Machine Learning HW6 Report

學號：B0690203　系級：資工二 姓名：邱譯

1. (1%) 請說明你實作之 RNN 模型架構及使用的 word embedding 方法，回報模型的正確率並繪出訓練曲線*

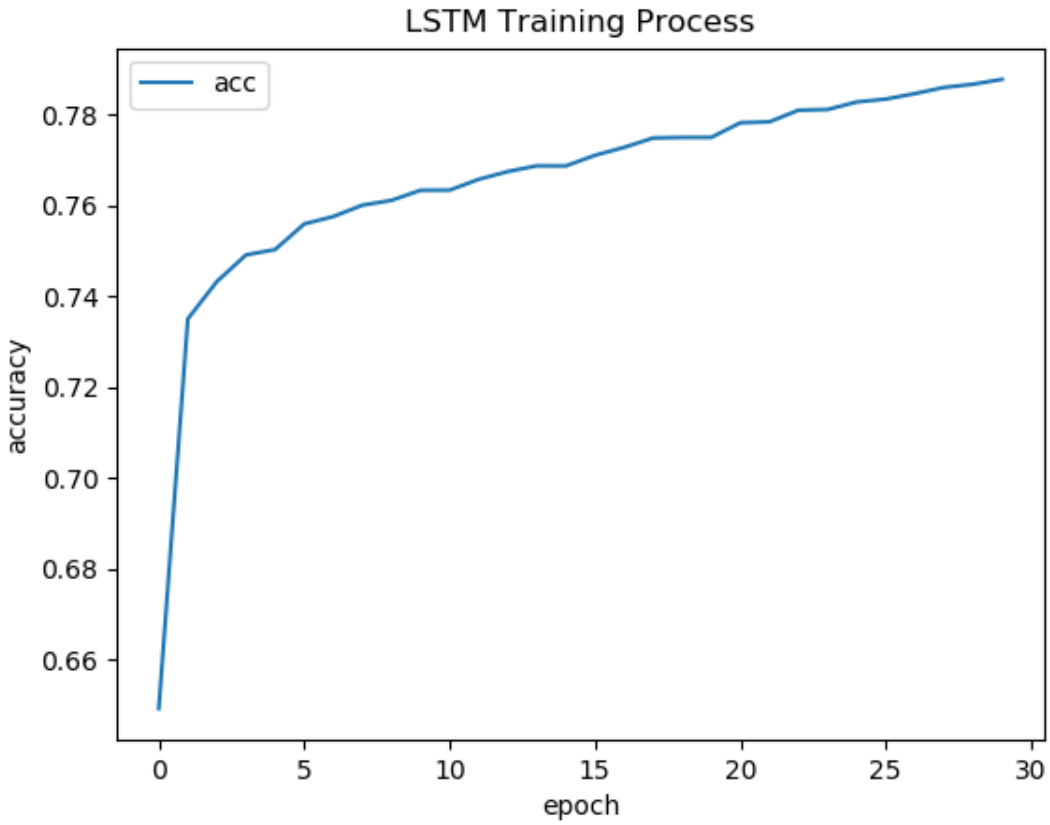
   RNN架構：

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 160, 200)          30200
_________________________________________________________________
dropout_1 (Dropout)          (None, 160, 200)          0
_________________________________________________________________
dense_2 (Dense)              (None, 160, 200)          40200
_________________________________________________________________
dropout_2 (Dropout)          (None, 160, 200)          0
_________________________________________________________________
lstm_1 (LSTM)                (None, 100)               120400
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 101
=================================================================
Total params: 190,901
Trainable params: 190,901
Non-trainable params: 0
```

   word embedding 方法：

   先利用jieba對句子做斷詞，再使用gensim的word2vec將詞轉為vector。

   訓練曲線：



正確率：

private: 0.75420

public: 0.76570

2. (1%) 請實作 BOW+DNN 模型，敘述你的模型架構，回報模型的正確率並繪出訓練曲線*。

模型架構：

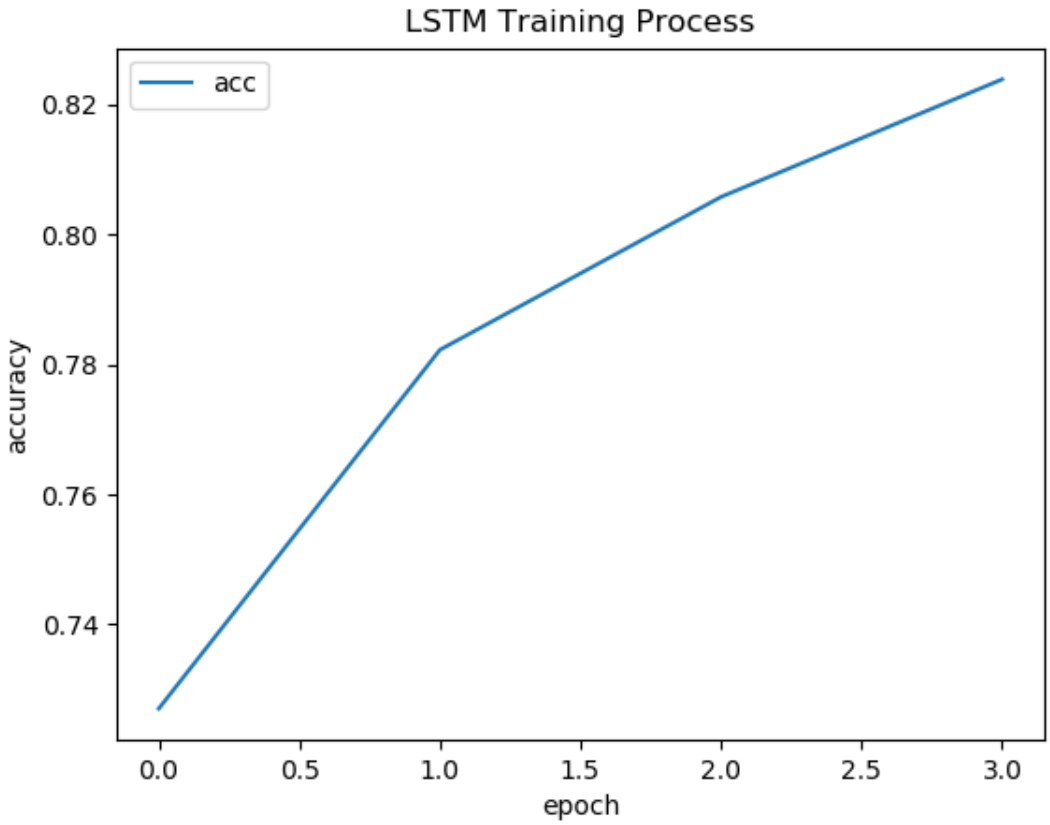
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 200)               3875800
_________________________________________________________________
dropout_1 (Dropout)          (None, 200)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 201
=================================================================
Total params: 3,876,001
Trainable params: 3,876,001
Non-trainable params: 0
```

正確率：

private: 0.75220

public: 0.75400

訓練曲線：



3. (1%) 請敘述你如何 improve performance（preprocess, embedding, 架構等），並解釋為何這些做法可以使模型進步。

   一開始我做完word2vec後直接丟進“一層LSTM接一層Dense”的model做訓練，正確率沒有很好，差不多過simple baseline，之後我在LSTM前先接了兩層Dense，正確率上升非常多，我認為是因為word2vec轉換出來的vector只代表了相似詞會有相似的vector，但這個vector與label並沒有關係，因此如果將句子的vector先進入Dense層，應能將原本的vector轉換為與label有關聯的vector，再進入LSTM，便能得到較好的正確率。

4. (1%) 請比較不做斷詞 (e.g., 以字為單位) 與有做斷詞，兩種方法實作出來的效果差異，並解釋為何有此差別。

   不做斷詞：private: 0.74980 public: 0.75220

   做斷詞：private: 0.75420 public: 0.76570

   不做斷詞的正確率低於有做斷詞的，我認為是因為有些字詞單一來看可能會是負面的，但跟其他自加在一起就並非是負面的，例如：“幹”這個字單一時為負面，但比如說用法為“幹嘛”時，即非負面。

5. (1%) 請比較 RNN 與 BOW 兩種不同 model 對於 "在說別人白痴之前，先想想自己"與"在說別人之前先想想自己，白痴" 這兩句話的分數（model output），並討論造成差異的原因。

   RNN:

   "在說別人白痴之前，先想想自己" : 0.47157198

   "在說別人之前先想想自己，白痴" : 0.55214006

   BOW:

   "在說別人白痴之前，先想想自己" : 0.653315

   "在說別人之前先想想自己，白痴" : 0.653315

可以看到在RNN中，model predict兩個句子的結果有所不同，因為文句的順序會影響RNN預測。

而在BOW中，model predict兩個句子的結果完全一樣，因為BOW完全不考慮順序，只考慮句子中包含的字詞。