學號：R06921037 系級： 電機所碩一 姓名：許哲瑋

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？**
   **(Collaborators: r06942010 蘇建翰)**

   Word embedding 主要使用 gensim 這個套件的 word2vec 來實現，將 trian file 裡面的單詞超過五次以上的紀錄在字典裡並轉換成 128 維的向量。

   RNN 的架構使用的是 LSTM，一共疊了三層，分別是 128，64，32，層與層之間都加上 Dropout，最後接上輸出層，使用的 activation 是 sigmoid，訓練 5 個 epoch。

```
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 30, 128)           131584
_________________________________________________________________
dropout_1 (Dropout)          (None, 30, 128)           0
_________________________________________________________________
lstm_2 (LSTM)                (None, 30, 64)            49408
_________________________________________________________________
dropout_2 (Dropout)          (None, 30, 64)            0
_________________________________________________________________
lstm_3 (LSTM)                (None, 32)                12416
_________________________________________________________________
dropout_3 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 33
=================================================================
Total params: 193,441
Trainable params: 193,441
Non-trainable params: 0
```

   最後這份程式碼上傳結果為 81.226%(public)，80.961%(private)的正確率(Kaggle 上面的最佳為 ensemble 過的結果)。

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: r06942010 蘇建翰)**

   **答：**利用 Keras 的 tokenizer.texts_to_matrix 方法，將每一句話轉換成 10000 維的向量，向量的每一維代表出現字的個數。最後接到一層輸出層，使用的 activation 是 sigmoid，訓練 5 個 epoch。

```
Build DNN model ...
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 1)                 5001
=================================================================
Total params: 5,001
Trainable params: 5,001
Non-trainable params: 0
_________________________________________________________________
None
```

   在 validation 上面達到近 79%準確率，在 test 預測為 78.13%(public)，77.67%(private)準確率。

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: r06942010 蘇建翰)**
   **答：**

   在 BOW 的模型裡，這兩句話預測為正面的情緒分數皆為 61.2755%。會造成分數相同的原因是因為這裡面所含的字全部都一樣，只是次序的不同，在 BOW 裡只考慮出現次數的話，這兩句話的情緒表現預測是相同的。在 LSTM(RNN)的模型裡，第一句話的正面情緒預測機率為 38.65%，第二句話的正面情緒預測機率為 98.05%。可以發現 LSTM(RNN)考慮了句子單詞順序關係，對於兩句話的預測才會有此落差。

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: r06942010 蘇建翰)**
   **答：**

   在 Kaggle 上面的表現有含標點符號的稍微好一點點(81.226% public)，不含標點符號的為(79.592%public)。猜想可能是因為標點符號有時候也隱含了情緒的意思(例如驚嘆號、問號)等等，但也會有些標點符號比較無法看出情緒(例如逗號、縮寫.)會是雜訊，但是綜合這些標點符號的考慮，有標點符號給予的資訊似乎比雜訊還要多一點。

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: r06942010 蘇建翰)**
   **答：**

   一開始做的時候我採取直接把 train 好的 model 預測沒有 label 的句子，根據預測的結果標上 label 後在一起拿進來 train 一次，結果發現預測的準確率從 81.226%(public)掉到 72.16%(public)。後來我採取信心水準 95%(超過 95%的標成 1，低於 5%標成 0，其餘資料不用)，然後拿進來和有 label 的 data 一起 train 一次，預測準確率從 81.226%進步到 81.474%。如果對於我們標上的 label 沒有太大的信心，反而會讓 model 的表現下降。