學號：R06723025 系級： 財金所碩一 姓名：林耘寬

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:)

我在 kaggle 上最佳的模型是由兩個類似架構的 GRU 模型 ensemble 而成。
模型的架構如下
Embedding layer ->
GRU (128, Dropout = 0.5)->
Dense ->
Dropout = 0.5 ->
…(重複三層)
Dense
每個句子長度限定 30，一個字用 128 維的向量去表示，並且使用 word2Vec 去建立字典。
Loss function 為 binary crossentropy，Optimizer 為 adam，batch size = 256。
準確率為：

Public	0.81640
Private	0.81586

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？

Bag of words 的模型架構在 tokenize 的時候從 texts_to_sequences 改成 texts_to_matrix，並且只用數層 DNN 模型，Dropout 皆為 0.5。
Loss function 為 binary crossentropy，Optimizer 為 adam，batch size = 256。

Public	0.79747
Private	0.79682

不論是 Public 或是 Private score 皆比 word2vec 方法做出來的低，原因可能是因為在 BOW 模型訓練中未考慮字出現的順序等等，所以會造成對於相同字但意思不同的兩句話作出不同的推論，如同下題。

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。

從人的角度來看，有些難去判斷到底這兩句話心情是好還是不好，畢竟是個好日子但又有點熱。
從 BOW 的模型當中，從下列圖表當中，可以發現兩者的分數相當接近，推測是因為 BOW 模型在訓練的時候，並未考慮到句子中字出現的順序，以及文法等等，因為兩句話出現的字皆相同，故他們得到的結果也較為相同。

而 RNN 模型訓練的時候是有考慮字出現的順序的，所以雖然字一樣，但訓練的結果不同，但從訓練的結果來看，感覺上機器未考慮到 but 這個轉折詞的重要性，所以這個模型可能還有改進的空間。

(0 的機率/1 的機率)	第一句話	第二句話
BOW	(0.22/0.78)	(0.22/0.78)
RNN	(0.68/0.32)	(0.08/0.92)

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。

可以發現沒有標點符號的 tokenize 方式結果略低於有的，推測是有時透過標點符號還是可以些許判斷一個人的情緒。

Public	0.81173
Private	0.81153

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。

我使用了 Self-training 來協助標記 label，做法便是將 label 的 data 去訓練一個模型，再用這個模型去 train unlabel 的 data，最後我們考慮機率大於 0.7 的情況下才將他加入原本的 train data 當中，可以發現準確率並無顯著提高，可能是因為一次便估計所有 unlabel data，因為 train 的時候電腦當掉幾次，所以只訓練成功

一次而已。
可能可以分批將 unlabel 資料加上 label，提高信心水準或者採用其他 simi-supervised 的標記方法可能會有比較好的效果。

Public	0.81291
Private	0.81252