學號：r05229014 系級： 大氣碩二 姓名：鄒適文

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: r05229016 羅章碩)

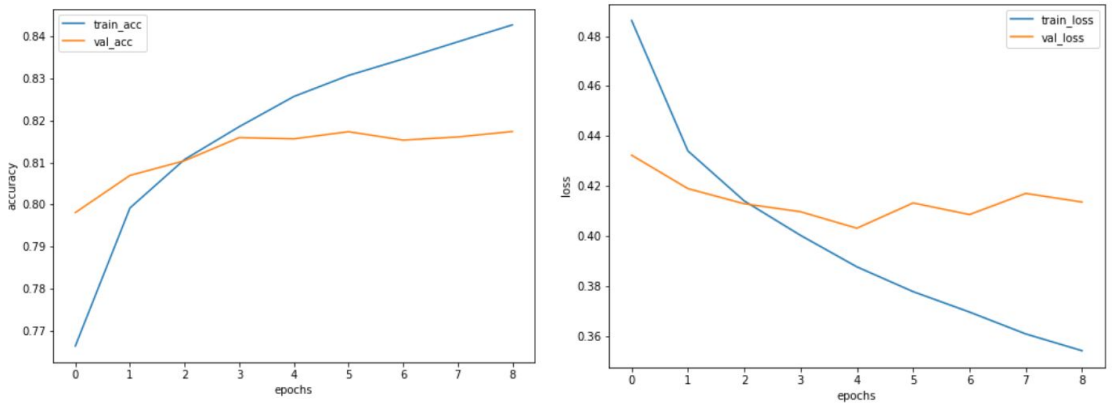
答：我先使用gensim的Word2vector訓練了我自己的詞向量，並以其當作RNN MODEL的embedding_layer，其中我把word在所有助教提供的data裡面，出現次數20次以下的字都濾掉，並且將其維度設為128維，最後把句train_label data裡面的字補到36作為我的training data，接著以下是RNN的模型架構。

架構：

Layer (type)	Output Shape	Param #
embedding_8 (Embedding)	(None, 36, 128)	2800512
lstm_12 (LSTM)	(None, 36, 300)	514800
lstm_13 (LSTM)	(None, 300)	721200
dense_19 (Dense)	(None, 128)	38528
dropout_3 (Dropout)	(None, 128)	0
dense_20 (Dense)	(None, 64)	8256
dropout_4 (Dropout)	(None, 64)	0
dense_21 (Dense)	(None, 1)	65

Total params: 4,083,361
Trainable params: 1,282,849
Non-trainable params: 2,800,512

訓練過程：



準確率：

在validataion_set的準確率有到達0.81740,在train_set之準確率有到達0.8428,而在kaggle上的分數為：private_set ： 0.81738　　public_set ： 0.81903

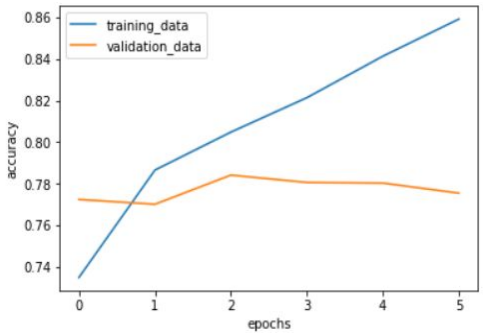
2. **(1%) 請說明你實作的 BOW model，其模型架構、訓練過程和準確率為何？ (Collaborators:** r05229016 羅章碩**)**

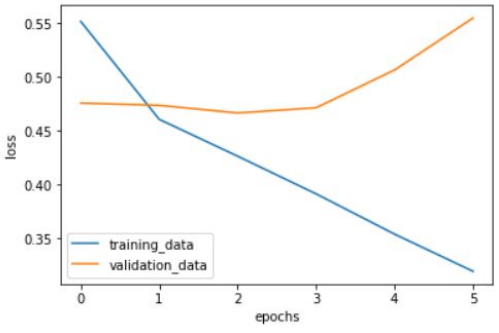
答：我的bow model，字典數量3000(記憶體不太夠QQ)， 並在keras的tokenizer.texts_to_matrix 使用 count模式，而bow model後面連著如下圖之dnn(batch_size=128)

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 200)	600200
dropout_1 (Dropout)	(None, 200)	0
batch_normalization_1 (Batch	(None, 200)	800
dense_2 (Dense)	(None, 400)	80400
dropout_2 (Dropout)	(None, 400)	0
batch_normalization_2 (Batch	(None, 400)	1600
dense_3 (Dense)	(None, 800)	320800
dropout_3 (Dropout)	(None, 800)	0
batch_normalization_3 (Batch	(None, 800)	3200
dense_4 (Dense)	(None, 1600)	1281600
dropout_4 (Dropout)	(None, 1600)	0
batch_normalization_4 (Batch	(None, 1600)	6400
dense_5 (Dense)	(None, 3200)	5123200
dropout_5 (Dropout)	(None, 3200)	0
dense_6 (Dense)	(None, 1)	3201

Total params: 7,421,401
Trainable params: 7,415,401
Non-trainable params: 6,000

接著我將資料以9:1切成training set, validataion set，並使用earlystopping(val_acc, patience=3)，而下兩圖是訓練的過程。在kaggle上public_set得到了0.76353的分數。





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: r05229016 羅章碩)

答：

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

模式	RNN	bag of word
分數	0.41354498	0.90514797

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

模式	RNN	bag of word
分數	0.93478239	0.90514797

首先，以我的認知，第二句話**"today is hot, but it is a good day"**的分數會要比第一句**"today is a good day, but it is hot"**，還要高，而RNN model因為有考慮順序、時間性的關係，在預測這兩句話的差別比較符合我24年以來的直覺，而bag of word因為他是不考慮順序、時間性的文字處理方法，所以這兩句話對他來說都是一樣的，所以或許可以說RNN在文字的情感分析上是比較適合的model。

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: r05229016 羅章碩)

   答：

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, 36, 128)           23040000
_________________________________________________________________
dropout_7 (Dropout)          (None, 36, 128)           0
_________________________________________________________________
lstm_5 (LSTM)                (None, 128)               131584
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 129
=================================================================
Total params: 23,171,713
Trainable params: 23,171,713
Non-trainable params: 0
```

我以上面的模型架構去做測試，只有一層LSTM，在有無標點符號的句子長度都補到36，字典用了全部的DATA取出最常出現的20000字，接著把同樣參數量的模型都TRAIN到earlystop。

沒有標點符號在kaggle上的分數：private 0.80424　public 0.80448

有標點符號在kaggle上的分數：　private 0.80364　public 0.80256

可以稍微看出把所有標點符號都濾掉，可以使正確率些微上升，顯然對於rnn而言，大部分的標點符號是沒有語意的情緒是沒有甚麼幫助的，因此使得把所有標點符號去掉會使得分數些微上升，但我認為這並不代表所有的標點符號都是沒有意義的(利如驚嘆號)。(但還沒有做測試)

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: r05229016 羅章碩)

我以原本TRAIN好的模型在沒有label的data上做預測，並且把將其分數大於0.5的標記為1，小於等於0.5的標記為0，接著把全部120萬+18萬的data(validataion_set為同一組)以同一參數量去做新的一次training，卻發現其在validataion_set上的分數略微下降，而在kaggle上public的分數都是0.80xx(因為計算資源不足沒辦法再做多的測試，平台又一直排隊...)

下面第一張圖是沒有做semi-surpervised training的val_acc，第二章是有做semi-surpervised training的val_acc，發現有做semi-surpervised反而在validation_set上的表現有略微下降，或許還需要一些參數的改動或是thrshold的調整才能達到比較好的效果，又或許可以用多一點的模式做ensamble做可能會有更好的效果。