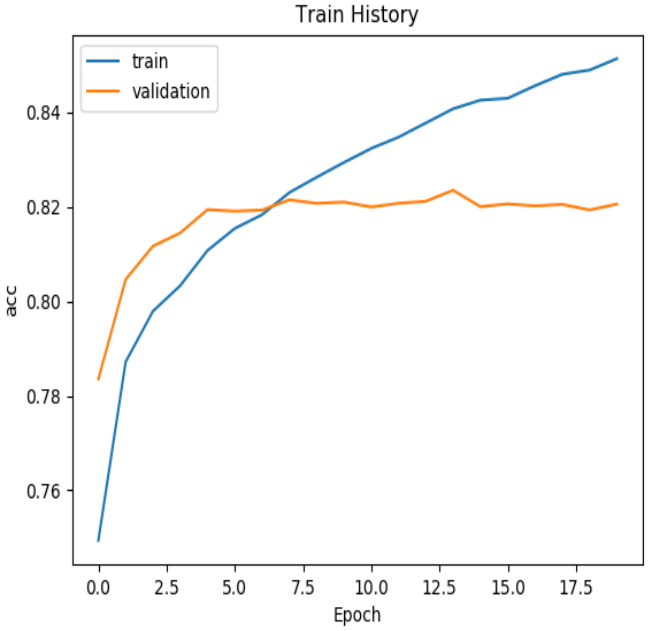
學號：B04901056 系級： 電機三 姓名：張承洋

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？
   我的 RNN Model 作法是利用 Word2Vec 將每一個字轉成 1*100 的向量，再將一句話中每個字對應到的向量排成一個二維陣列，我預設一句話有 25 個字(其餘補 0)，因此每個 sample 會是一個 25*100 的二維陣列。

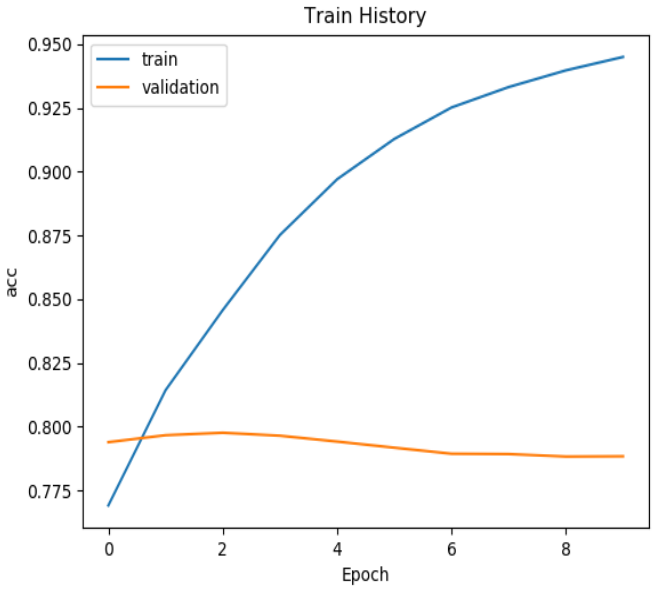


Layer (type)	Output Shape	Param #
bidirectional_1 (Bidirection	(None, 25, 512)	548352
bidirectional_2 (Bidirection	(None, 25, 256)	492288
bidirectional_3 (Bidirection	(None, 128)	123264
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 256)	33024
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
dropout_3 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
dropout_4 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 1)	65

Total params: 1,238,145
Trainable params: 1,238,145
Non-trainable params: 0

```
model = Sequential()
model.add(Bidirectional(GRU(256, recurrent_dropout=0.25, dropout=0.25, activation='tanh', return_sequences=True ), input_shape=(25,100)))
model.add(Bidirectional(GRU(128, recurrent_dropout=0.25, dropout=0.25, activation='tanh', return_sequences=True )))
model.add(Bidirectional(GRU(64, recurrent_dropout=0.25, dropout=0.25, activation='tanh' )))
model.add(Dropout(0.5))
model.add(Dense(256 , activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(128 , activation = 'relu' ))
model.add(Dropout(0.4))
model.add(Dense(64 , activation = 'relu' ))
model.add(Dropout(0.4))
model.add(Dense(1,activation='sigmoid'))
adam = Adam(lr=0.001,decay=1e-5,clipvalue=0.5)
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
print(model.summary())
train_history = model.fit(train_in, labels, epochs=20, batch_size=512 , validation_data=(vi,vo))
```

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？
   我的 BOW Model 作法是利用 tokenizer 將所有字建成一個字典，再利用 texts_to_matrix function 將每一句話轉成 1*nb_word 的向量(我設定 nb_word=12000，太大的 nb_word 會讓記憶體爆掉…) 。



Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	3072256
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
dropout_3 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 1)	65

Total params: 3,113,473
Trainable params: 3,113,473
Non-trainable params: 0

```
model = Sequential()
model.add(Dense( input_dim = 12000 , units = 256 , activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(128 , activation = 'relu' ))
model.add(Dropout(0.4))
model.add(Dense(64 , activation = 'relu' ))
model.add(Dropout(0.4))
model.add(Dense(1,activation='sigmoid'))
adam = Adam(lr=0.001,decay=1e-5,clipvalue=0.5)
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
print(model.summary())
train_history = model.fit(train_in, labels, epochs=10, batch_size=128 , validation_data=(vi,vo))
```

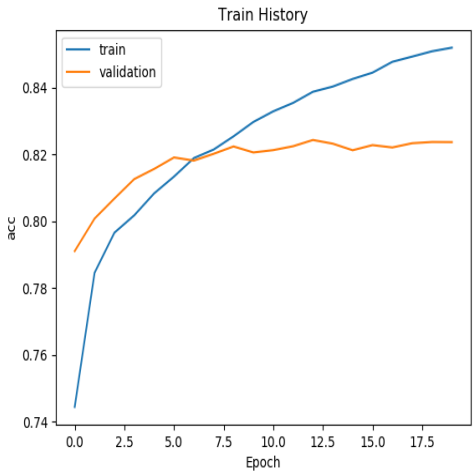
3. (1%) 請比較 bag of word 與 RNN 兩種不同 model 對於"today is a good day, but it is hot"與"today is hot, but it is a good day"這兩句的情緒分數，並討論造成差異的原因。

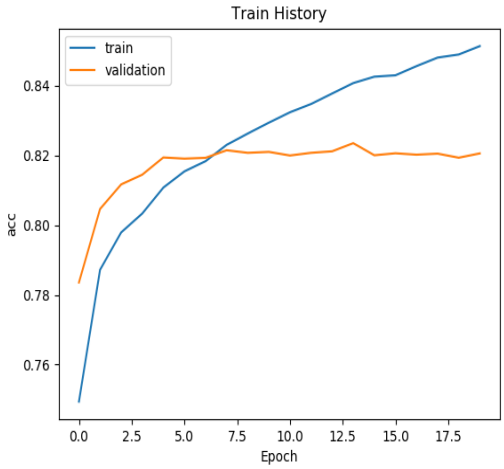
```
Bag of word :
today is a good day, but it is hot  :  [ 0.67348498]
today is hot, but it is a good day  :  [ 0.67348498]

RNN (w2v) :
today is a good day, but it is hot  :  [ 0.20078555]
today is hot, but it is a good day  :  [ 0.72455174]
```

由上圖可以很明顯看出，對 BOW model 來說這兩句話是相同的，因為這兩句話雖然排列順序不同但包含的 word 一樣，凸顯出 BOW 無法將一句話前後語意納入考量的缺點；相反地，RNN model 使用了有記憶性的 GRU，利用 recurrence 的特性判斷出這兩句話語氣上的差異，進而產生更為精確的 prediction。

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。

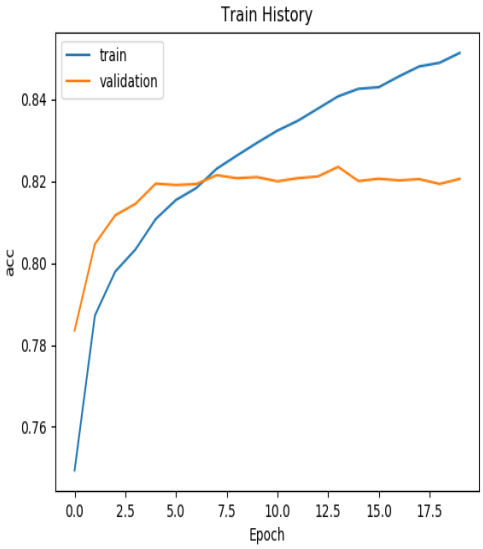
有包含標點符號(↓)　　無包含標點符號(↓)

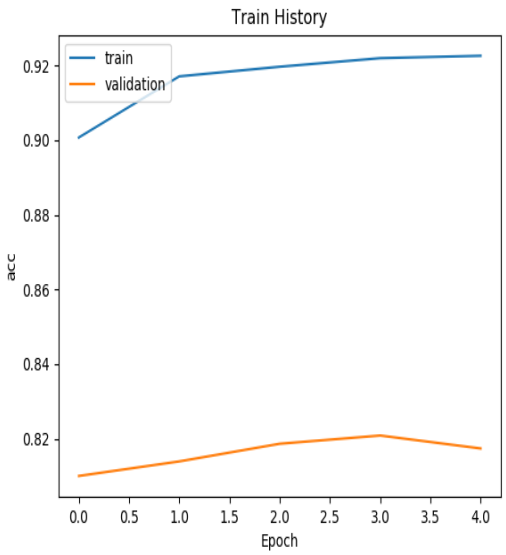




由上兩張圖可以看出，包含標點符號訓練出來的 model 較不包含標點符號的 model 有稍微高的準確率，但差異不明顯(不到 1%)。除此之外，能觀察到前者的 acc 對 epoch 關係折線圖較後者曲折 ，並且需要較多的 epoch 才會達到 accuracy 的穩定值，我認為標點符號雖然能表達一句話的情緒起伏，但同一標點符號在不同語句代表的意思可能有明顯差異，讓 model 在訓練時需要花費時間去歸納標點符號的意義，例如：同樣是驚嘆號，在以下兩句話的意義完全相反”下禮拜又有 deadline 了！”與”下禮拜 deadline 延期了！”，model 在訓練時可能需要利用 RNN 配合其他關鍵詞去修正參數。

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。(左為 supervised，右為 semi-supervised)





在訓練完 supervised model 之後，我利用此 model 對 nolabel 的 data 進行 predict，結果大於 0.96(threshold)者設定 label 1，小於 0.04(threshold)者設定 label 0，這樣的設定大約會讓 50%的 nolable data 的加入 training data，再將增量過後的 training data 丟下去 train。semi-supervised 的結果不如預期好(但我有刻意簡化 semi-supervised 的 model 複雜度，減少訓練時間)，但在 acc-epoch 折線圖方面比 supervised model 更加平滑，我推測只要將 model 作適當修正，理論上能提高 1~2%的準確度。