GAI Project 3 Report

PEFT on GLUE benchmarks

資訊 114 F7401254 張暐俊

1. Model analysis

本次使用的模型是 google bert 底下的 bert-base-uncased，以下為導入之模型、tokenizer 與 data_collator。

```
checkpoint = "google-bert/bert-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(checkpoint)
```

```
data_collator = DataCollatorWithPadding(tokenizer=tokenizer)
```

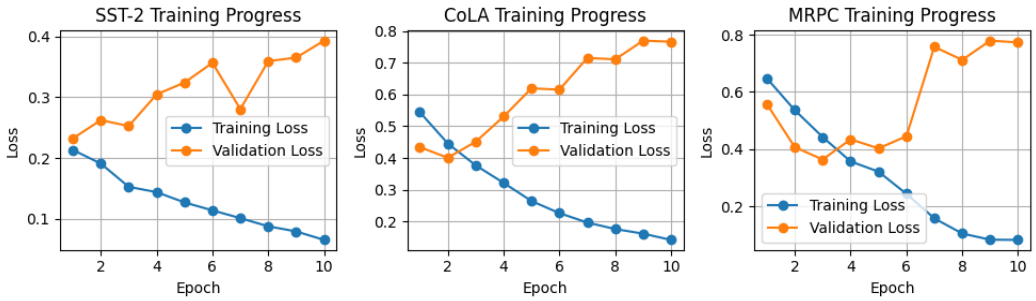
```
{datasets_name}_model = AutoModelForSequenceClassification.from_pretrained(checkpoint, num_labels=2)
```

以下的結果是使用 Lora 的方式 train 出來的結果，下表為各種 datasets 使用之超參數。

TrainingArguments	SST2	CoLA	MRPC
num_train_epochs	10		
learning_rate	5e-4		
per_device_train_batch_size	16		
per_device_eval_batch_size	16		
gradient_accumulation_steps	1		
warmup_steps	500		
weight_decay	0.01		
evaluation_strategy	Epoch		
save_strategy	Epoch		
save_total_limit	10		
seed	42		

LoraConfig	SST2	CoLA	MRPC
r	8		
lora_alpha	16		
lora_dropout	0.01		
bias	none		
task_type	SEQ_CLS		

下圖為三種 dataset 的 train losing rate 跟 validation losing rate 結果。



2. PEFT Discussion

A. Bitfit 最主要的重點是他在訓練時只更新 bias 的參數，因此在這種訓練的方式下，可以大幅度地縮小需要訓練的 parameter，同時因為是訓練 bias 的參數，也能在更新極少量參數的情況達到不錯的效果。

```
# Freeze all parameters except biases
for name, param in model.named_parameters():
    if 'bias' not in name:
        param.requires_grad = False
```

下表為我能找到對於我訓練使用的三種資料集來說，最好的超參數設定。主要在調整 learning rate 的部分著手。SST-2 及 MRPC 皆相較於 CoLA 來說更輕鬆就達到網路上大多資料引用的 paper 所達到的標準，而 CoLA 則是我更改了無數次超參數後，最後發現還是只調整

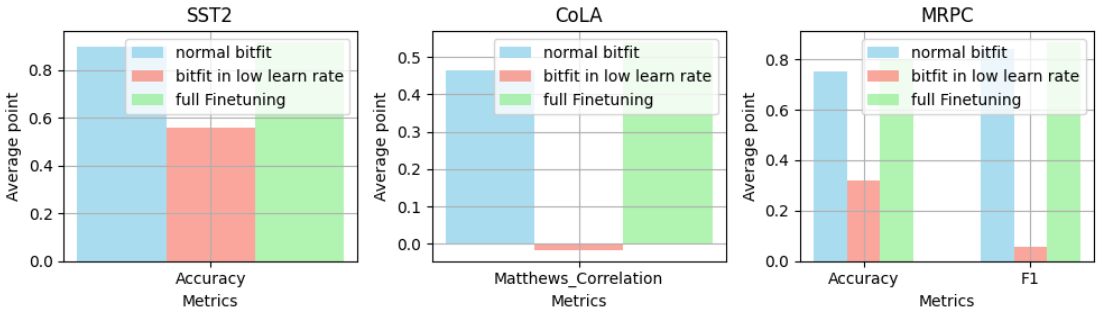
learning rate 有相對較好的結果。

TrainingArguments	SST2	CoLA	MRPC
num_train_epochs	5		
learning_rate	5e-4	1e-3	5e-4
per_device_train_batch_size	16		
per_device_eval_batch_size	16		
gradient_accumulation_steps	1		
warmup_steps	500		
weight_decay	0.01		
evaluation_strategy	Epoch		
save_strategy	Epoch		
save_total_limit	10		
seed	42		

B. 而相對於 full-finetunig 來說，bitfit 需要用相對較大的 learnig rate 來做訓練，因為 bitfit 模型相對來說較小，需要較大的 learning rate 協助。

TrainingArguments (full-finetuning)	SST2	CoLA	MRPC
learning_rate	1e-5	1e-5	1e-5

下圖結果為三個情況下的比較，藍色為使用上方表格，也就是在正常情況下的 bitfit，其實可以發現僅相對 full-finetune 的情況分數來的相對較低一些，但是當在 bitfit 使用與 full-finetuning 一樣相對較低的 learning rate 時(下圖紅色)，訓練結果就十分的糟糕。

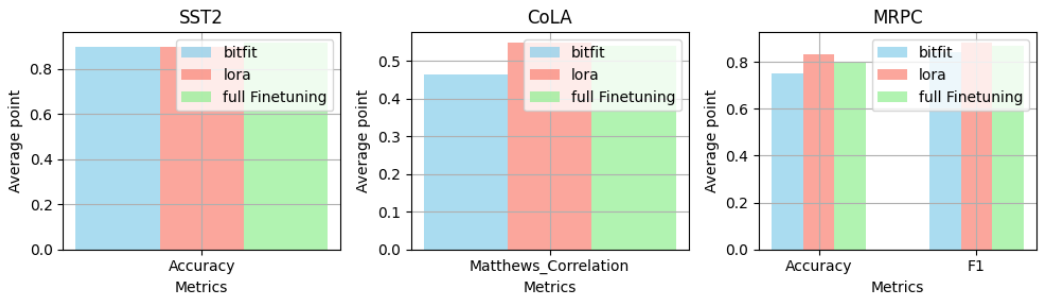


3. PEFT Comparison

A. 下圖為 bitfit、lora 與 full-finetunig 三者的評測結果比較圖以及使用之超參數。

TrainingArguments	SST2	CoLA	MRPC
learning_rate (full-finetuning)	1e-5	1e-5	1e-5
learning_rate (bitfit & lora)	5e-4	1e-3	5e-4
num_train_epochs	5		
per_device_train_batch_size	16		
per_device_eval_batch_size	16		
gradient_accumulation_steps	1		
warmup_steps	500		
weight_decay	0.01		
evaluation_strategy	Epoch		
save_strategy	Epoch		
save_total_limit	10		
seed	42		

LoraConfig	SST2	CoLA	MRPC
r	8		
lora_alpha	16		
lora_dropout	0.01		
bias	none		
task_type	SEQ_CLS		



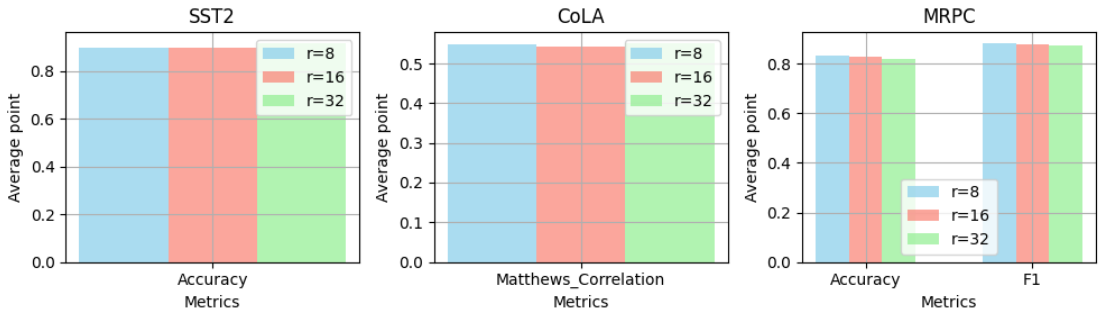
由上柱狀圖可以發現，相對而言，在使用 lora 模型會有相對較好的表現，且通常會相對 full-finetunig 的訓練節果來說還更好一些。而 bitfit 的訓練結果則是略遜於兩者。

B. 根據 lora 的架構設定以及實作後的結果可以發現 r 的不同會影響到能訓練的 parameter 的數量，但是對於訓練的結果而言，影響並沒有那麼的大，而且 r 的大小也並非是越大越好，或許對於模型以及欲訓練的資料集而言，都有最適合他們的 rank(r)。

下表為在不同 r 的情況下 parameter 數量的改變。

R=8	
trainable params	296,450
all params	109,780,228
trainable%	0.2700395193203643
R=16	
trainable params	591,362
all params	110,075,140
trainable%	0.537234837947969
R=32	
trainable params	1,181,186
all params	110,664,964
trainable%	1.0673531687951392

下圖為在不同 r 下，不同訓練集的評分結果。



下表為使用之超參數

TrainingArguments	SST2	CoLA	MRPC
learning_rate	5e-4	1e-3	5e-4
num_train_epochs	5		
per_device_train_batch_size	16		
per_device_eval_batch_size	16		
gradient_accumulation_steps	1		
warmup_steps	500		
weight_decay	0.01		
evaluation_strategy	Epoch		
save_strategy	Epoch		
save_total_limit	10		
seed	42		

LoraConfig	SST2	CoLA	MRPC
lora_alpha	16		
lora_dropout	0.01		
bias	none		
task_type	SEQ_CLS		