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。

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 40)                0
_________________________________________________________________
embedding_16 (Embedding)     (None, 40, 128)           2560000
_________________________________________________________________
bidirectional_29 (Bidirectio (None, 40, 512)           788480
_________________________________________________________________
bidirectional_30 (Bidirectio (None, 256)               656384
_________________________________________________________________
dense_30 (Dense)             (None, 256)               65792
_________________________________________________________________
dropout_15 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_31 (Dense)             (None, 1)                 257
=================================================================
Total params: 4,070,913
Trainable params: 4,070,913
Non-trainable params: 0
```

   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

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: 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。

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 20000)             0
_________________________________________________________________
dense_17 (Dense)             (None, 256)               5120256
_________________________________________________________________
dense_18 (Dense)             (None, 1)                 257
=================================================================
Total params: 5,120,513
Trainable params: 5,120,513
Non-trainable params: 0
```

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

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: 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。

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: 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

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: 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