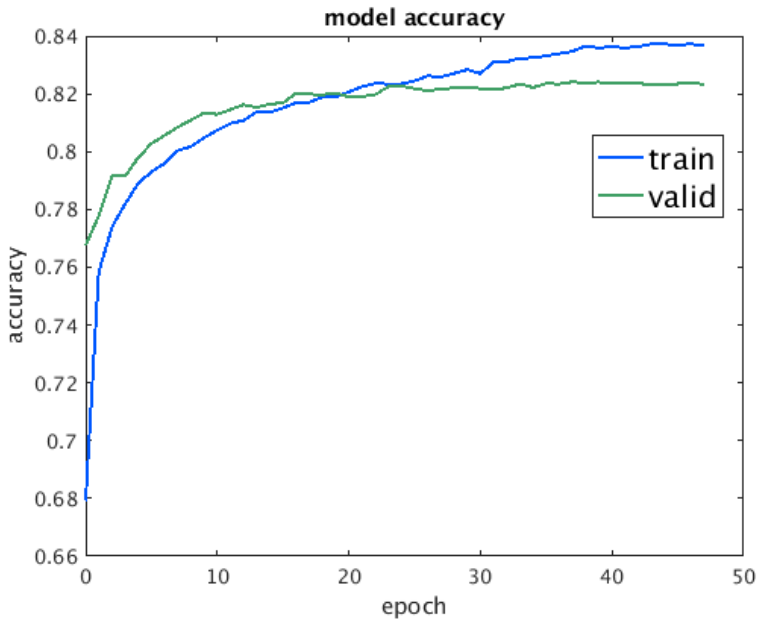
學號：R05943136 系級： 電子碩二 姓名：盧真玄

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？

Layer (type)	Output Shape	Param #
masking_1 (Masking)	(None, 39, 69)	0
bidirectional_1 (Bidirection	(None, 256)	152064
dense_1 (Dense)	(None, 256)	65792
batch_normalization_1 (Batch	(None, 256)	1024
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 512)	131584
batch_normalization_2 (Batch	(None, 512)	2048
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 256)	131328
batch_normalization_3 (Batch	(None, 256)	1024
dropout_3 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 128)	32896
batch_normalization_4 (Batch	(None, 128)	512
dropout_4 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 1)	129

Total params: 518,401
Trainable params: 516,097
Non-trainable params: 2,304

訓練過程的準確率如下：



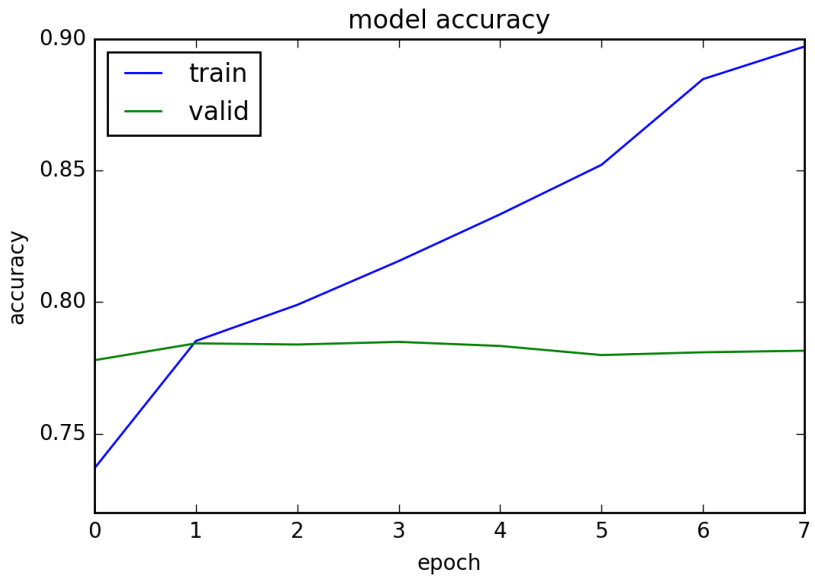
Kaggle Accuracy:
Public: 0.82523
Private: 0.82329

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: 賴又誠)

Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 256)	689152
batch_normalization_2 (Batch	(None, 256)	1024
dropout_2 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 512)	131584
batch_normalization_3 (Batch	(None, 512)	2048
dropout_3 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 256)	131328
batch_normalization_4 (Batch	(None, 256)	1024
dropout_4 (Dropout)	(None, 256)	0
dense_6 (Dense)	(None, 128)	32896
batch_normalization_5 (Batch	(None, 128)	512
dropout_5 (Dropout)	(None, 128)	0
dense_7 (Dense)	(None, 1)	129

Total params: 989,697
Trainable params: 987,393
Non-trainable params: 2,304

訓練過程的準確率如下：



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。

   我的結果如下：
   [1]. today is a good day, but it is hot
   [2]. today is hot, but it is a good day

Model	[1]	[2]
RNN	0.21733524	0.9658373
BOW	0.63751124	0.63751124

   從結果可以看出 RNN 預測得較準確，原因是 RNN 有加入時間的概念，而一句話的先後順序確實會影響它想表達的意思，例如例子裡的兩句話，從人類的眼裡看很清楚知道[1]是 negative，因為 hot 這個字眼在 but 後面；而[2]就是 positive，因為 good day 這個字眼放在 but 後面。假如沒有加入時間的概念，這兩句話根本是一樣的，因此才會造成我們所看到的結果。

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。

   我嘗試了有無標點符號兩種不同的 tokenize 方式，結果在 kaggle 上的準確率發現有加標點符號的正確率會比去掉標點符號的高，結果如下：

標點符號	public	private
有	0.82523	0.82329
無	0.81960	0.81767

   有標點符號的 predict 結果大約都比無標點符號的高了 0.05，我猜這是因為很多時候標點符號可以更正確的表示出一段文字的感情，也正因如此，沒有標點符號的結果當然會比較差。

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。

Method	How	Accuracy
Supervised	200000 labeled	0.82251
Semi-supervised	200000 labeled + 200000 non-labeled	0.82523

   可以從結果看出用 semi-supervised 正確率會比較高，我猜原因有兩個，第一是資料量變多了，第二是讓字典裡的字變多了，這樣就能避免再 testing 讀到不認識的字。