Homework 5 Report -Text Sentiment Classifation

學號：B04501095 系級： 土木三 姓名：黃平瑋

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？**
   將training data 有label和沒有label的一起丟進gensim word2Vec 轉換成128維的vector ,只取出現次數大於三次的word, 其餘視為沒有出現, 再將所有的句子padding成40個字的長度, 不足則在後面補0

RNN的model結構訓練為下圖：

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 40, 256)           365568
_________________________________________________________________
lstm_2 (LSTM)                (None, 40, 256)           525312
_________________________________________________________________
lstm_3 (LSTM)                (None, 256)               525312
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               131584
_________________________________________________________________
batch_normalization_1 (Batch (None, 512)               2048
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 256)               131328
_________________________________________________________________
batch_normalization_2 (Batch (None, 256)               1024
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 128)               32896
_________________________________________________________________
batch_normalization_3 (Batch (None, 128)               512
_________________________________________________________________
dropout_4 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 258
=================================================================
Total params: 1,715,842
Trainable params: 1,714,050
Non-trainable params: 1,792
```

其他的訓練參數：
LSTM activation function: tanh
DNN activation function: relu
LSTM Drop out: 0.2
DNN Drop out: 0.3
Loss function = binary_crossentropy, optimizer = adam
batch size = 1024, epoch = 50(with early stoping and checkpoint)
training acc: 0.8365 , public acc: 0.83088, private acc: 0.82859

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？**

先用keras的Tokenizer取出出現次數前30000的字, 透過keras 的 **text_to_matrix**紀錄每個句子這30000個字出現的次數, 最後output一個 (200000, 30000) 的array當作training data, 並透過下圖的model訓練

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 1024)              30721024
_________________________________________________________________
dropout_1 (Dropout)          (None, 1024)              0
_________________________________________________________________
batch_normalization_1 (Batch (None, 1024)              4096
_________________________________________________________________
dense_2 (Dense)              (None, 512)               524800
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
batch_normalization_2 (Batch (None, 512)               2048
_________________________________________________________________
dense_3 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
batch_normalization_3 (Batch (None, 256)               1024
_________________________________________________________________
dense_4 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_4 (Dropout)          (None, 128)               0
_________________________________________________________________
batch_normalization_4 (Batch (None, 128)               512
_________________________________________________________________
dense_5 (Dense)              (None, 2)                 258
=================================================================
Total params: 31,417,986
Trainable params: 31,414,146
Non-trainable params: 3,840
```

Dropout rate = 0.4
Activation function = relu
loss = binary_crossentropy, Optimizer = Adam, Learning rate = 1e-4
batch size = 1024, epoch = 50(with early stoping and checkpoint)
training acc: 0.79884, public acc: 0.79725, private acc:0.79103
我的bow模型明顯比RNN差了不少

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。**

Sentence 1: "today is a good day, but it is hot"

Sentence 2: "today is hot, but it is a good day"

在RNN和bag of word 模型中, 上面兩句的預測結果均為正面(output為1), 但機率略有不同

	sentence 1	sentence 2
RNN	positive: 0.773	positive: 0.8749
Bag of Word	positive: 0.7498	positive : 0.7498

可以看出Bag of Word 預測兩個句子情感的機率是完全相同的, 因為兩個句子中的各個word出現次數完全相同, 只是順序稍微調換了一下, 但因為Bag of Word 訓練的過程不會將sequence的順序考慮進去, 所以造成兩個句子在Bag of word模型中看起來是一樣的data, 所以訓練的結果也是相同的
在RNN中由於有LSTM所以neural network是會將sequence考慮進去, 所以最後的兩個句子的結果也不一樣

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。**
要將標點符號去除, 我使用的是gensim.parsing.preprocessing 中的 strip_punctuation, 即可去除, 若沒有使用這個function, 即是把標點符號當成一般的字一起去做word embedding (RNN訓練的model和參數和第一題相同)

	training acc	public testing acc	private testing acc
含標點符號	0.8365	0.83088	0.82859
不含標點符號	0.82340	0.82302	0.82083

可以從上表看出, 不論是在training set還是testing set 中, 保留標點符號的表現都比去除標點符號的結果高了將近**1%**, 推測會有這樣的結果,是因為某些標點符號可能表達出人的情感, 如驚嘆號,問號,或是刪節號

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。**
我從no_label的data中挑選了400000筆, 用 supervised training的方式 train了20個 epochs, 再把model拿去fit這400000筆沒有no_label data, 將output結果 > 0.9 和 < 0.1 的data標上**pseudo label 1, 0** ,剩餘的資料則不取, 再將這些data和原本的data合在一起繼續train,最後的結果有些微的進步 (RNN訓練的model和參數和第一題相同)

	training acc	public testing acc	private testing acc
有semi-supervised	0.83972	0.83415	0.83133
無semi-supervised	0.83654	0.83088	0.82859