學號：R06922152 系級： 資工碩一 姓名：袁晟峻

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: 翁瑋 )

   答：先使用 gensim 用 training_label、nolabel 和 testing_data 去訓練一個 word2vec model ， 再將 training_label 的資料丟進 model 內得到對應的 vector，vector 作為 RNN model 的 input，RNN model 的架構為一層 masking layer、兩層 bidirectional lstmu(分別是 128 個 neuron 和 64 個 neuron)以及一層的 dense layer，activation 都使用 sigmoid，最後出來的結果在 kaggle public 上得到 82.4%的準確率。

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
masking_1 (Masking)          (None, 40, 128)           0
_________________________________________________________________
bidirectional_1 (Bidirection (None, 40, 256)           263168
_________________________________________________________________
batch_normalization_1 (Batch (None, 40, 256)           1024
_________________________________________________________________
bidirectional_2 (Bidirection (None, 128)               164352
_________________________________________________________________
batch_normalization_2 (Batch (None, 128)               512
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 429,185
Trainable params: 428,417
Non-trainable params: 768
```

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: 陳柏堯)

   答：使用 keras 的 Tokenizer 來實作 BOW，將 training_label.txt 和 testing_data.txt 讀進來，濾掉一些符號後取最常出現的前 1000 個單字當作 dictionary，然後用 texts_to_matrix 將 training_label.txt 的資料做轉換，丟進五層的 DNN train 一發後在 kaggle public 上得到 77% 的準確率。

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 128)               128128
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_3 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_4 (Dense)              (None, 16)                528
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 17
=================================================================
Total params: 139,009
Trainable params: 139,009
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: )
   答：

| | Bag of word | RNN |
|---|---|---|
| today is a good day, but it is hot | 0.60716099 | 0.3333807 |
| today is hot, but it is a good day | 0.60716099 | 0.93466687 |

BOW model 只考慮關鍵字的出現次數，不會考慮出現次序，所以對組成單字一樣但順序不一樣的句子 predict 出的結果不會有差別。

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：

1.包含標點符號：kaggle public 0.80793

2.去掉標點符號：kaggle public 0.81192

根據自己做出的結果，去掉標點符號的準確率比較高，推測可能的原因為大部分的標點符號無法作為判斷語句情緒的依據。

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：我使用最簡單的 self-training，在標記 label 的部分使用兩種方式，第一種是用 0.5 當作 threshold，第二種是只取 predict 出的值>0.8 或<0.2 的(比較有把握?)來標記 label，兩種方式跟沒有做 semi-supervised 的 model 相比在 kaggle 上都獲得較高的 public 成績，而第二種標記 label 的方式(取>0.8 和 <0.2)又比第一種在 kaggle 上的 public 成績高，所以由實驗結果得出 semi-supervised 對提升我做的 model 的準確率是有幫助的。