataset, ensuring the model has enough opportunities to learn patterns in the data.

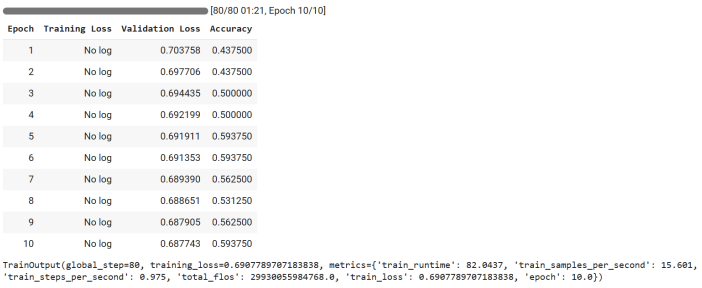
### 2.4 Trainer

The Hugging Face Trainer manages the training process and is responsible for model initialization, training parameter settings, setup of training and validation datasets, tokenizer configuration, and custom evaluation metrics.

## 2.5 LoRA & IA3 Configurations and Their Corresponding Results

- LoraConfig_1

```
peft_config_1 = LoraConfig(
    lora_alpha=16,
    lora_dropout=0.1,
    r=64,
    bias="none",
    task_type="SEQ_CLS",
    target_modules=["q_lin", "v_lin", "k_lin", "out_lin"],
)
```



[80/80 01:21, Epoch 10/10]

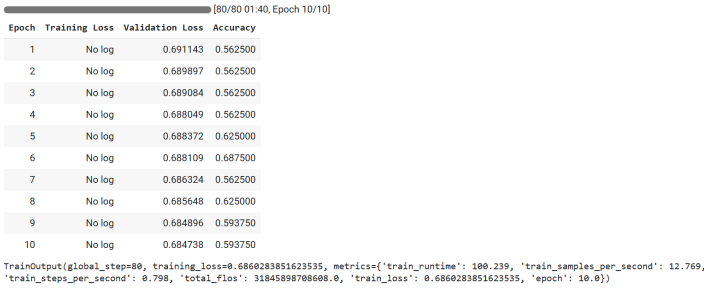
| Epoch | Training Loss | Validation Loss | Accuracy |
|---|---|---|---|
| 1 | No log | 0.703758 | 0.437500 |
| 2 | No log | 0.697706 | 0.437500 |
| 3 | No log | 0.694435 | 0.500000 |
| 4 | No log | 0.692199 | 0.500000 |
| 5 | No log | 0.691911 | 0.593750 |
| 6 | No log | 0.691353 | 0.593750 |
| 7 | No log | 0.689390 | 0.562500 |
| 8 | No log | 0.688651 | 0.531250 |
| 9 | No log | 0.687905 | 0.562500 |
| 10 | No log | 0.687743 | 0.593750 |

TrainOutput(global_step=80, training_loss=0.6907789707183838, metrics={'train_runtime': 82.0437, 'train_samples_per_second': 15.601, 'train_steps_per_second': 0.975, 'total_flos': 29930055984768.0, 'train_loss': 0.6907789707183838, 'epoch': 10.0})

Figure 1: Training result LoraConfig_1

- LoraConfig_2

```
peft_config_2 = LoraConfig(
    lora_alpha=16,
    lora_dropout=0.1,
    r=64,
    bias="none",
    task_type="SEQ_CLS",
    target_modules=["q_lin", "v_lin", "k_lin", "out_lin", "ffn.lin1", "ffn.lin2"],
)
```



[80/80 01:40, Epoch 10/10]

| Epoch | Training Loss | Validation Loss | Accuracy |
|---|---|---|---|
| 1 | No log | 0.691143 | 0.562500 |
| 2 | No log | 0.689897 | 0.562500 |
| 3 | No log | 0.689084 | 0.562500 |
| 4 | No log | 0.688049 | 0.562500 |
| 5 | No log | 0.688372 | 0.625000 |
| 6 | No log | 0.688109 | 0.687500 |
| 7 | No log | 0.686324 | 0.562500 |
| 8 | No log | 0.685648 | 0.625000 |
| 9 | No log | 0.684896 | 0.593750 |
| 10 | No log | 0.684738 | 0.593750 |

TrainOutput(global_step=80, training_loss=0.6860283851623535, metrics={'train_runtime': 100.239, 'train_samples_per_second': 12.769, 'train_steps_per_second': 0.798, 'total_flos': 31845898708608.0, 'train_loss': 0.6860283851623535, 'epoch': 10.0})

Figure 2: Training result LoraConfig_2

- LoraConfig_3

```
peft_config_3 = LoraConfig(
    lora_alpha=32,
    lora_dropout=0.2,
    r=32,
    bias="none",
    task_type="SEQ_CLS",
    target_modules=["q_lin", "v_lin", "k_lin", "out_lin"],
)
```


| Epoch | Training Loss | Validation Loss | Accuracy |
|---|---|---|---|
| 1 | No log | 0.697127 | 0.437500 |
| 2 | No log | 0.693607 | 0.468750 |
| 3 | No log | 0.691058 | 0.531250 |
| 4 | No log | 0.689418 | 0.593750 |
| 5 | No log | 0.689410 | 0.531250 |
| 6 | No log | 0.688931 | 0.562500 |
| 7 | No log | 0.687193 | 0.687500 |
| 8 | No log | 0.686551 | 0.625000 |
| 9 | No log | 0.685798 | 0.593750 |
| 10 | No log | 0.685638 | 0.593750 |

TrainOutput(global_step=80, training_loss=0.6866413593292237, metrics={'train_runtime': 80.8688, 'train_samples_per_second': 15.828, 'train_steps_per_second': 0.989, 'total_flos': 29163718895232.0, 'train_loss': 0.6866413593292237, 'epoch': 10.0})

Figure 3: Training result LoraConfig_3

- IA3Config

```
peft_config_4 = IA3Config(
    task_type=TaskType.SEQ_CLS,
    target_modules=["q_lin", "v_lin", "k_lin", "out_lin", "ffn.lin1", "ffn.lin2"],
    feedforward_modules=["ffn.lin1", "ffn.lin2"],
)
```


| Epoch | Training Loss | Validation Loss | Accuracy |
|---|---|---|---|
| 1 | No log | 0.691732 | 0.562500 |
| 2 | No log | 0.691682 | 0.562500 |
| 3 | No log | 0.691486 | 0.562500 |
| 4 | No log | 0.691044 | 0.562500 |
| 5 | No log | 0.691567 | 0.625000 |
| 6 | No log | 0.691581 | 0.531250 |
| 7 | No log | 0.690272 | 0.625000 |
| 8 | No log | 0.689837 | 0.625000 |
| 9 | No log | 0.689332 | 0.625000 |
| 10 | No log | 0.689244 | 0.625000 |

TrainOutput(global_step=80, training_loss=0.6946804046630859, metrics={'train_runtime': 268.5065, 'train_samples_per_second': 4.767, 'train_steps_per_second': 0.298, 'total_flos': 28424323344000.0, 'train_loss': 0.6946804046630859, 'epoch': 10.0})

Figure 4: Training result IA3Config