學號：R06921083 系級： 電機碩一 姓名：鄭克宣

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:)
   答：

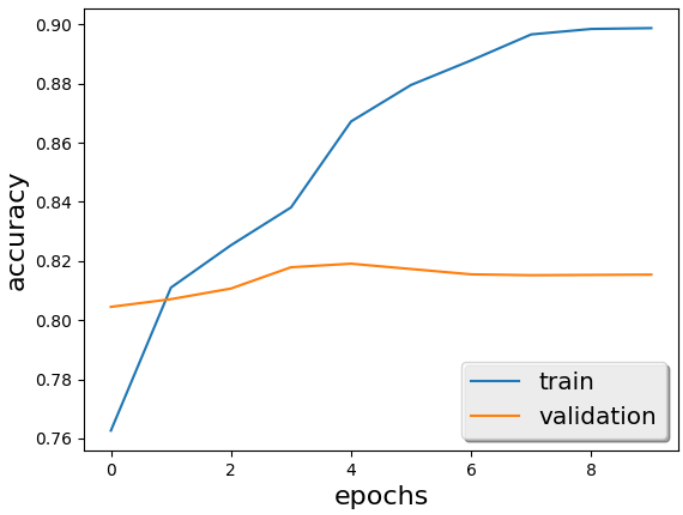
1) 訓練過程

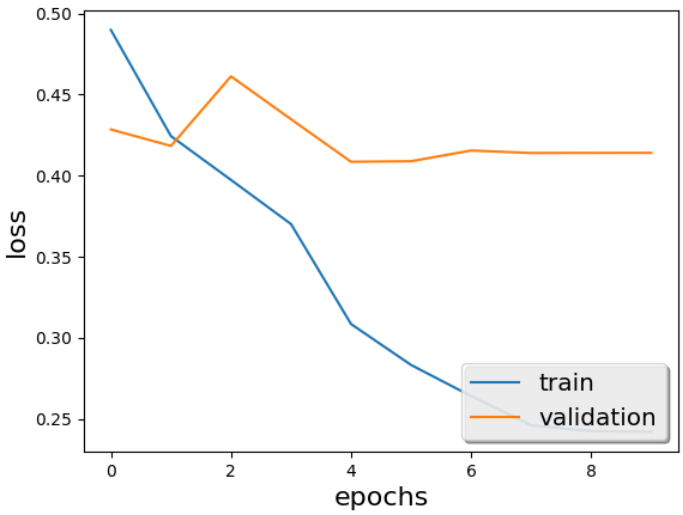
先使用 word2vec 做 word vector，再將文字轉化成 word id，其中 padding 時，使用 id 為 0 補齊，其中 id 為 0 所對應的文字為 'i'，再使用如下 bidirectional LSTM 搭配 DNN 訓練，其中 Activation 皆是 ReLU (除 LSTM 層未用及輸出層用 sigmoid)，Optimizer 使用 Adam，Loss 採 Cross Entropy 計算，另外有做 Early Stopping，Reduce Learning Rate on Plateau ，以及每個 epoch 紀錄模型，最後手動挑選最好的預測。Epoch 和 Loss 及 Accuracy 關係如下圖。

2) 模型及準確率

Model	Accuracy (上) & Loss (下)

```
Layer (type)                 Output Shape
=================================================
embedding_1 (Embedding)      (None, None, 100)
_________________________________________________
bidirectional_1 (Bidirection (None, None, 512)
_________________________________________________
bidirectional_2 (Bidirection (None, 512)
_________________________________________________
dropout_1 (Dropout)          (None, 512)
_________________________________________________
dense_1 (Dense)              (None, 256)
_________________________________________________
dropout_2 (Dropout)          (None, 256)
_________________________________________________
dense_2 (Dense)              (None, 256)
_________________________________________________
dropout_3 (Dropout)          (None, 256)
_________________________________________________
dense_3 (Dense)              (None, 512)
_________________________________________________
dropout_4 (Dropout)          (None, 512)
_________________________________________________
dense_4 (Dense)              (None, 512)
_________________________________________________
dropout_5 (Dropout)          (None, 512)
_________________________________________________
dense_5 (Dense)              (None, 1024)
_________________________________________________
dropout_6 (Dropout)          (None, 1024)
_________________________________________________
dense_6 (Dense)              (None, 1024)
_________________________________________________
dropout_7 (Dropout)          (None, 1024)
_________________________________________________
dense_7 (Dense)              (None, 1)
=================================================
```





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:)
答：

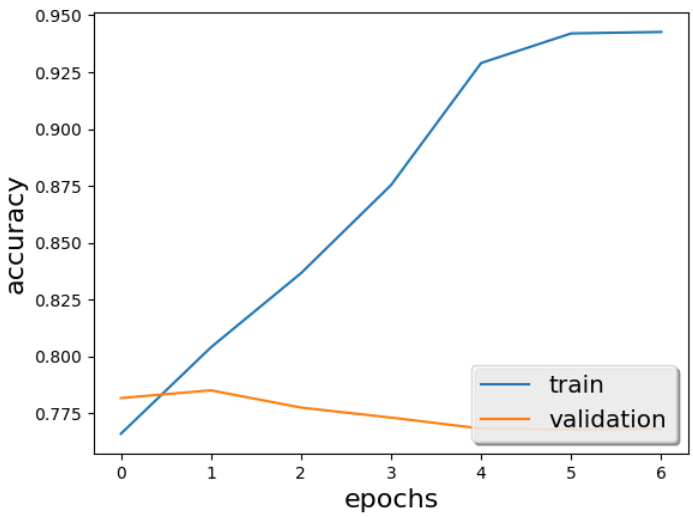
1) 訓練過程

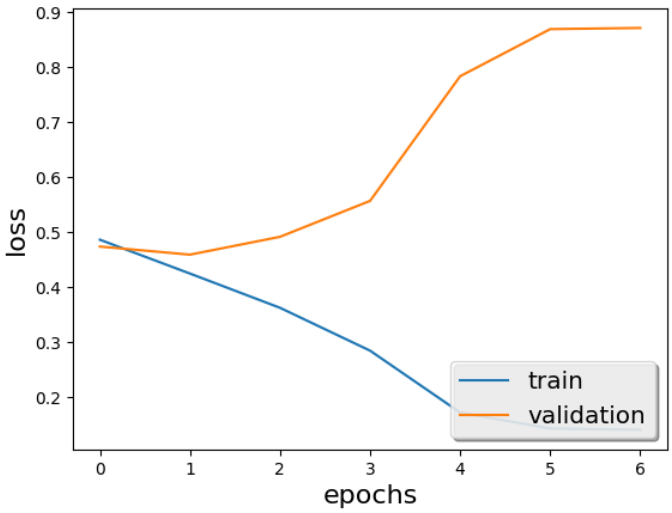
只取頻率最高的 2000 字做 BOW (超過 2000 會有 Memory Error) ，在使用以下 DNN 模型訓練，如同第 1 題，Activation 皆是 ReLU (除輸出層用 sigmoid)，Optimizer 使用 Adam，Loss 採 Cross Entropy 計算，另外有做 Early Stopping，Reduce Learning Rate on Plateau ，以及每個 epoch 紀錄模型，最後手動挑選最好的預測。Epoch 和 Loss 及 Accuracy 關係如下圖。

2) 模型及準確率

Model	Accuracy (上) & Loss (下)

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	512256
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 256)	65792
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 512)	131584
dropout_3 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 512)	262656
dropout_4 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 1024)	525312
dropout_5 (Dropout)	(None, 1024)	0
dense_6 (Dense)	(None, 1024)	1049600
dense_7 (Dense)	(None, 1)	1025





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:)
   答：

1) 情緒分數 (輸出層經 sigmoid 算出的值)

	RNN (LSTM)	BOW (DNN)
today is a good day, but it is hot	0.957	0.7453
today is hot, but it is a good day	0.938	0.7453

2) 討論

可以發現這兩句的使用的字及其字數接相同，在 BOW 中所得出來的值在這兩句都一樣，個人猜測之所以 BOW 之所以分數沒那麼高是因為有 'but' 這字。令人意外的是我的 RNN model 竟然是 "today is a good day, but it is hot" 比較接近正面情緒。

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:)
   答：

1) 比較"有無"包含標點符號兩種不同 tokenize 的方式

個人不包含標點符號是使用 keras 的 text_to_sequence 去掉所有標點符號，含標點符號則是使用 re 套件去做。

2) 準確度影響的討論

	Kaggle Public Accuracy	Kaggle Private Accuracy
Exclude char	0.82044	0.81926
Include char	0.81428	0.81412

將標點符號加入訓練，並沒有提昇準確度，不過相差也不算太多，但是訓練時間較長 (padding 後，feature 數增加)。

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:)
   答：

1) 方法

使用 2 個模型，分別為 GRU 和 LSTM 預測，將兩個預測為相同答案標記，在將其加入原本的 Train Data 一起使用第 1 題模型訓練。

2) 準確度比較

	Kaggle Public Accuracy	Kaggle Private Accuracy
Supervised	0.82044	0.81926
Semi-Supervised	0.81536	0.81350

Semi-supervised 並沒有比較好，我想可能是決定未標記資料的 label 的模型過於類似，但也可能是已標記資料原本標記就不是很好，所以作 semi-supervised 沒什麼進步。