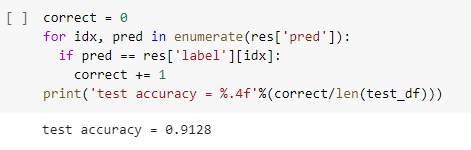
BERT

資管三A黃彥程 109403526

Accuracy:

0.9128



網址:

BERT 0.9128

<https://colab.research.google.com/drive/1J0meDYpPV0NuK5G2eU46uThUDjX1jPkO>

DistilBERT 0.6546

<https://colab.research.google.com/drive/17YlzcSjSNney9OgN6PeLddTSEwtaJ0ia>

這種BERT的變形比較小巧，可以在有限資源內取得更好的成效。

RoBERTa 0.5284

<https://colab.research.google.com/drive/1E-bwD0e58bVmcJmK6I6eSLaGPPwBGpIK>

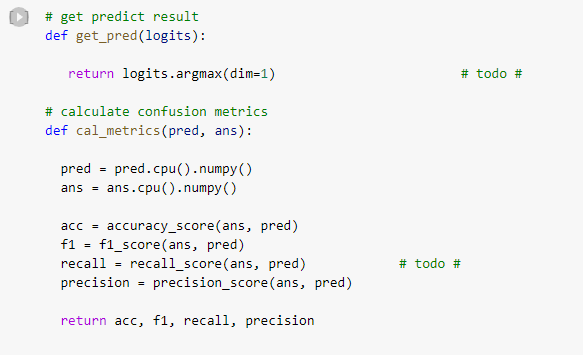
這種BERT變形比較厚實，可以比BERT有更好的效能，但所使用之模型較大，需要的資源較多。

過程:

Todo1&2

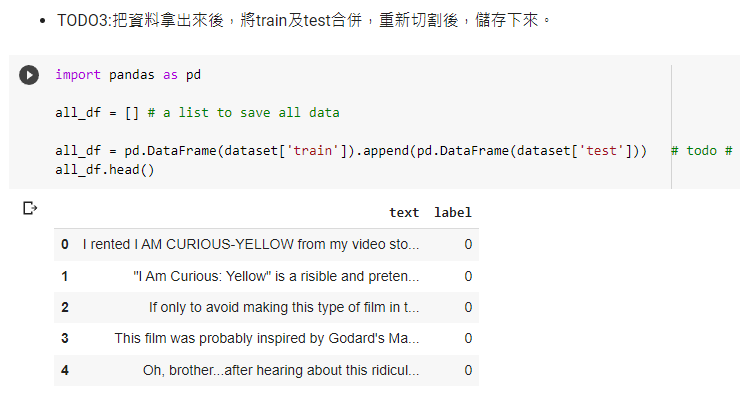
在做之前，雖然有看過助教提供的範本，但由於之前除了寫這堂課第一次作業之外，沒碰過機器學習程式碼，再說我多看幾次也覺得這次作業跟上次的可以說八竿子打不著，整個架構都不一樣，上次作業可以線性按照助教給的提示往下做(雖然也是查很久怎麼寫)，但這次作業我看到前兩個todo，頓時間不知道怎麼反應，硬是往下找這兩個函式會在哪裡用到，上下來回比對，才稍微比較了解，圖片作業中是我後來看到範本有一樣的動作所以就參考過來了。

Call\_metric 的部分則是寫到tokenizer跟evaluate時看到需要回傳那些參數(各種score)，才回頭去查然後寫出來。

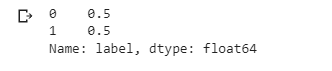


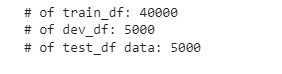
Todo 3

用append把他們連上



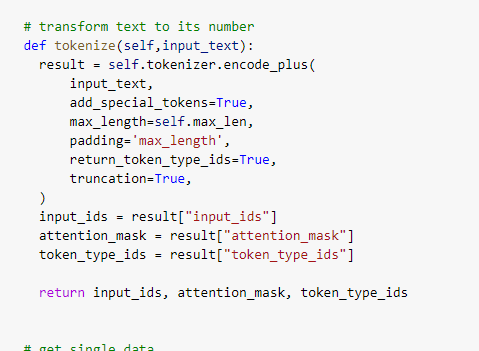
檢查是否正確





Todo 4

依bert-base-uncased設置好的tokenizer格式會回傳input\_ids, attention\_mask, token\_type\_ids。



Todo 5

按照助教給的提示依序加入，除了

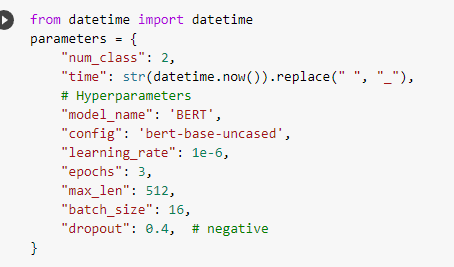
bt\_output = bert\_output.pooler\_output

這行是我在查資料時發現大家都有寫到的。



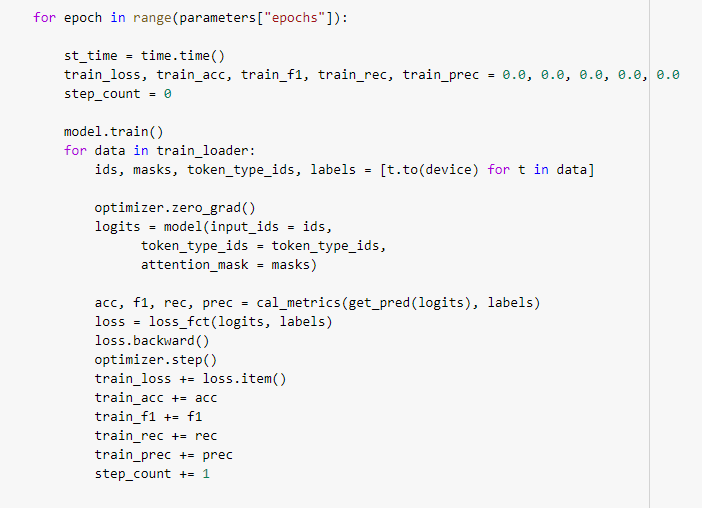
Hyperparameter

在train model時自己嘗試learning rate, len ，那裏寫了negative是我在嘗試時提醒自己就先別動dropout=0.4。

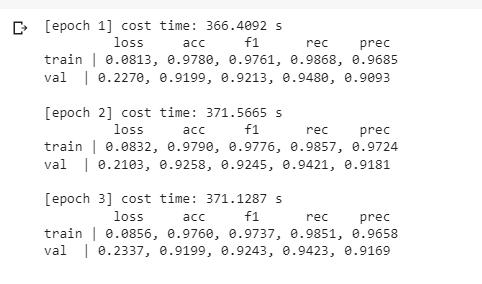


Todo 6

參照evaluate寫出訓練方式。

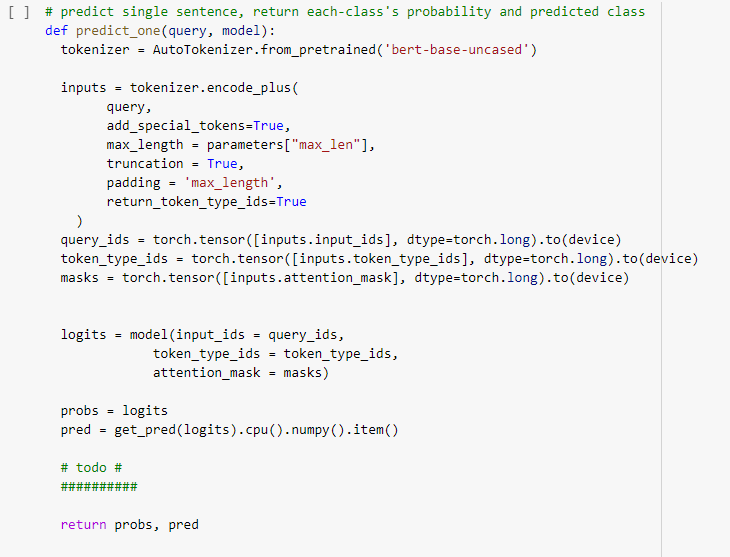


一段時間過去了↓



Todo 7

裡面的’bert-base-uncased’寫 parameter[‘config’]可能比較好，我忘記改了，不過也是一樣的東西不影響。



心得:

首先就是成就感，上次就覺得作業內容蠻有趣充實，這次雖然看上去比上次亂，個人感覺也比較難，寫之前心裡沒有一個底這作業大概長怎樣要怎麼寫，但不停查資料慢慢把內容寫出來，然後逐漸理解整個到底怎麼運作，還是很有趣很有成就感的，然後就是一些過程，那些東西都寫出來之後，就要開始調整參數(Hyperparameter)去train 寫好的model，剛開始概念都不清楚，參數可以說是在抽樂透，有幾次跑到80以上，但我又覺得不夠，多測幾次沒有意外果然是卡回50左右，就好像在說:「這模型跟我一樣，學不會」，後來想說不行，我還是去了解一下那些參數的影響，最後是試出了還可以的分數。

至於其他兩個模型，比較值得注意的是，DistilBERT這個模型的tokenizer是沒有回傳token\_type\_id的，我想是因為他這個模型本身的目的就是為了迎合當資源有限時能夠更有效的達到目標。

而RoBERTa我有發現到同樣的batch size他要使用的learning rate比較低，不然recall\_score跟f1\_score會一起0.0000，要到1e-7~1e-9才會比較有結果。

做完本次作業我有感覺到我數據整理的能力，跟對機器學習有顯著得提升，雖然我知道這些都還只是一些很淺的內容，不過確實有收穫!