Machine Learning HW6 Report

學號：b06705001　　系級：資管二

姓名：楊力行

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:

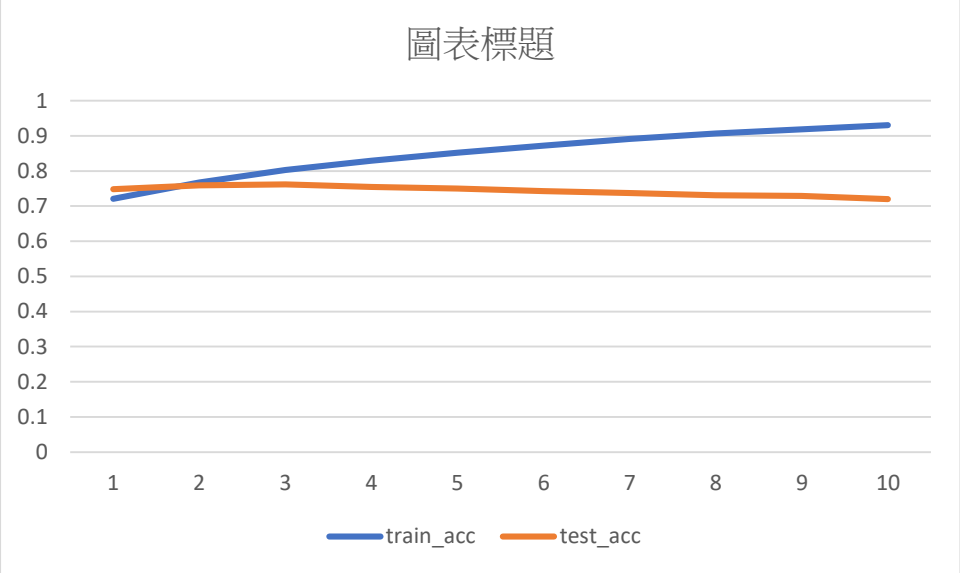
   一層 trainable 的 embedding_layer，一層 128 個 units 的 LSTM layer，dropout 為 0.3，一層 1 個 units 的 sigmoid 的輸出層，loss="binary_crossentropy", optimizer="adam"，batch 128,epoch =1

   word embedding :

   word2vec.Word2Vec(newdata,size=1200,min_count=5,sg=0)

   sg=0 用的方法是 CBOW 每個詞的詞向量有 1200 維，在全部 training data 中出現次數<5 次的詞不紀錄

   acc :　0.76430 0.76430



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*。

   BOW 中取最常出現的前一萬詞

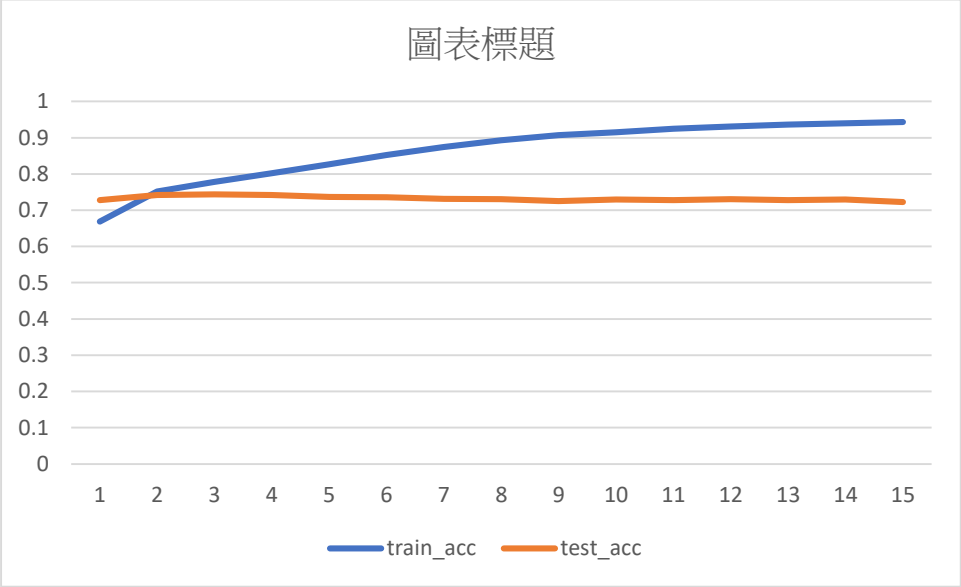
   模型是 2 層 256units relu 的 dense 層接上 1 層 1 unit 的 sigmoid 輸出層

loss="binary_crossentropy", optimizer="adam"

batch 為 3000 15 個 epoch

acc : 0.53969 0.54210

acc 遠低於 val_acc 有可能是因為我在 train 的時候沒有把 test 的句子也丟進 BOW 中來計算最常出現的，而是只用 train data 的。



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。

   先把資料的 119017 之後的資料刪掉，因為是重複的，這會導致模型對這句話過度重視。

再來是讓我準確上升很多的方法，就是增加每個詞 embedding 的詞向量，因為詞向量維度的增加可以更準確和詳盡的描述詞與其他詞的聯繫以及該詞的意義，所以可以更準確的判斷。

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。

   有斷詞 0.76430 0.76430

   沒斷詞 0.74250 0.75040

   因為同一個字在不同的詞彙中代表的意義可能會完全不同，一個可能是正面的，但另一個可能就是負面的，沒斷詞的在這種情況就會導致模型難以判斷。

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。

   BOW 兩句分數都是 0.00430526

   RNN 則分別是[0.5518582]　[0.5996439]

   BOW 因為只看詞是否出現，和出現幾次，而不看出現的順序，所以當詞都沒變，只改動順序時分數會完全一樣。

   RNN 則不同，可以根據詞的出現順序和距離來推斷是否為仇恨，因此分數會不同，可以較為準確。