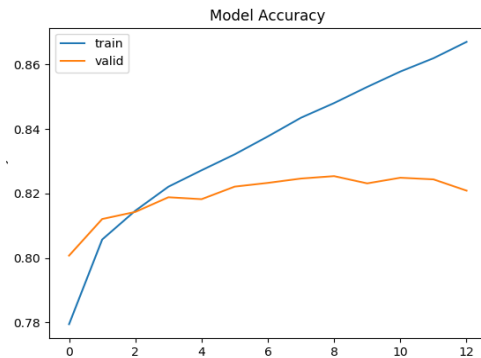


1. (1%) 請說明你實作的 RNN model，其模型架構、訓練過程和準確率為何？

(Collaborators:)

訓練過程我其實沒有用太複雜的架構，純粹改自於助教的 code，我將參數加深加厚，然後將標點符號消去做資料處理完以後，和 testing data 消去逗號的問題後基本上就完成了，實做 RNN 架構及訓練時的準確率如下圖：

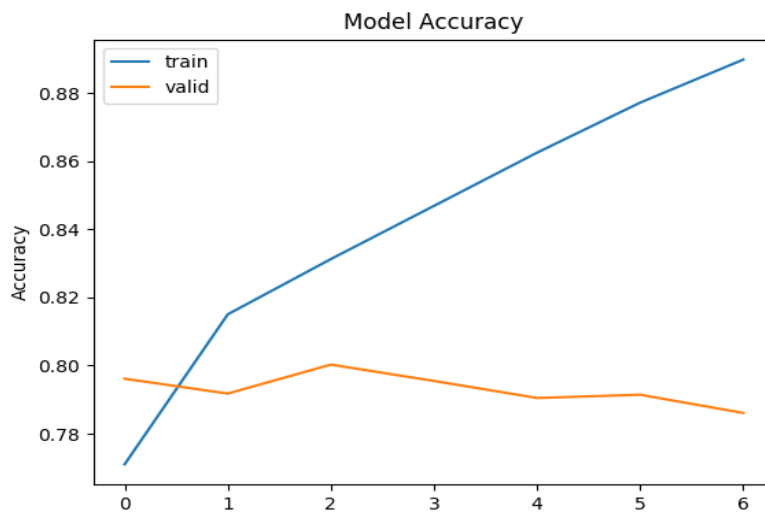
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 40)	0
embedding_1 (Embedding)	(None, 40, 128)	2560000
lstm_1 (LSTM)	(None, 512)	1312768
dense_1 (Dense)	(None, 256)	131328
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 1)	257
Total params: 4,004,353		
Trainable params: 4,004,353		
Non-trainable params: 0		



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:)

我是改用助教所提供的 simple code 下去改的，出來的結果沒有比較好，猜測原因是因為重複出現的字詞 被簡化的其實效用不大，而且順序也許是重要的。



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:)

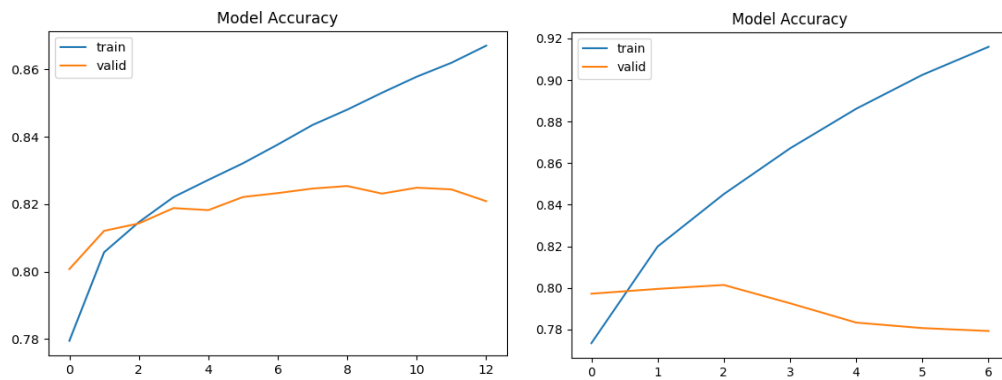
BOW 會挑出 today(1) is(2) a(1) good(1) day(1) but(1) it(1) hot(1)，對兩個句子來說結果一樣，RNN 就是照順序的丟進去所以兩個句子 會有先後的差別，一個是 good (+)-> but(-)另一個是 but(-) -> good(+)。所以實際丟進 train 好的 model BOW 出來的結果分數相同(0.6945)，RNN 的結果則是前面分數比較高(0.3526)，後面分數比較低(0.9063)。

```
[[ 0.69449592]
 [ 0.69449592]] [[ 0.35263261]
 [ 0.90630227]]
```

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:)

無標點符號對於準確率的預測較佳，我認為原因是因為標點符號的出現是為了讓人類段句，大部分的時候並沒有包含情緒在裡面，而且即是說是要表達情緒得標點符號，例如：!!!! 或 ???? 通常也只是代表更加強烈的意思本身為中性，沒有對應到這次要預測的情緒當中。實驗結果如下：左圖是有消去的，右圖沒有



5. (1%) 請描述在你的 semi-supervised 方法是如何標記 label，並比較有無 semi-supervised training 對準確率的影響。

(Collaborators:)

發現使用了 semi-supervised learning 對於準確率並沒有太大的幫助，我所使用的架構為助教所提供的 sample code 下去改的標記 label 的方法是設定 threshold 當預測出來的結果高於0.9則納入合法的 label.開始學習，然而這很看一開始的預測好不好，如果你一開始的預測就很差，節國即使你把 threshold 條的在嚴格也沒啥幫助我自己的話是沒有用到 semisupervised training 來產生最終的結果。而且在我發現一個奇怪的現象是在 semisupervised training 後的 validation accuracy 雖然提高了，但是傳上 kaggle 的結果卻慘不忍睹(0.6多)