|  |
| --- |
| 運用工具 |
| 1. 運用助教的code加以撰寫bert。 |
| 各參數值 |
|  |
|  |
| 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 |
|  |
| 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> |