學號 ： R06522828　系級 ： 機械碩一　姓名 ： 王榆昇
Collaborators: r06222023 王宇昕、r06522825 董士豪

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:)

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 40, 200)	0
gru_1 (GRU)	(None, 256)	350976
dense_2 (Dense)	(None, 1)	257

optimizer : adam、batch_size : 256、epochs : 20、loss : binary_crossentropy

我先使用 gensim 套件裡的 Word2Vec 函式，讓機器讀取 label 和 nolabel 的資料，並訓練出 200 維的詞向量，參數如下:

```
Word2Vec(self.data, size = 200, window = 8, min_count = 10, iter = 15);
```

再來我將 lable 的 data 取出、分割，設定 time_step = 40，也就是固定每個句子有 40 個單字，不足者往前補 0，並將每個單字轉換成上面訓練好的詞向量後傳入 GRU 層:

```
RNN_cell = GRU(256, return_sequences=False, dropout=0.3);
```

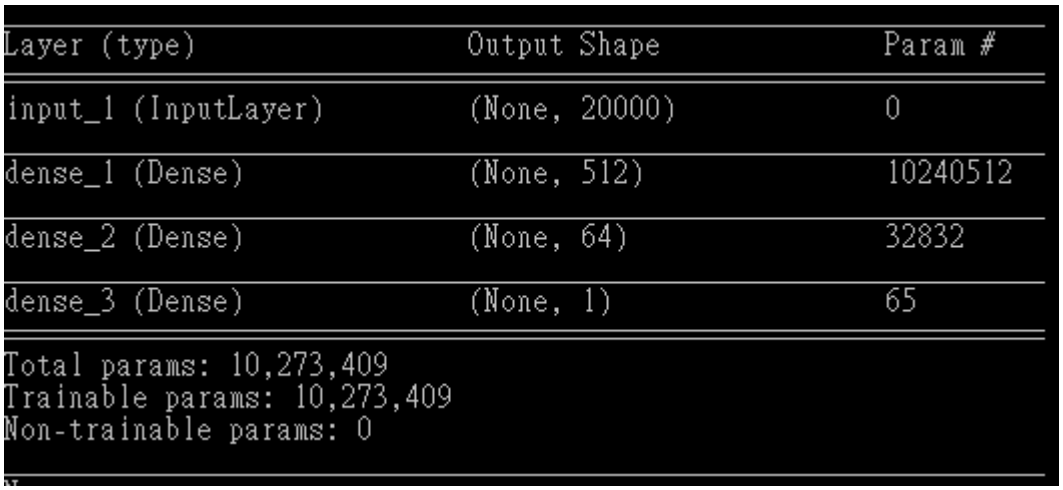
再經過一層 NN(hidden_size = 256、activation = relu)，最後通過 hidden_size = 1、activation = 'sigmoid' 的 NN 作為輸出。

結果:

	Public	Private
Kaggle	0.82252	0.82107

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:)

模型架構如下圖:



Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 20000)	0
dense_1 (Dense)	(None, 512)	10240512
dense_2 (Dense)	(None, 64)	32832
dense_3 (Dense)	(None, 1)	65

Total params: 10,273,409
Trainable params: 10,273,409
Non-trainable params: 0

optimizer: adam、batch_size: 256、epochs: 50

我將 90%資料作為 training data，10%資料作為 validation data，並建立 20000 個字的字典，把每個句子轉換成 20000 維向量後，輸入 3 層的 DNN 作訓練。訓練模型收斂很快，第 2 個 epoch 就 train accuracy 就到達 0.8521，val_acc 為 0.7840，但之後隨 train_acc 上升 val_acc 反而下降，最後使用 checkpoint 留下來的 best model。

	Public	Private
Kaggle	0.79627	0.79475

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:)

	Former	Later
BOW	0.58481699	0.58481699
RNN	0.35144705	0.96788132

從上表可以明顯看出，BOW 只看句子裡出現了甚麼單字，完全不考慮句子裡字的順序，因此兩句所預測的結果會一樣；相較之下 RNN 能照順序輸入單字，產生時序模型把單字的順序考慮進去(因果關係)，因此對這兩句能得到不同結果，也較符合真正的語意情緒。

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:)

利用 string 刪掉標點符號:

```
for idx, word in enumerate(cont):
    cont[idx] = word.translate(str.maketrans("","", string.punctuation));
```

Punctuation	Public	Private
Included	0.82252	0.82107
Not Included	0.81995	0.81993

對於這個 dataset，我們可以發現標點符號對於一句話的情緒還是提供了重要的訊息，我推測特別是 '!'、'?'、'...' 這三種標點符號特別重要，由標點符號配合 RNN 因果推論的能力，使得預測結果更準確。

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:)

Semi	Public	Private
Included	0.82254	0.82077
Not Included	0.82252	0.82107

將 nolabel 的 data 讀進來後，用同樣的 Word2Vec 模型轉換向量，並進入模型預測

```
semi_pred = RNN_model.predict(x_semi_all, batch_size=1024, verbose=True);
```

得到預測值後，設定 threshold = 0.02，也就是預測值 >0.98 或 <0.02 的資料才會加入下次的 training data，並在每個 epoch 後都重新 predict 並處理一次。

在加入 semi data 後，kaggle 上的分數並沒有太多變化，我認為是因為 semi 裡多加的訓練資料，都是模型裡非常確定結果的資料，換言之沒有太多"新"的東西供模型學習，故沒有太大差別。