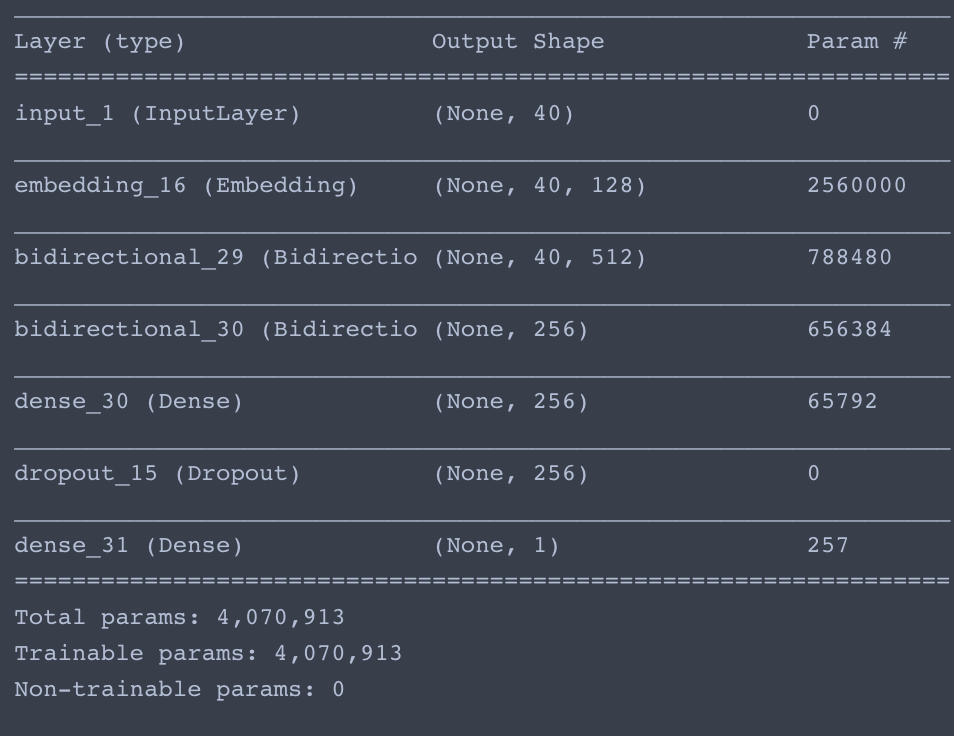
學號：R06725041 系級： 資管碩一 姓名：彭証鴻

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: sample code、keras official document、訓練RNN http://pcse.pw/74VB7 )

答：

**模型架構：**

我的RNN Model包含了一層Embedding layer、兩層雙向LSTM和一層Dense Layer，且optimizer選用Adam，Loss function選用binary\_crossentropy，output layer的activation function 採用較符合這次task的sigmoid function。最後我的epoch數量設為30，因為發現train到後面會有overfitting，所以最後以epoch15的model來predict最後結果。



**訓練過程：**

一開始沒有做semi-supervised，就過simple baseline了，但在助教公布strong baseline後，有發現光靠training\_data.txt數量是不夠的，所以後來就先夠過semi-supervised來label，最後再一起train。

Training的過程有發現做這個task容易出現overfitting，所以也在過程中加入L2 regularization、dropout layer來盡量避免overfitting。

**準確率：**

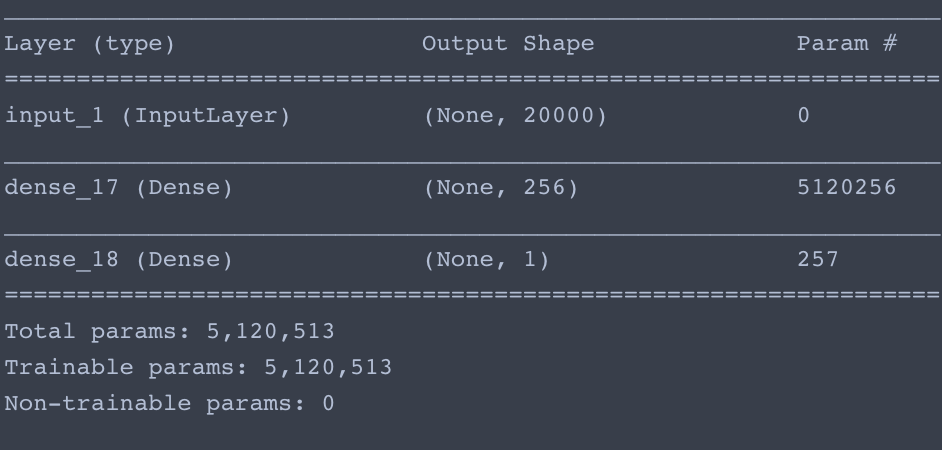
Public score: 0.81938 Private score:0.81690

1. (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: sample code、keras official document)

答：

**模型架構：**

模型架構為一層Dense， 一層output layer，並選用Adam作為optimizer，loss function則是binary crossentropy，與RNN一樣總共訓練30個epoch。



**訓練過程：**

train\_X經過tokenizer.texts\_to\_matrix()後，直接當作input，丟進DNN中，雖然訓練的速度比RNN快很多，但val\_acc到76%就上不去了。

**準確率：**

Public score: 0.76798 Private score:0.76940

1. (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: sample code)

答：

“ Today is a good day, but it is hot”, “Today is hot, but it is a good day”

BOW v.s. RNN

在bag of word model下，兩句的情緒分數相同，皆為0.5900535(正面情緒)，而RNN model下第一句的情緒分數為0.07861663(負面)，第二句則為0.00180691(負面)，儘管在RNN model下情緒是一樣的，但可以看出來第二句比第一句更接近負面情緒。

1. (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: sample code、keras official document)

答：

因為僅是為了比較”有無”包含標點符號對準確率的影響，所以為了節省等待時間我將batch\_size調得很大，所以這邊提供的準確率並非最好model的準確率。

**準確率：**

有包含標點符號： 無包含標點符號：

Public score:0.80588 Public score:0.79802

Private score:0.80359 Private score:0.79796

1. (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: sample code)

答：

做semi-supervised前，我先透過training\_data.txt訓練出四個的model，並透過各model的val\_acc來將predict的結果做weighted-sum，並將threshold設為0.2，意謂當結果大於0.8時，我們將此data label為1，結果小於0.2時，我們將此data label為0，最後再將符合threshold條件的data，與一開始的training\_label.txt中的concatenate起來，之後再將這些training data拿下去train新的model，由結果可看出因為training data數量變多，model學得更好了。

準確率：

Semi-supervised前：

Public score: 0.81441 Private score:0.81437

Semi-supervised後：

Public score:0.81938 Private score:0.81690