學號：R05922080 系級： 資工碩二 姓名：王鵬傑

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: 無)

```
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 32, 1024)          20481024
_________________________________________________________________
bidirectional_1 (Bidirection (None, 32, 1024)          4721664
_________________________________________________________________
bidirectional_2 (Bidirection (None, 64)                202944
_________________________________________________________________
dense_1 (Dense)              (None, 128)               8320
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 129
=================================================================
Total params: 25,414,081
Trainable params: 25,414,081
Non-trainable params: 0
```

資料的部分是先透過 Tokenizer(20000, 32)將資料處理好

模型先通過 Embedding

經過第一層 Bidirectional(GRU(512, return_sequences=True, dropout=0.4))

再經過第二層 Bidirectional(GRU(32, dropout=0.4))

最後的 Output 利用 Dense(128)

以及 Dense(1, activation='sigmoid')

Optimizer 為 Adam，Learning rate 為 Default 值

Batch size 為 512, Epoch 為 10

訓練過程，經過 4 個 Epoch 就 Earlystopping

Epoch	Train Loss	Train Acc	Valid Loss	Valid Acc
1	0.5021	0.7493	0.4458	0.7962
2	0.4160	0.8090	0.4195	0.8095
3	0.3763	0.8314	0.4140	0.8155
4	0.3445	0.8482	0.4280	0.8141

最後在 Kaggle 上分數為

	Public	Private
Accuracy	0.81852	0.81625

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: 無)

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 32, 1024)          20481024
_________________________________________________________________
flatten_1 (Flatten)          (None, 32768)             0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              33555456
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 1025
=================================================================
Total params: 54,037,505
Trainable params: 54,037,505
Non-trainable params: 0
```

資料的部分是先透過 Tokenizer(20000, 32)將資料處理好

模型的部分先將資料做 Embedding

再將資料用 Flatten 攤開

再接一層 Dense(1024)經過 LeakyReLU 後

最後為 Dense(1, activation = 'sigmoid')用來做 Predict

過程當中經過 4 個 Epoch 時就 Earlystopping 了

Optimizer 為 Adam，Learning rate 為 Default 值

Batch size 為 512, Epoch 為 10

訓練時的 Accuracy 如下：

Epoch	Train Loss	Train Acc	Valid Loss	Train Loss
1	3.7465	0.5829	0.5763	0.6995
2	0.5057	0.7551	0.5039	0.7523
3	0.4246	0.8062	0.4942	0.7684
4	0.3672	0.8385	0.5263	0.7632

最後的 Kaggle 結果為：

	Public	Private
Accuracy	0.77008	0.76797

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: 無)

	"today is a good day, but it is hot"	"today is hot, but it is a good day"

BOW	0.3687818	0.4851423
RNN	0.5460985	0.9421483

原因可能有幾項

a. Bag of word 的準確率本身就沒有到很高，因此 predict 會錯誤
b. Bag of word 沒有時序的概念，因此兩句話對他來說是相同的

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: 無)

	Public	Private
有使用標點符號	0.81852	0.81625
無使用標點符號	0.78480	0.78416

標點符號在語句當中也扮演重要的角色，以下句來當作例子

Testing data 中 yes it ' s 3 : 50 am . yes i ' m still awake . yes i can ' t sleep . yes i ' ll regret it tomorrow . haha i love you mr Saturday. 在有標點符號的 model 當中為正面，而反而在無使用標點符號的 model 則為負面。

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: 無)

a. 利用 Pretrain 好的 model 先對 no label 的 data 做 Predict，若 predict 出來是 1 的機率>0.9 則將他 label 為 1，若<0.1 的話 label 為 0。
b. 將新標記的 Data 加入原本 label 好的 data。
c. 再重新訓練一次模型，重複五次同樣的 RNN model。

	Public	Private
Accuracy	0.81479	0.81182

有可能是因為 Threshold 的調整，會影響整體的效果，因此在這邊用 Semi-supervised 效果並沒有比較好。