Machine Learning HW6 Report

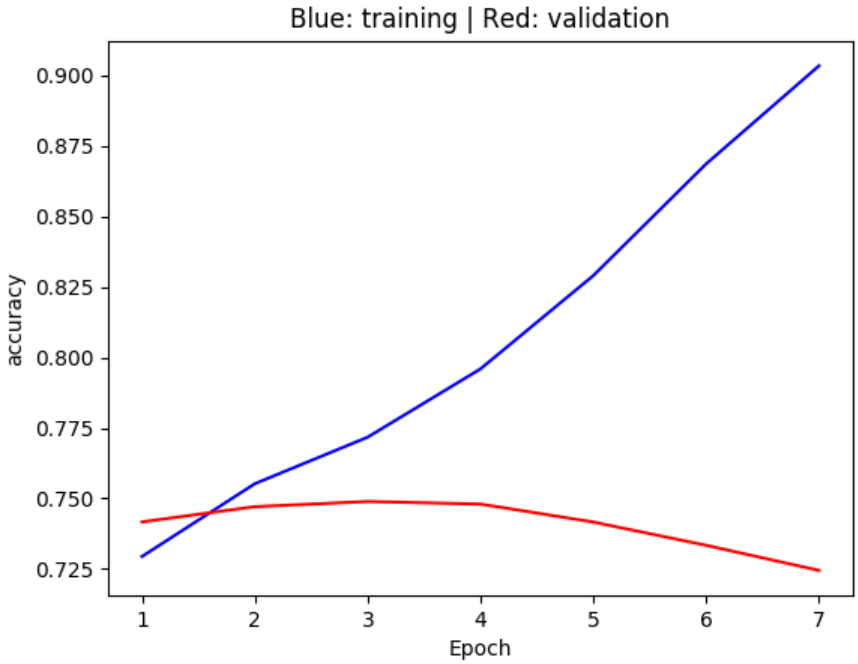
學號：B09401069　系級：電機四　姓名：林志皓

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*

word embedding:
我使用 jieba 和 gensim 的 Word2Vec 套件，先利用 jieba 把每句留言切成數個單詞，在交由 word2vec 套件學出合適的 embedding,我設定的參數是 300 維的 vector,window size = 5 (需要參考多少附近的單詞), iteration= 20 (越大有可能可以學出更好的 word embedding)

RNN 架構：
hidden size = 512, 使用一層, 使用 bidirection。input 為利用 word2vec model 轉為 300 維的向量，針對長度不一的句子，我統一取出 40 個向量，長度超過 40 的取前 40，小於 40 的則重複取至 40，以便調大 batch size 進行訓練。

訓練曲線如下圖：(正確率：0.75283)

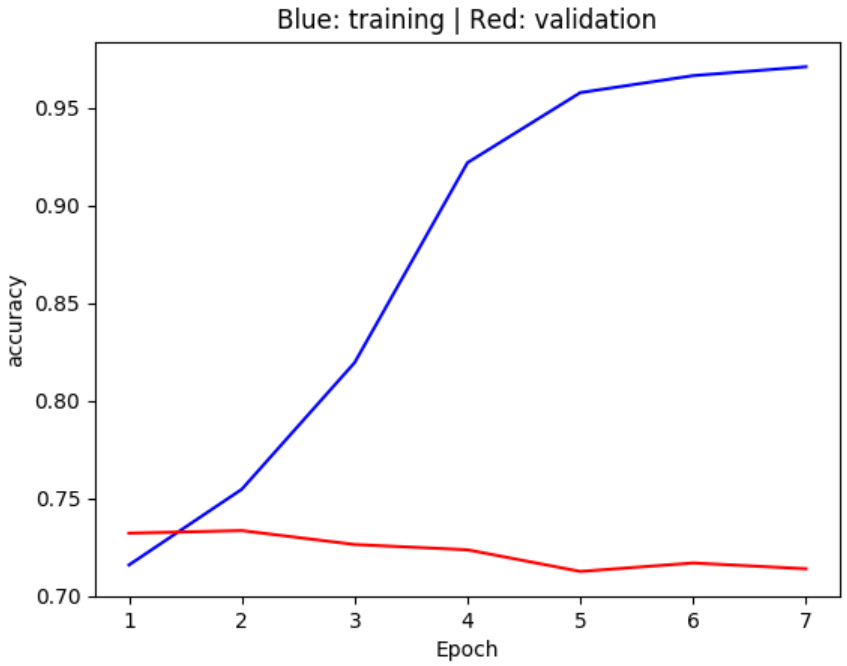


2. (1%) 請實作 BOW+DNN 模型，敘述你的模型架構，回報模型的正確率並繪出訓練曲線*。

資料處理：
我使用 jiaba 將 training data 的所有句子切成單詞，然後計算出現的次數，取次數最高的前 2048 的字作為參考的字典。對於每一句子，切成單詞後，統計字典中字詞在該句子中出現的字數，形成 2048 維的向量，以此進行訓練。

模型設計：
全為 full connected layer, 總共為 7 層，神經元的數量分別為 1024, 512, 256, 128, 64, 32, 1。沒有特別的地方。

訓練曲線如下圖：（正確率：0.73385 ）



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。

資料處理:
在 word embedding 的部份，我先分析 jieba 分詞的狀況，檢視有沒有無法被正確切出的字詞，然後增加該詞在 jieba 中的頻率，例如:"森氣氣","台男","台女","8+9"等，讓句子能被切的更漂亮;而對於許多像"B7","B25"等指稱代詞，我通一換為"B"替代，然而經過我嘔心瀝血的處理後，表現並沒有顯著進步。而調整 word2vec 的 iteration 數反而有差，增加至 20 時比起 5 可以把字詞的 word embedding 學的更好，把相關的字詞分的更近，進而增加表現。

模型設計：

我使用 LSTM 作為模型基礎，實驗各種設計，以下是有提昇的方法：
a. 增加 LSTM 的 hidden layer size -> 增強模型強度
b. bidirection -> 能學到更多字詞間的對應關係
c. 使用兩層 hidden layer -> 增強模型強度

Ensemble:
我使用 4 種不同架構進行 ensemble，每個眉形又分別使用不一樣的 sequence length(40, 50, 60, 80, 100)進行預測(訓練時皆為 40)。如此便有 4 * 5 = 20 種不一樣的預測結果，然後投票(uniformly)，產生最後的結果，如此一來確實可以讓每個資料分的更好，因為一個模型分錯的句子可能被其他模行正確分類 (這就是民主制度的優點)

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。

如果做斷詞，單一模型的 validation accuracy 大約都能到達 0.748，然而若不做斷詞，validation accuracy 大約下降 0.03~0.04 左右。我認為這是因為由單字組成的單詞所包含的意義，應大於分別單字意義相加，像是"白痴"並不能視為"白"和"癡"的意思相加，因此如此的 word embedding 會使學習更加不易，造成 accuracy 下降。

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。

在進入 sigmoid 之前的分數：
RNN -> 分別為 -0.5222 和 -0.319
BOW -> 分別為 -0.09742 和-0.09417
可以明顯發現，RNN 在這兩句話有不同的分數，而 BOW 則是兩者分數差不多。
我推測因為 RNN 有考慮字詞的順序，而 BOW 只考慮字詞的組成，而這兩句話的字詞組成可以說是幾乎差不多的，因此換成 BOW 的 input 時應是非常相近的 vector,因此無法分辨不同，但若是考慮順序的 RNN，則能順利分出不同。