HW5

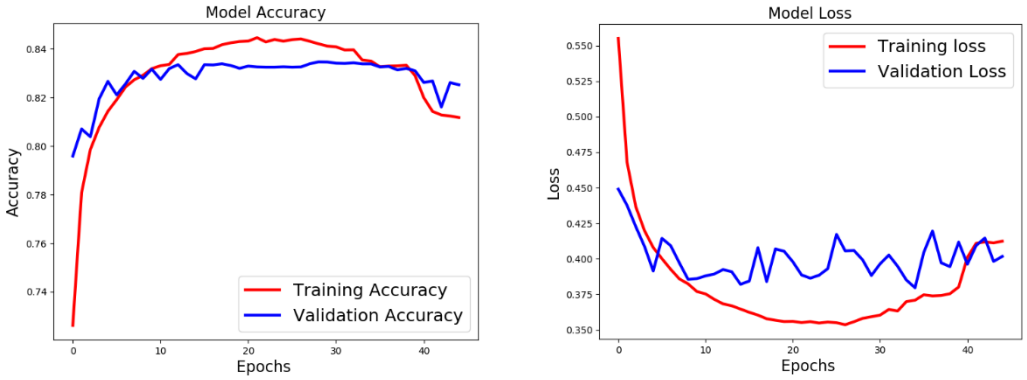
學號：b04901020 系級： 電機三 姓名：解正平

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:無)

答：

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 40, 1024)	5246976
gru_1 (GRU)	(None, 40, 1024)	6294528
gru_2 (GRU)	(None, 40, 512)	2360832
gru_3 (GRU)	(None, 256)	590592
dense_1 (Dense)	(None, 1024)	263168
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 512)	524800
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 256)	131328
dense_4 (Dense)	(None, 1)	257

Total params: 15,412,481
Trainable params: 15,412,481
Non-trainable params: 0



RNN 架構除了使用 LSTM 以外，我還有使用參數較少的 GRU layer，其中我疊了四層 RNN、四層 DNN，RNN units 大小分別為 1024,1024,512,256，DNN units 大小為 1024,512,256,1，另外我使用 Dropout=0.5，activation = 'selu'，optimizer='Adam，loss function='binary_crossentropy。

整個 model 設計除了 RNN 以外我還有使用 gensim 的 Word2Vec 來達到 wordembedding 的效果，其參數 size=256，min_count=5，也是就將整個 data 中的每個句子出現超過 5 次以上的單字才會保留起來做訓練，並將這些單字轉為長度 256 的 vector，使每個單字之間有一定的關聯，另外 word_to_sequence 我將參數 max_length 設為 40，也就是說每次 input 句子會湊到 40 個字的句子，如果不夠就會補 0，如果太長會捨去前後的字。

Public score	Private score
0.83186	0.83146

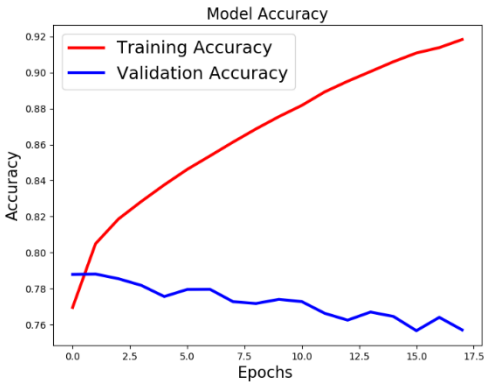
2. (1%) 請說明你實作的 BOW model，其模型架構、訓練過程和準確率為何？ (Collaborators:)

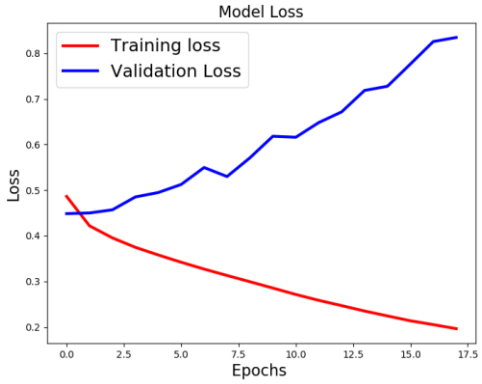
答：

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 1024)	20481024
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 512)	524800
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 256)	131328
dense_4 (Dense)	(None, 1)	257

Total params: 21,137,409
Trainable params: 21,137,409
Non-trainable params: 0

使用 text_to_matrix()來實作 bow model，但因為我設定字典大小 num_words 為 20000，所以 input 進來會有大量參數，需要對 20000 個字的字典去看有哪些字。接著我疊四層 DNN layer，units 大小分為 1024,512,256,1，另外我使用 activation = ‘selu’，loss function=’binary_crossentropy，optimizer=’Adam，Dropout=0.5。





觀察訓練過程，發現大概從第二個 epoch 開始，就有 overfitting 的現象，不但 val_acc 下降，val_loss 也上升很多，即使 DNN training model 與第一題相同但因為無法考慮文法及前後詞的順序，相對整體 accuracy 比 RNN 來的低。

Public score	Private score
0.78959	0.79165

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：

	RNN	BOW
today is a good day, but it is hot	0.2840781	0.5084212
today is hot, but it is a good day	0.9746355	0.5084212

從 RNN model 觀察結果發現，第二句"today is hot, but it is a good day" 具有相當 positivie 的分數，推測是因為後面子句帶有肯定的 good，而且接在 but 這個轉折語氣後方，更能凸顯前後差異，因為是後者 positive 夠高才能與前者子句作為轉折；相反的，"today is a good day, but it is hot" 雖然也是轉折語氣，但是變成前者子句帶有肯定的 good，而後半段轉折會使整個分數偏向 negative，但機器判斷時可能會因為整句話有出現 good 這個詞使得分數並沒有非常極端。從上述其實可以推測，機器有學到人類語句後半段子句通常比較重要，連結詞後的從屬子句會較為重要，如同本舉例轉折語氣後的句子影響分數。

另一方面，BOW 因為不會考慮句子的前後順序，所以對機器來說，這兩個句子是相同的 input 因此會得到相同的分數，但相對來說較難判斷語句的向性，不會有較極端的分數出現，多圍繞在 0.5 左右。

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)
答：
沒標點符號：
`filters='!"#$%&()*+,./:;<=>?@[\]^_`{|}~\t\n'`
有標點符號：
`filters=''`
有包含標點的 tokenize 方式 accuracy 較高，推測因為某些標點其實會使語氣改變或是增強情緒，比如說“！“,“？“等等都是表達情緒的關鍵。

	Public score	Private score
沒標點	0.82522	0.82421
有標點	0.83299	0.82982

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：
我標記 label 的方式是將 semi data 透過我的 model predict 出來的結果設 threshold = 0.05，就是取 predict 的值比 0.95 大的話標記為 1，比 0.05 小的話標記為 0，再將這些標完後的 data 加入 training data 中，總共跑了 7 次 iteration。

Iteration	label data	non-labeldata	val_acc
0	0	1178614	
1	214294	964320	0.7847
2	464547	714067	0.7949
3	576373	602241	0.7979
4	685771	492843	0.8045
5	755489	423125	0.8100
6	804373	374241	0.8097
7	832540	346074	0.8115

	Public score	Private score
沒使用 semisupervisied	0.81917	0.81836
有使用 semisupervisied	0.81262	0.81052

使用 semisupervisied 的準確率反而下降，推測是因為這些 data 沒有經過很好的 preprocess，導致 model 較難 train 的好，為了增加更多 data 但也不要拿不好的 data，準確率下降大約 0.7%。