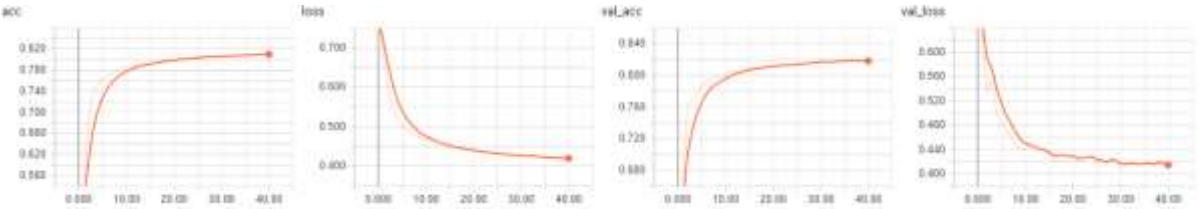
學號：R06725035 **系級**： 資管所碩一 **姓名**：陳廷易

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: 參閱 reference 中所 google 到其他網站或作者所分享的寫法、參照小老師手把手做法以及參考助教 sample code)*

答：在前處理方面，於標點符號部分會去重複，如：!!!變為!，且僅會留下問號(?)驚嘆號(!)及句點(.)。而其他部分也僅會留下文字與數字，並依照 text 句子內容，將一些較為口語化的文字進行修正，如:luv 轉為 love、2u 轉為 to you 等等。而重複字元或是 typo 的部分也會進行修正或轉換為單一字母，如：sadddd 轉換為 sad、wierd 轉換為 weird 等等。接下來，會利用 gensim.parsing.porter.PorterStemmer 進行 stemming，將一些複數結尾或第三人稱結尾的詞去除，在保留單字完整的資訊下，減少 token 數量。架構方面則是採用三層的 bidirectional LSTM，接兩層全連通層，activation function 採用 selu，kernel initializer 為 lecun normal，dropout rate 為 0.7，optimizer 為 adam，loss function 為 categorical cross entropy，詳細架構如右所述。訓練過程如下所示：

Layer (type)	Output Shape	Param #
bidirectional_1 (Bidirection	(None, 40, 256)	394240
bidirectional_2 (Bidirection	(None, 40, 256)	394240
bidirectional_3 (Bidirection	(None, 256)	394240
batch_normalization_2 (Batch	(None, 256)	1024
dense_3 (Dense)	(None, 64)	16448
dropout_2 (Dropout)	(None, 64)	0
batch_normalization_3 (Batch	(None, 64)	256
dense_4 (Dense)	(None, 64)	4160
dropout_3 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 2)	130

Total params: 1,204,738
Trainable params: 1,204,098
Non-trainable params: 640



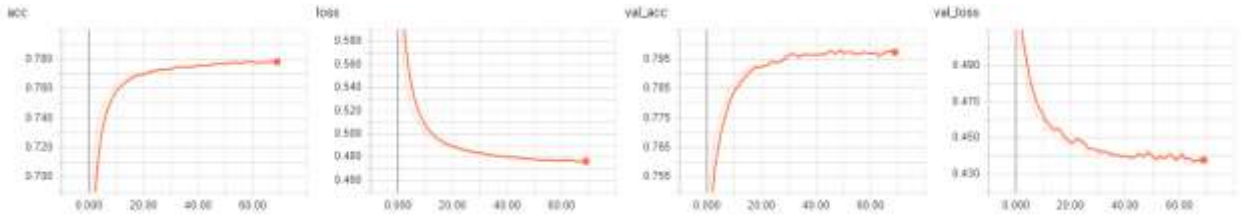
最終訓練了 private mode 與 public mode 模型，經結合上傳至 kaggle 後的訓練準確率 public score 為 0.82764，private score 為 0.82651。

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: 參閱 reference 中所 google 到其他網站或作者所分享的寫法、參照小老師手把手做法以及參考助教 sample code)*

答：首先利用 keras 套件中的 tokenizer.texts_to_matrix 將句子進行轉換，僅挑選出出現頻率最高的前 20000 個維度製作成向量。在架構方面，若增加 hidden fully-connected layer 可令準確度相當程度地提升，因此採用兩層 dense 最後輸出為 sigmoid 值，activation function 為 selu，kernel initializer 為 lecun normal，dropout rate 為 0.6，optimizer 為 adam，loss function 為 binary cross entropy。其他詳細架構如左所示，訓練過程如下所示：

Layer (type)	Output Shape	Param #
input_5 (InputLayer)	(None, 20000)	0
dense_9 (Dense)	(None, 4096)	81924096
dropout_6 (Dropout)	(None, 4096)	0
batch_normalization_3 (Batch	(None, 4096)	16384
dense_10 (Dense)	(None, 2048)	8390656
dropout_7 (Dropout)	(None, 2048)	0
dense_11 (Dense)	(None, 1)	2049

Total params: 90,333,185
Trainable params: 90,324,993
Non-trainable params: 8,192



訓練準確率上傳至 kaggle 的 public score 為 0.79602；private score 為 0.79956。

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: 參閱 reference 中所 google 到其他網站或作者所分享的寫法、參照小老師手把手做法以及參考助教 sample code）
答：

	第一句話 label	第二句話 label	Result
bag of word	0	1	[0.42217308] [0.6514508]
RNN	0	1	[[0.9331367 , 0.06686332], [0.0269928 , 0.97300714]]

根據上表可看出兩種方法以最終結果而言皆認為第一句話為負面，而第二句話為正面。然而 BOW model 對兩句話預測的結果分數相近，會造成如此差異推測有可能是因為 BOW 因基本上僅對字數進行統計做為特徵，在兩句話構成組合相似的情況下較不易區分兩句的差別；然而 RNN model 因多加考量或記憶詞彙前後的關係，因此較有機會抓到句子的重點所在，故表現較好更能區分兩句的差別。

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: 參閱 reference 中所 google 到其他網站或作者所分享的寫法、參照小老師手把手做法以及參考助教 sample code)
答：若有標點符號處理方式，則僅會留下問號(?)驚嘆號(!)及句點(.)，同時會將多個重複者移除至僅剩單個標點符號。在 public score 為 0.81986；private score 為 0.81992。

若為無標點符號的處理方式，則將所有標點符號移除，僅由空格代替。在 public score 為 0.81295；private score 為 0.81049。

兩者差異大約為 1%，有標點符號 tokenize 方式表現較好一點，或許因為在此 task 中，標點符號能代表出其他額外的資訊，尤又以驚嘆號(!)與問號(?)更能一定程度地表達情緒，因此能對預測結果造成不同影響。

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: 參閱 reference 中所 google 到其他網站或作者所分享的寫法、參照小老師手把手做法以及參考助教 sample code)
答：在 semi-supervised training 中，首先利用有 label 的 data，訓練了四個 model(架構基本上皆與 1.相同，僅所切的 training data 與 validation data 不同)令各自之 validation 準確率皆能達到 0.8 以上。接著將四個 model predict 沒有 label 的 data(training_nolabel 以及 testing_data)，將四個 model 所預測出來的分數進行相加取平均，僅挑選出 predict 正面情緒分數超過 0.95 或小於 0.05 的 data。利用原本 training data 加上自己 label 過後且超過 threshold 的 data 進行 fine-tune，選擇較小的 learning rate 並且固定第一層的參數，最後 predict 出 testing data 上傳至 kaggle 得到的準確率為：0.82002 與 0.82030。相較於僅利用 label data 所訓練的準確率為：0.81082

與 0.81280。

可見在此 task 中，若透過 ensemble 方法進行 predict 並設立較嚴格的 threshold，猶如 data augmentation 效果，可些微提升準確率。

Code Reference：

- https://flyyufelix.github.io/2016/10/08/fine-tuning-in-keras-part2.html
- https://ntumlta.github.io/2017fall-ml-hw4/RNN_model.html
- https://www.kaggle.com/githubsearch/sentiment-classification-of-tweets
- https://www.kaggle.com/kazanova/sentiment140/kernels
- https://www.douban.com/note/136065595/
- https://tw.answers.yahoo.com/question/index?qid=20061211000010KK04746
- https://www.douban.com/group/topic/7403219/
- https://www.ptt.cc/man/Eng-Class/D5D2/D87/M.1145065626.A.2AE.html
- https://wenku.baidu.com/view/622aaa8bcc22bcd126ff0cc1.html