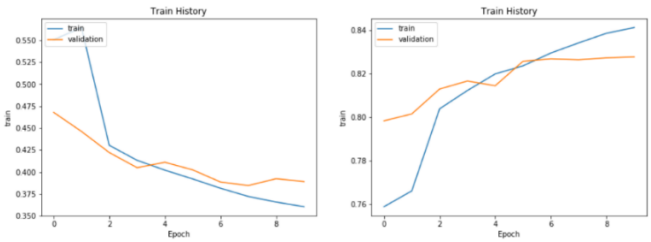
學號：b03505052 系級： 工海四 姓名：李吉昌
Collaborators:林後維 b03505040、劉祐瑄 b03303032、楊書文 b03902130
Reference: https://github.com/thtang/ML2017FALL/tree/master/hw4 (參考字串處理的方法)

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？
   模型架構：

```
model = Sequential()
embedding_layer = Embedding(len(w2v_matrix), output_dim = 300, weights=[w2v_matrix], input_length = 39, trainable=False)
model.add(embedding_layer)
model.add(Bidirectional(LSTM(128,activation="tanh",dropout=0.3,return_sequences = True, kernel_initializer='he_uniform')))
model.add(Bidirectional(LSTM(128,activation="tanh",dropout=0.3,return_sequences = False, kernel_initializer='he_uniform')))
model.add(Dense(2, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['acc'])
```

   訓練過程：

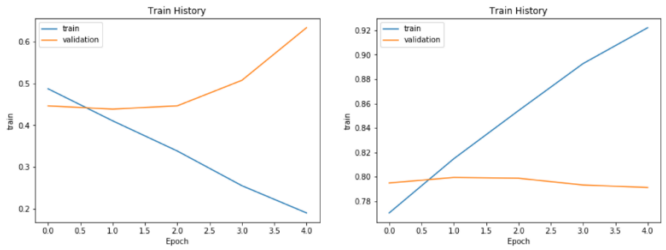


   準確率：private: 0.82826 | public: 0.82995
   Preprocessing 的部分除了把三個以上的連續字母換成一個字母以外也把 I'm 換成 Im 減少字彙量，另外發現 data 中有不少亂碼以及拉丁文，且數字也很容易被當成 to 或是其他語意混用，這些全部利用編碼將他過濾。
   實作上 Embedding Layer 的對應 matrix 是由 gensim 做 word2vec 得來的，
   嘗試過 Embedding Layer trainable 改成 True，雖然 acc 上升較快，但最後得到的最高分數較低。

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？
   模型架構：

```
model = Sequential()
model.add(Dense(input_shape = (len(tra_x[0]),) , units = 1024, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(units = 512, activation = 'relu'))
model.add(Dropout(0.4))
model.add(Dense(units = 128, activation = 'relu'))
model.add(Dropout(0.3))
model.add(Dense(units = 64, activation = 'relu'))
model.add(Dropout(0.2))
model.add(Dense(units = 32, activation = 'relu'))
model.add(Dropout(0.1))
model.add(Dense(2, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['acc'])
```

   訓練過程:



   準確率：private: 0.79844 | public: 0.79804

Preprocessing 和 RNN 相同。
實作上取至少出現過 100 次的字代表 bow 的 vector，約略四千多維，漸進減少每一層 neural 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。

RNN :

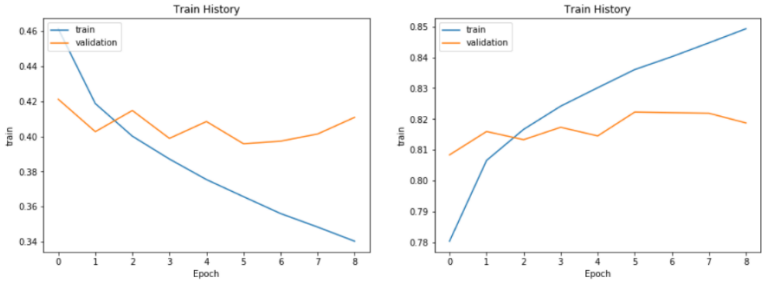
	class 0	class 1
sentence1	0.843966	0.156034
sentence2	0.010782	0.989218

BoW:

	class 0	class 1
sentence1	0.385361	0.614639
sentence2	0.331196	0.668804

BoW 判斷的結果較不確定，其差異的原因應該在於 good 和 but 在判斷 label 上佔了比較大的比重，而 RNN 除了判斷字的頻率也考慮其相對位置，也因次做出了不一樣的判斷。

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。



準確率 : private: 0.81744 | public: 0.81868
沒有標點符號的時候表現比較差，自己的猜想是標點符號像是驚嘆號可能就包含激動的語氣，或是像問號也有可能被當成質問在，情緒判斷上可以做為依據。

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。
實作 semi 的方法是將每一個 epoch 得出的 model 儲存起來，共八個 model，再將 nonlabel 的 train data 做出 8 種 predict 的結果，如果每一個 epoch 的結果高過 0.9 或是小於 0.1 就做為 semi 的 train data，這樣約略得到 20 萬筆，然後在 load 之前 train 好的 model 作為 pre-train，但是分數表現上不太有進步，如果 opitimizer 採 adam 甚至會退步，後來改成 sgd 的話，public 上的分數從 0.82995 變成 0.83044。