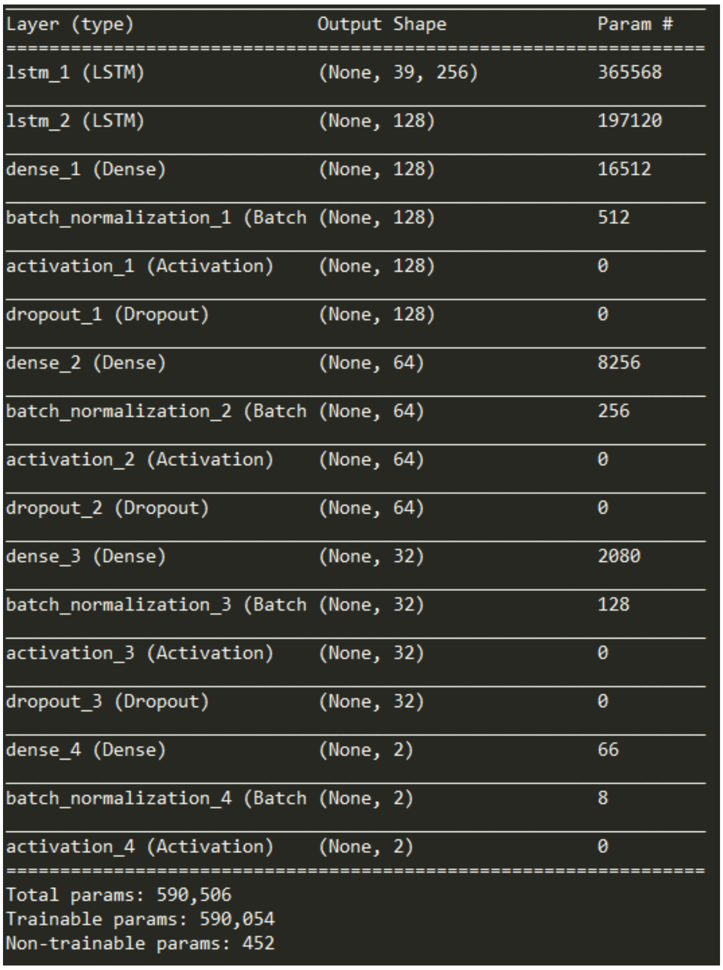
學號：R06942082 系級： 電信碩一 姓名：黃釋平

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: 鄭立晟_R05942148、姚嘉昇_R06922002)
答：

經過多次的嘗試，依然達不到標準，所以參考同學的 model，不同的是我對資料做了兩次 normalize，x=(x-mean)/var(x)，一次是 word 自己的 vector (dim=100) 做，第二次是進 model 之前對所有 vector 做。

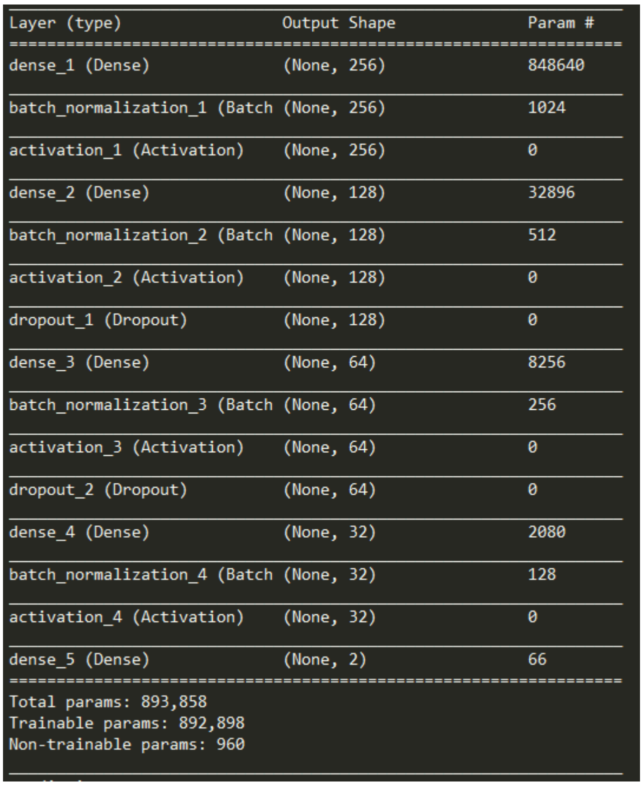
神奇的是，我在這樣的架構下試著多加 Dense(16)、Dense(8)、Dense(4)三層，使其遞減到 2，結果上傳卻降低，且應該沒有 overfitting。



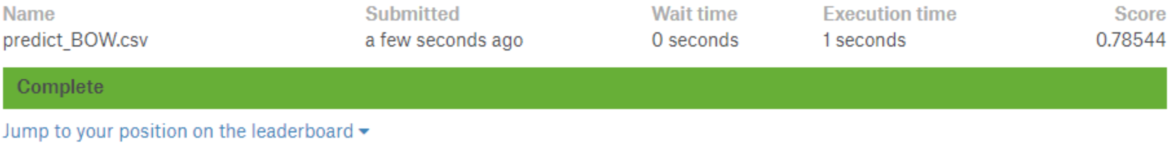
```
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 39, 256)           365568
_________________________________________________________________
lstm_2 (LSTM)                (None, 128)               197120
_________________________________________________________________
dense_1 (Dense)              (None, 128)               16512
_________________________________________________________________
batch_normalization_1 (Batch (None, 128)               512
_________________________________________________________________
activation_1 (Activation)    (None, 128)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256
_________________________________________________________________
batch_normalization_2 (Batch (None, 64)                256
_________________________________________________________________
activation_2 (Activation)    (None, 64)                0
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_3 (Dense)              (None, 32)                2080
_________________________________________________________________
batch_normalization_3 (Batch (None, 32)                128
_________________________________________________________________
activation_3 (Activation)    (None, 32)                0
_________________________________________________________________
dropout_3 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 66
_________________________________________________________________
batch_normalization_4 (Batch (None, 2)                 8
_________________________________________________________________
activation_4 (Activation)    (None, 2)                 0
=================================================================
Total params: 590,506
Trainable params: 590,054
Non-trainable params: 452
```

Best result: epoch=4, validation:0.820, training:0.834, kaggle:0.822

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：



```
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 256)               848640
_________________________________________________________________
batch_normalization_1 (Batch (None, 256)               1024
_________________________________________________________________
activation_1 (Activation)    (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               32896
_________________________________________________________________
batch_normalization_2 (Batch (None, 128)               512
_________________________________________________________________
activation_2 (Activation)    (None, 128)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 64)                8256
_________________________________________________________________
batch_normalization_3 (Batch (None, 64)                256
_________________________________________________________________
activation_3 (Activation)    (None, 64)                0
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_4 (Dense)              (None, 32)                2080
_________________________________________________________________
batch_normalization_4 (Batch (None, 32)                128
_________________________________________________________________
activation_4 (Activation)    (None, 32)                0
_________________________________________________________________
dense_5 (Dense)              (None, 2)                 66
=================================================================
Total params: 893,858
Trainable params: 892,898
Non-trainable params: 960
_________________________________________________________________
```



Name	Submitted	Wait time	Execution time	Score
predict_BOW.csv	a few seconds ago	0 seconds	1 seconds	0.78544

Complete

Jump to your position on the leaderboard ▾

使用 corpora 來做 BOW，這是目前調到最好的參數，**validation=0.786**，**上傳結果 0.785**，差異不大，但略比 LSTM 差，可能是**沒考慮到字的順序**。

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：

Kaggle Test	LSTM	bag of word
"today is a good day, but it is hot"	**[0.267 0.732]**	**[0.435 0.533]**
"today is hot, but it is a good day"	**[0.0003 0.9997]**	**[0.465 0.513]**

RNN 很確定判斷為好，兩句確定性差異較大;
BOW 雖然也都判斷為好，但只接近五成，且兩句結果差不多
可見字的順序確實影響到判斷結果。

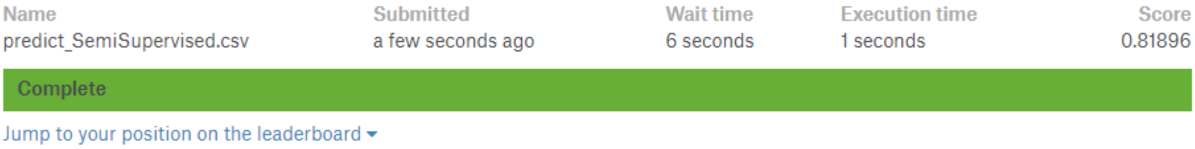
4. (1%) 請比較"有無"包含標點符號兩種不同 tokenize 的方式，並討論兩者對準確率的影響。
 (Collaborators:)
答：

Kaggle Test	LSTM	bag of word
保留標點符號	0.82211	0.78544
去標點符號	0.81038	0.78158

去除標點符號後平均表現皆變差，也許**符號對情緒的判斷是有參考價值的**。

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：

Kaggle Test	Non	Semi-Supervised
LSTM	0.82211	0.81896



Name	Submitted	Wait time	Execution time	Score
predict_SemiSupervised.csv	a few seconds ago	6 seconds	1 seconds	0.81896

Complete

Jump to your position on the leaderboard ▾

做了 Semi-Supervised 後，validation accuracy 飆高到 0.94，training accuracy 則是 0.86，都有提高，但上傳結果卻稍微降低一點。