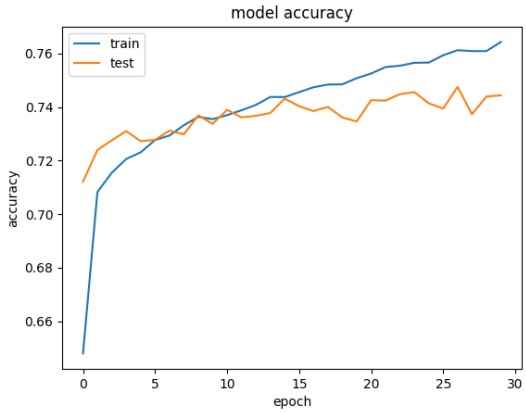
Homework4 Report

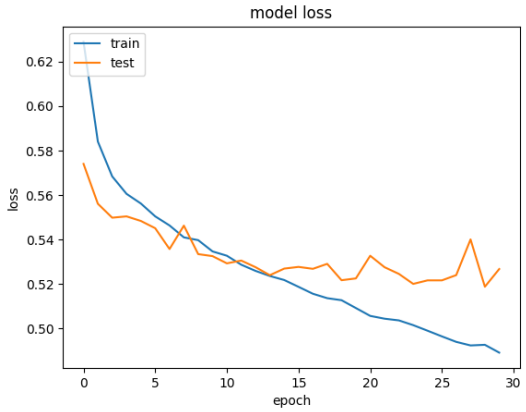
Professor Pei-Yuan Wu
EE5184 - Machine Learning

姓名：洪正皇
學號：R07922050

1. (0.5%)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。

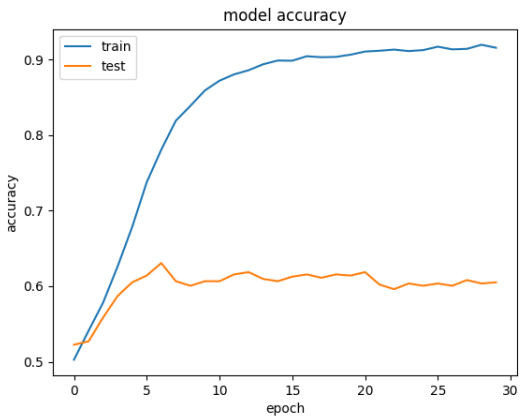
我疊了兩層 LSTM 輸出 units 依序是 256、128，且想說這次資料量 12 萬筆算滿多，在這兩層中我都用了 0.3 的 dropout，最後用兩層 DNN 中間一樣放了 0.5 的 dropout 使結果為一個值的輸出。Word embedding 是先用 jieba 將輸入切成字詞，再透過 word to vector 的方式，將每個字詞轉成一個向量，並固定每一筆輸入的字詞數，直接將一句話中的這些向量丟進 LSTM 中訓練，結果如下：

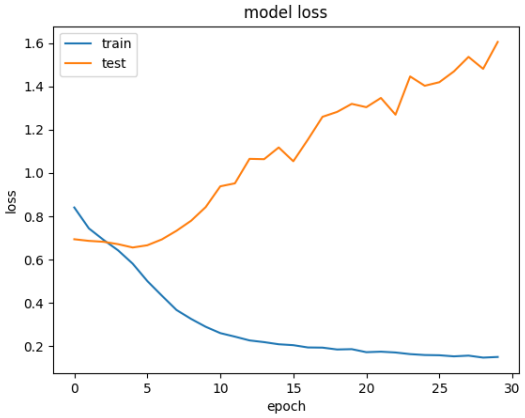




(0.5%) 請實作 BOW+DNN 模型,敘述你的模型架構, 回報正確率並繪出訓練曲線。

一樣先透過 jieba 將輸入切成字詞(words)，另外因為 BOW 所需的記憶體空間較大，因此我只使用 1 萬筆 data，在考慮需要取用哪些 words 時，我認為頻率最高的一部份詞沒有參考意義，例如：我、了、拉……等等，因此我將頻率最高的 1000 個 words 刪掉，另外出現次數小於 3 次的 words 我也不取用，之後用四層 DNN 逐漸降低維度並輸出一個數值，但即使加入了 0.5 的 dropout 看起來也有點 overfit，結果如下：

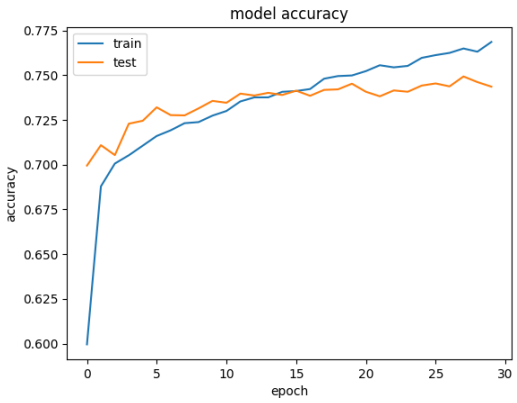


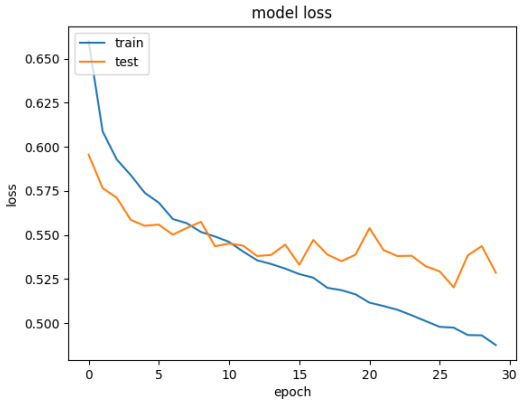


2. (1%) 請敘述你如何 improve performance(preprocess, embedding, 架構等), 並解釋為何這些做法可以使模型進步。

一開始使用一層 LSTM，我的分數始終差 strong baseline 一點點，而如同上一題提到的，我認為 12 萬筆的資料算多，因此直接把 model 變複雜看能不能提高準確率，以及經過助教的手把手教學，我就疊了兩層 LSTM，但也有點擔心 overfit 所以加了 dropout，果然複雜的 model 在 data 足夠時可以表現好一點。在 preprocess 的部分，雖然我沒有多加著墨，不過我想可以再參考顏文字及表情符號來提高準確率，且我發現讀檔時我寫的不夠好，改進的部分還可以濾掉每一行後面的換行符號。

3. (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。





以上圖片是不做斷詞時的結果，有做斷詞的結果如第一題所示 LSTM，loss 值的下降速率有沒有斷詞都差不多，我想是因為相同 model 的關係，只是輸入有些不同，因此參數更新的速率都差不多。有趣的是在前 30 epochs 這邊，不做斷詞的 accuracy 比有做斷詞的情況下高了 0.01，雖然差異不是非常的大，但我猜測是因為，不做斷詞的情況下，word to vector 的 words 較少，例如：我、我們、你、你們，不做斷詞的話就會被當成三個字而已，有做斷詞的情況就是四種不同的字詞，因此不做斷詞的情況下的輸入相對比較簡單，可以比較快訓練起來，但因為相對簡單，此 model 能做到的極限也會較低，礙於時間不夠，不然我想繼續訓練到 100 或更高的 epochs 時，有斷詞的 model 準確率應該會在某個 epoch 超過沒斷詞的 model。

4. (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。

	在說別人白痴之前,先想想自己	在說別人之前先想想自己,白痴
RNN	0.5032	0.4922
BOW	0.6190	0.6190

這兩句話透過 jieba 斷詞後的字詞都相同，但順序不同，在 RNN 中會被順序影響預測結果，因此這兩句話的分數有些微差異，而我的 model 可能有點失敗，RNN 中的分數看起來似乎反了。

在 bag of words 中，只記錄字詞出現幾次，而不紀錄順序，因此分數都相同，而我認為”白癡”這個字詞在 BOW 中影響滿大的，因此判斷為惡意言論的分數很高。

因此這兩者最主要的差別是 RNN 與 DNN 會不會去關心字詞出現的順序，而 RNN 考慮前後文的關係，雖然句子中有一個負面的”白癡”出現，但配合前後文之後被 model 判斷較不是惡意言論，不像 DNN 容易被單一字詞影響結果。