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: none)

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 36, 64)	4977088
conv1d_1 (Conv1D)	(None, 36, 64)	12352
leaky_re_lu_1 (LeakyReLU)	(None, 36, 64)	0
batch_normalization_1 (Batch	(None, 36, 64)	256
max_pooling1d_1 (MaxPooling1	(None, 18, 64)	0
dropout_1 (Dropout)	(None, 18, 64)	0
bidirectional_1 (Bidirection	(None, 18, 1024)	2363392
batch_normalization_2 (Batch	(None, 18, 1024)	4096
dropout_2 (Dropout)	(None, 18, 1024)	0
bidirectional_2 (Bidirection	(None, 512)	2623488
batch_normalization_3 (Batch	(None, 512)	2048
dropout_3 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 128)	65664
batch_normalization_4 (Batch	(None, 128)	512
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
batch_normalization_5 (Batch	(None, 64)	256
dropout_5 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 1)	65

Total params: 10,057,473
Trainable params: 10,053,889
Non-trainable params: 3,584

以上為我的 RNN model 架構，在 training data 的句子中，最長的一句有 36 個字，因次我在經過 embedding layer 前會把每一句句子都 padding 到 36 的長度，然後 10 % 資料為 validation data。另外以下是我設為固定的參數：

- **embedding_vecor_length = 64**
- **EPOCHS = 7**
- **BATCHSIZE = 64**

一開始我模型在過 embedding layer 後，是馬上接 lstm 層，試著調 hidden size 看看我的 val_acc 大都徘徊在 0.78 多，有時會到 0.79，後來我試著再進 RNN 前加上

一層 CNN，後來 val_acc 的準確率都能在 0.79 上，不過這樣其實很免強才過了 simple baseline。 另外 RNN model 真的是訓練練得超久，以 mac 直接訓練，平均一圈要一萬多秒初。

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: none)

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 512)	5120512
batch_normalization_1 (Batch	(None, 512)	2048
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
batch_normalization_2 (Batch	(None, 256)	1024
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 128)	32896
batch_normalization_3 (Batch	(None, 128)	512
dropout_3 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 64)	8256
batch_normalization_4 (Batch	(None, 64)	256
dropout_4 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 1)	65

Total params: 5,296,897
Trainable params: 5,294,977
Non-trainable params: 1,920

這是我的 BOW model 架構，一開始我以所有 training data 來做字典，parse 完後有 7 萬多個字，所以在丟進去 DNN 做訓練時會有記憶體不夠的問題，後來我只取 10000 個字來做 bag of words 才有辦法做訓練。
在參數的方面我設定 EPOCHS = 4， BATCHSIZE = 64 與設定 10 % 資料為 validation data，這些都是固定的。
我會試著調整 hidden size 與 dropout 的參數去做訓練，不過整體來說 val_acc 大都都徘徊 0.79 多左右。

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: none)

首先我先新增 txt 檔包含這兩句，並執行 predict。

MODEL / SENTENCE	today is a good day, but it is hot	today is hot, but it is a good day
BOW	0.64194441	0.64194441
RNN	0.33159536	0.82324272

首先可以看出 BOW model 預測的一模一樣，因為 BOW 在處理 data 本來就沒有在意文字順序，所以這兩句雖然句子順序不同，但是打包成 bag 後 encoding 的 vector 一模一樣，所以預測的結果一樣。
而 RNN model 的 encoding vector 本身就不一樣，而且 lstm 層又在意文字順序，自然結果預測就不一樣。

而就這兩句的結果，我覺得 RNN model 就預測的很對，左邊那句先誇讚但後句就帶點抱怨，右邊則是樂觀的情形，所以預測是正確的。

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: noen)

因為 RNN model 實在是訓練太久了，這題我以訓練 BOW model 來做實驗。參數的方面一樣我設定 EPOCHS = 4， BATCHSIZE = 64 與設定 10 % 資料為 validation data。只差在設定 tokenizer = Tokenizer(num_words=NUMSOFWORDS, filters=“”)， filters 特別設為空的。

	with punctuation	without punctuation
val_acc	0.7929	0.7938
kaggle	0.79240	0.79244

我的實驗結果是把符號給去掉，效果都會比帶有符號的 encoding 效果都還好，但是分數其實也差異不大。

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: none)

在 semi-supervised 我標記 label 的方式是先訓練20萬筆有標記的資料，然後預測 no label 的資料，並幫他們標上我預測的結果。接著再把所有資料再放下去一起做訓練。在 RNN model 我只試了一次，因為資料量超級大，我並沒有 GPU 可以 train，所以花了非常多的時間再訓練。val_acc 最後也為 0.79 多而已，所以丟上 kaggle 只有 0.79403。我想是因為原本的 model 就沒有很強。以這種方式做 semi-supervised 其實效果不會很好。下次應該試其他 semi-supervised 的方法。