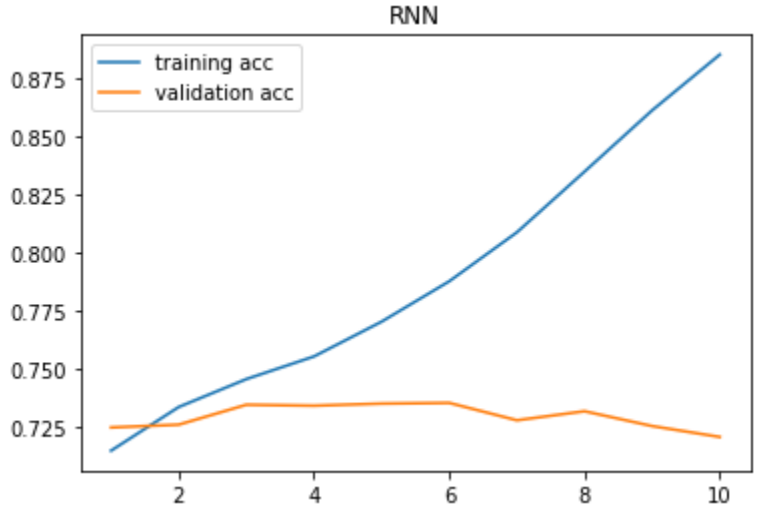
Machine Learning HW6 Report

學號： B06507007 系級：材料二 姓名:王致雄

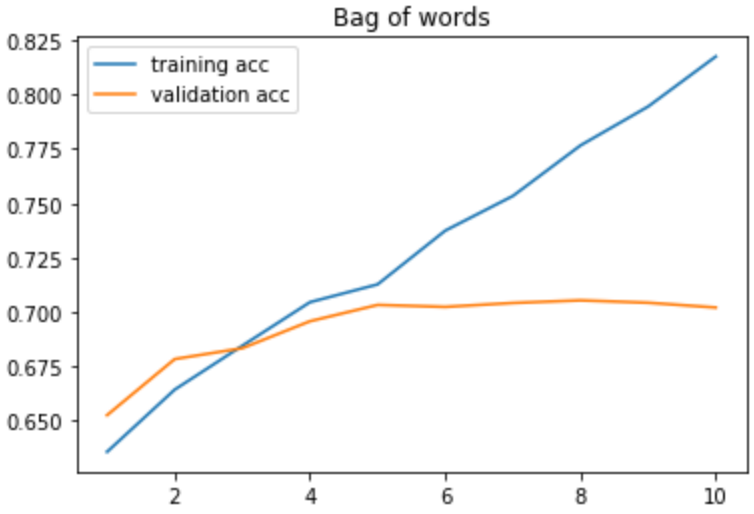
1. (1%) 請說明你實作之 RNN 模型架構及使用的 word embedding 方法，回報模型的正確率並繪出訓練曲線*

   使用 jieba 將訓練資料斷句，再使用 word2vec 將斷句後的資料訓練成 28216 維的詞向量，再放進 RNN 訓練。
   RNN 架構為 3 層 LSTM，dropout=0.4，輸出維度是 256。再將 RNN 輸出的資料放進 DNN。DNN 的架構為 256 變 128，經過 RELU()，再 dropout=0.5，最後 128 變 1，通過 sigmoid 後，大於 0.5 即為 1，小於 0.5 即為 0。
   Optimizer 為 adam，learning rate=0.001。Loss function 使用 binary cross entropy loss。Padding=30。共有 1605889 參數。
   Public:73.69% Privite:73.14% validation:73.7%



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*。

   Bag of words DNN 架構為 4 層 fully connected feed forward layer，其中第一層把 input 變為 350，第二第三層不變，第四層再 output 1 個數字。中間都有使用 Dropout=0.5
   Public:70.51% Privite:70.24% validation:70.54%

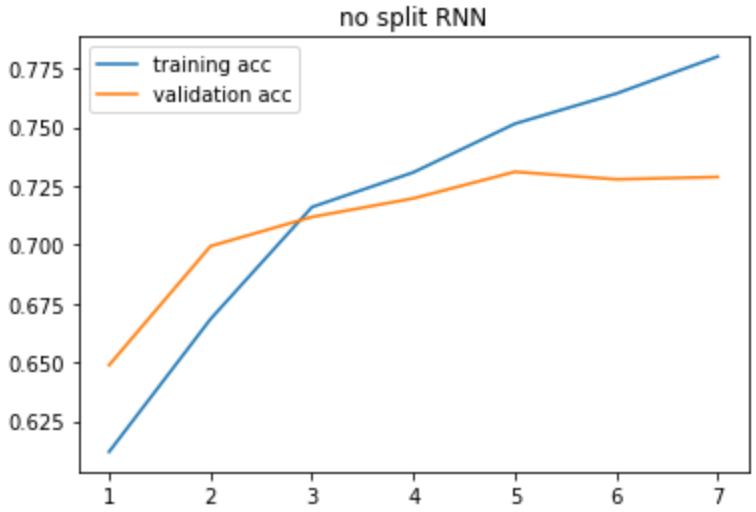


3. (1%) 請敘述你如何 improve performance（preprocess, embedding, 架構等），並解釋為何這些做法可以使模型進步。

   preprocess 的部分我有改 word2vec 的 iteration，預設是 5，改成 16 之後雖然詞向量要訓練比較久，但是對 RNN 的訓練結果也有幫助(約 0.8%)。也更架構的部分從 1 層 LSTM 改為使用 3 層 LSTM 可以更快達到 validation set 的最高值，且精準度可以提高約 1%。

   原因的話我想 preprocess 增加 iteration 的話，詞向量可以更加精準的表示詞與詞之間的關聯性，使模型較好訓練。而使架構一定程度複雜化，並使用 dropout 避免 overfit，可擴大尋找最佳模型的樣本，能有效增加精確度。
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。

   Public:72.8% Privite:72.75% validation:73.1%



首先我發現訓練出來的詞向量 vocabulary 少很多(4000，原本約 25000)，而 validation 也較慢達到高峰，也比有斷詞少了一點精確度。
我認為應該是有斷詞，比起一個把每一個字當成詞能更明確得表示語意，使訓練較為容易。

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 對第一句的判斷是 0.4338，非惡意，對第二句的判斷是 0.5772 為惡意。
BOW 對第一句的判斷是 0.6573，為惡意，對第二句的判斷是 0.6824 為惡意。
造成如此差異的地方應在於給定順序，RNN 可以一定程度判斷一個有惡意的詞(白癡)是否會使整句話變惡意，但 BOW 不考慮順序，很可能只看到白癡就判斷整句話為惡意。