學號：R06944034 系級： 網媒碩一 姓名：黃禹程

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: 葉韋辰 R06922130、鄭克宣 R06921083、丁縉楷 R06922129)

答：僅 train 3 個 epoch，

val_acc 的過程為 0.8107->0.8295->0.8158（validation set 比例：0.07）

第二個 epoch 結果比第三個 epoch 還要好，在 kaggle 的分數亦是如此

模型的部分，從 keras 的模型摘要來看（如下圖），先通過兩層 RNN，再接六層 DNN，每層後面都接 Dropout

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 100)         5529000
_________________________________________________________________
bidirectional_1 (Bidirection (None, None, 512)         731136
_________________________________________________________________
bidirectional_2 (Bidirection (None, 512)               1574912
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_2 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 256)               65792
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 512)               131584
_________________________________________________________________
dropout_4 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_4 (Dense)              (None, 512)               262656
_________________________________________________________________
dropout_5 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_5 (Dense)              (None, 1024)              525312
_________________________________________________________________
dropout_6 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_6 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dropout_7 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_7 (Dense)              (None, 2)                 2050
=================================================================
Total params: 10,003,370
Trainable params: 10,003,370
Non-trainable params: 0
_________________________________________________________________
```

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: 葉韋辰 R06922130、鄭克宣 R06921083、丁縉楷 R06922129)

答：

train 5 個 epoch，每次的 val_acc 的過程為 0.5531-> 0.7092-> 0.7092-> 0.7092-> 0.7092 發現後面已經沒有再成長就停止了
模型的部分，疊了四層 DNN，activation function 使用 relu，每層後面都接 Dropout

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 256)               65064704
_________________________________________________________________
activation_1 (Activation)    (None, 256)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 256)               65792
_________________________________________________________________
activation_2 (Activation)    (None, 256)               0
_________________________________________________________________
dropout_2 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 256)               65792
_________________________________________________________________
activation_3 (Activation)    (None, 256)               0
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 514
_________________________________________________________________
activation_4 (Activation)    (None, 2)                 0
=================================================================
Total params: 65,196,802
Trainable params: 65,196,802
Non-trainable params: 0
_________________________________________________________________
```

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: 葉韋辰 R06922130、鄭克宣 R06921083、丁縉楷 R06922129)

答：

	RNN	BOW
today is a good day, but it is hot	0.52915359 0.47084644	0.41607726 0.58392274
today is hot, but it is a good day	0.11483829 0.88516164	0.41607726 0.58392274

在 RNN 的模型中會考慮到語句順序，因此在此題中，交換前後句會造成分數差異，而 bag of word 僅探討句子中出現的字，因此將前後具交換而不抽換句字時，bow 的分數會是相同的。

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: 葉韋辰 R06922130、鄭克宣 R06921083、丁縉楷 R06922129)

答：

「無」包含標點符號的 tokenize 法在 kaggle 上得到 0.81866 的分數
「有」包含標點符號的 tokenize 法在 kaggle 上得到 0.81997 的分數

無標點符號的版本比有標點符號的差，推測是考慮標點符號後，透過斷句更能貼近原句想要敘述的意思

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: 葉韋辰 R06922130、鄭克宣 R06921083、丁縉楷 R06922129)

答：

	non semi	semi
Kaggle public	0.81866	0.81359
Kaggle private	0.81743	0.81285

Semi-supervised 的效果不論在 public 或 private 都較差一些,
我的推測是：利用 predict 出的結果當作 training data 時，就已經承受了 data 中有 noise 的風險（semi data 有誤），有可能誤導了 model fit 的過程，一種丟入垃圾產出垃圾的概念。