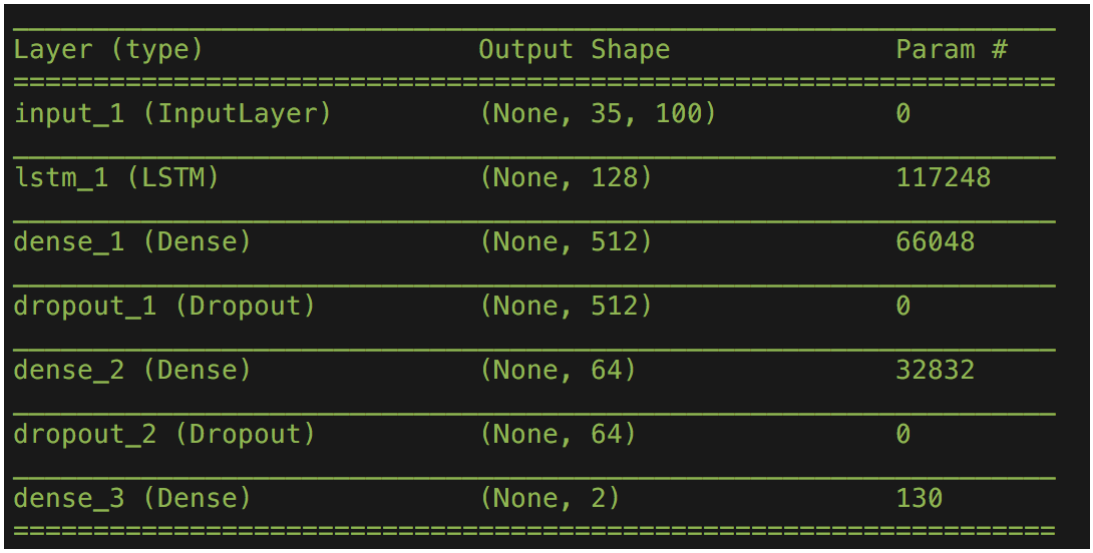
Report

學號：B05902002 系級： 資工二 姓名：李栢淵

(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？

答：

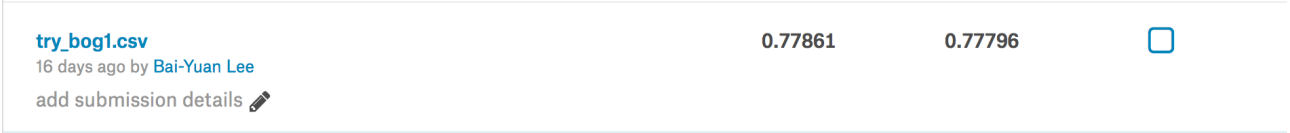
首先透過 gensim 將每一個字變成長度為 100 的 vector，經過一層 LSTM 後直接接 DNN，中間兩個 Dropout 都設很高(0.5) 防止 overfit，有設定 early stop，patience 為 3，平均大概三十幾個 epoch 就會停止，optimizer 為 adam，loss function 為 categorical cross entropy，準確率為：0.82209。下圖為模型架構截圖，文字預處理時有包含驚嘆號跟問號，其他刪去。



(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？

答：

透過 sklearn.feature_extraction.text 的 CountVectorizer先建一個 dictionary ，把每個句子都變成1-of-N enocoding，其中N為30000，只取30000個出現頻率較高的字，然後接 DNN，中間四個 Dropout 設的更高(0.8) ，因為超級容易 overfit，沒有設定 early stop，只讓他跑 4 個 epoch，optimizer 為 adam，loss function 為 categorical cross entropy，準確率為：77796。下圖為模型架構截圖。



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 30000)             0
_________________________________________________________________
dense_1 (Dense)              (None, 2048)              61442048
_________________________________________________________________
dropout_1 (Dropout)          (None, 2048)              0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               262272
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 512)               66048
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_4 (Dense)              (None, 64)                32832
_________________________________________________________________
dropout_4 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_5 (Dense)              (None, 2)                 130
=================================================================
Total params: 61,803,330
Trainable params: 61,803,330
Non-trainable params: 0
_________________________________________________________________
None
```

(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。

答：

第一句：today is a good day , but it is hot

第二句：today is hot , but it is a good day

bag of word給的分數：

```
---- Predict... ----
1/1 [==============================] - 0s
(2, 2)
[[0.3364392 0.6635608]
 [0.3364392 0.6635608]]
```

它認為兩句情緒分數一樣高，都是偏正面。

RNN給的分數：

```
---- Predict... ----
1/1 [==============================] - 0s
(2, 2)
[[0.4768873  0.5231127 ]
 [0.02647213 0.97352785]]
```

它認為兩句情緒分數不一樣高，第一句比較中立，第二句正面。

因為bag of word沒有考慮字的先後順序，所以兩句的分數會相同，而RNN考慮的字的順序，所以兩句結果會不一樣。

(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。
答：

模型架構跟參數跟第一題一樣，只有把標點符號拿掉，感覺只有變差一點點，應該是標點符號對於整句的理解還是有一定的重要性，才會有一點差別，。

有標點符號：

沒有標點符號：

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

我將 nolabel data 上用一個我做最好的 model 去 predict，如果判斷是0或是1的機率大於0.997，再把它挑出來並標記對應的label，加入training data，這樣抓出來的資料大概有四萬多筆。

最後的結果比本來沒標label的版本差了一點，而且在執行的過程中，semi-supervised training 的 acc 比 val_acc 高很多，顯然這樣做出來的結果更 fit training data，更 overfit 了，所以變差了。

有 semi-supervised training：

沒有 semi-supervised training：