學號：R05323040　系級：經濟所碩二　姓名：田家駿

1. (1%) 請說明你實作的 RNN model，其模型架構、訓練過程和準確率為何?
   (Collaborators: 葉政維，我有分享我作法的整個流程給他看)

答：我的 best model 使用的是 Gensim 的 word to vector 把每一個字組成一個一百維的向量，且每一句我設一個最大長度 40 個字，小於 40 個字後面就補零，因此 LSTM input 為（20000,40,100)，LSTM 因發現滿容易有 overfit 的問題，因此除了做 dropout 外，我還使用了 l2 的 regularization 讓 overfit 問題不這嚴重，上傳的準確率為我最好的成績 0.8208（public)

```
>>> model.summary()
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 40, 128)           117248
_________________________________________________________________
dropout_1 (Dropout)          (None, 40, 128)           0
_________________________________________________________________
lstm_2 (LSTM)                (None, 64)                49408
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                4160
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 65
=================================================================
Total params: 170,881
Trainable params: 170,881
Non-trainable params: 0
_________________________________________________________________
```

2. (1%) 請說明你實作的 BOW model，其模型架構、訓練過程和準確率為何?
   (Collaborators:)

答：在 BOW 我使用 gensim.corpora 的 Dictionary 實做，此外我使用把低於 60 次出現的字給濾掉讓沒一筆 feature 的維度不要太高，濾掉之後我的維度是 4691，接著在丟到疊了三層的 fully connected 中去 train，因這筆資量 insample 很容易就衝高，但 outsample 卻很難爬升，因五我沒一層還是去做 drop out 去控制 overfit , 上傳的成績為 0.79 (public)

3. (1%) 請比較 bag of word 與 RNN 兩種不同 model 對於"today is a good day, but it is hot"與"today is hot, but it is a good day"這兩句的情緒分數，並討論造成差異的原因。
   (Collaborators:)

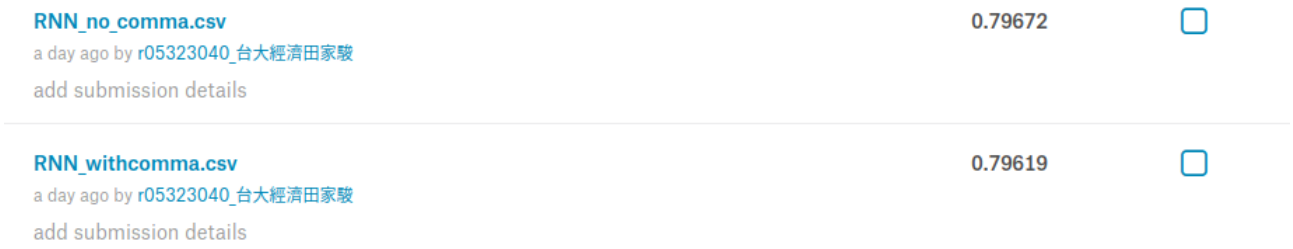
答：在 bag of word 中兩句預測出來的情緒複數皆為 0.731804，而 RNN 第一句分數為 0.8003

第二句則為 0.9625431 , 可以看到兩句在 bag of word 中並沒有差別，但在 RNN 中分數就有明顯差別 ，原因為 bag of word 本身用詞點字去排序就已經打亂了句子的先後順序，因此 bag of word 無法判別有順序的資訊，但 LSTM 可以抓到有序的資料。

4. (1%) 請比較"有無"包含標點符號兩種不同 tokenize 的方式，並討論兩者對準確率的影響。
   (Collaborators:)

答：這裡比較我使用的是助教 sample code 的模型，做完 token 之後在放到 embeded layer 幫我 tranform 成每一個字一百維的向量，在此題中我使用正規表達把下面得標點濾掉比較，

re.sub('[{,},\^,$,&,*,_,-,<,=,>,@,:;~`<>/.!~?#%]','', lines[i]) , 上傳成績可以看到濾掉標點符號之後準確率有提升了 一點 ，因其實有一些奇怪的符號本身並無意義，word to vector 之後也只是增加了一些不必要的資訊。



5. (1%) 請描述在你的 semi-supervised 方法是如何標記 label，並比較有無 semi-surpervised training 對準確率的影響。
   (Collaborators:)

答： 因 gensim 使用 word to vector 會吃掉記憶體很大的容量，無法把 unlabel 一百多萬筆都吃下來， 這裡比較我使用的是助教 sample code 的模型，做完 token 之後在放到 embeded layer 幫我 tranform 成每一個字一百維的向量，我先用我上一題 train 好的模型給每一筆 unlabel data 上機率 , 再把機率大於 0.8 及小於 0.2 的抽出來放進 label data 中 , 這個步驟增加了我七十幾萬的樣本數，因此 sample size 為 九十萬 ，再把新的 training data 放到 LSTM 訓練，上傳的成績增加了 0.002 左右。