

運用工具

1. 運用助教的 code 加以撰寫 bert。

各參數值

```
# Input limitation
max_query_length = 64
max_input_length = 512
# 3~15
num_negatives = 3 # num. of negative documents to pair with a positive document
# num_negatives = 2

# Model finetuning
model_name_or_path = "bert-base-uncased" # 指定 英文的BERT-BASE 預訓練模型
# 2~5
# max_epochs = 4
max_epochs = 3
learning_rate = 2e-5
# learning_rate = 3e-5
dev_set_ratio = 0.2 # make a ratio of training set as development set for rescoring weight sniffing
max_patience = 0 # earlstop if avg. loss on development set doesn't decrease for num. of epochs
# batch_size = 4
batch_size = 2 # num. of inputs = 8 requires ~9200 MB VRAM (num. of inputs = batch_size * (num_negatives + 1))
num_workers = 2 # num. of jobs for pytorch dataloader
```

`alpha = 2.8`

`weighted_scores = bm25_scores + alpha * bert_scores`

1. 多數參數值皆參考助教釋出的，其中有改的只有將 max_epochs 改為 3，但在下面的實驗可看出其實每次都 early stop 在第 2 次，所以可得知 1 epochs 會使模型產生較大準確度，而 learning rate 改成 2e-5。
2. Alpha 值設為 2.8，該參數也是經由實驗測出此參數值是為最好的。

計算 back propagation

```
modeloutput = model(input_ids=input_ids,
                    token_type_ids=token_type_ids,
                    attention_mask=attention_mask,
                    labels=labels) #將data傳入model進行forward propagation

loss = modeloutput[0] #計算loss
total_loss += loss
loss.backward() # 根據loss進行back propagation，計算gradient
optimizer.step()

4
if i % 3 == 0: # 黑魔法來囉~
    optimizer.step() # 更新所有的参数
    optimizer.zero_grad() #清空前一次的gradient
```

1. 由上圖可看出，我每過 2 回合才回更新一次參數，若沒有加此行，而是照傳統的每經過 1 batch 就更新一次參數，在準確率上會相差 0.02~0.03 之準確度。

心得

其實這次作業如果助教沒有釋出 code，原本是有想放棄這次作業，因為實在太難懂了，但原因是因為自己本身除了對 nlp 完全沒概念外，對一些機器學習的基本概念也是「完全」沒學過，因此，其實就連光看助教的 code 都很吃力了，大概也花了 2 天理解，也上網聽了李弘毅老師的課程，才勉強理解部分程

式碼，剩下時間想辦法把 TO-DO 做出來。

一開始做出來的準確率大概 0.44 左右，就停滯在那邊，會突然想到不要每次更新參數，是由下面網址 1 的概念發想，他於內文所說「Pytorc 不幫你自動清空 gradient，這樣子可以有更大的彈性去做一些操作(黑魔法)，畢竟，誰規定每一次 iteration 都要清空 gradient」，因此，看到該句話後，我就想說試試看不要每次都清空，結果準確率就蠻大幅的提升的，但會造成準確度變高之可能的原因我還不清楚，可能需要再繼續研究。

總之，這學期上這堂課獲得的新知識真的很多，雖然後期有點吸收不良，導致後期很多作業都是盲改、運氣好才莫名其妙的做出來了，相較於其他人其實用到的技術都是最基本的，但也透過這學期的課堂知道了自己與其他人缺乏的能力在哪裡。

參考網址

1. <https://meetonfriday.com/posts/18392404/>
2. https://leemeng.tw/attack_on_bert_transfer_learning_in_nlp.html