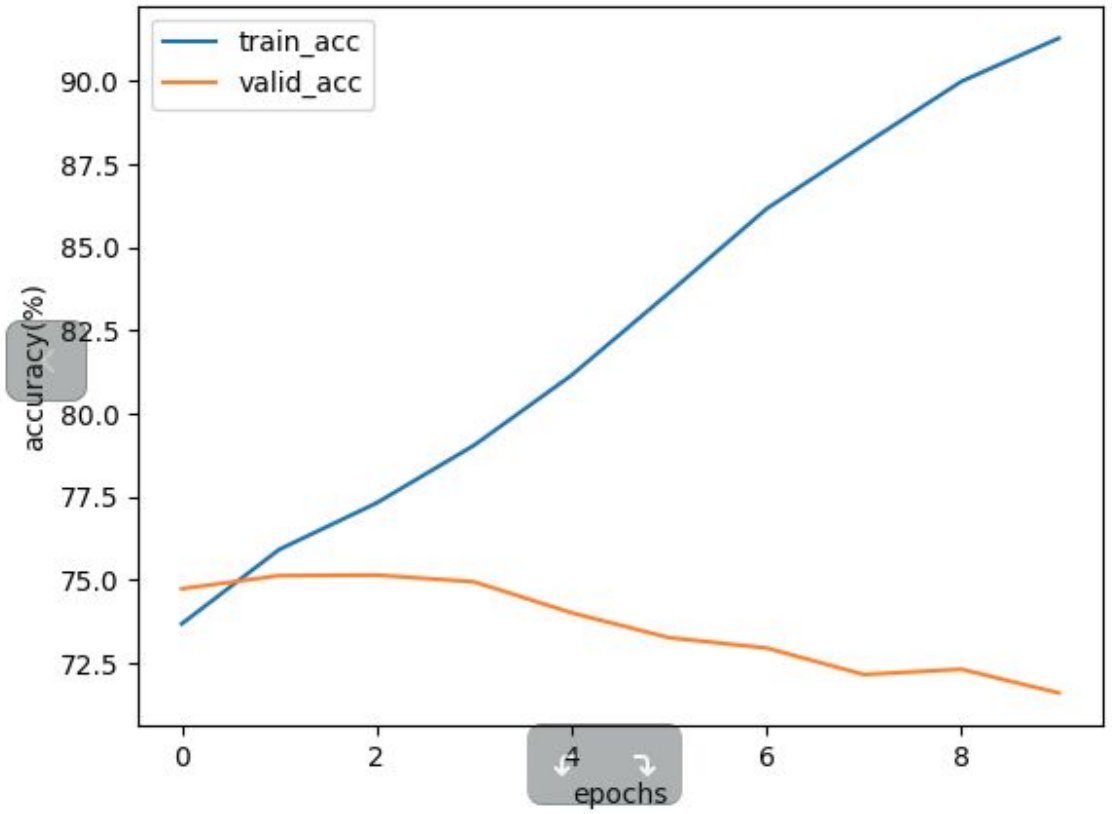
Machine Learning HW6 Report

學號：b04507025　系級：電機四　　姓名：韓秉勳

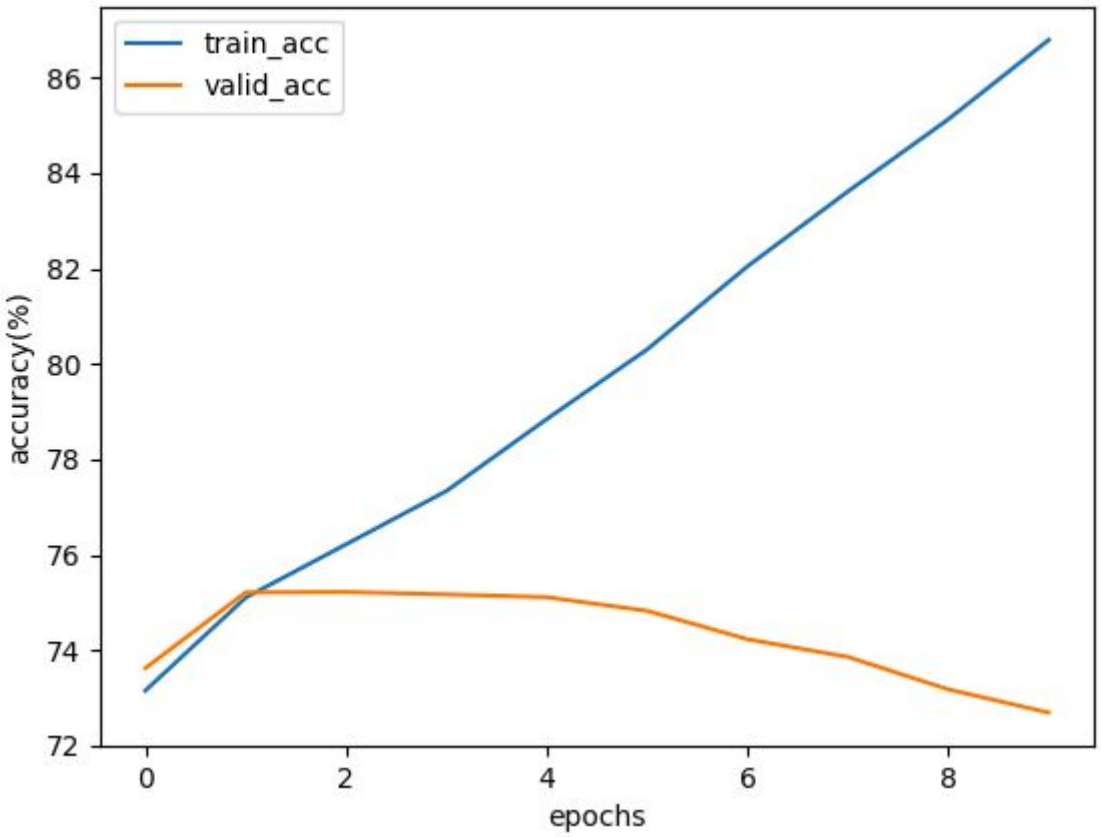
1. (1%) 請說明你實作之 RNN 模型架構及使用的 word embedding 方法，回報模型的正確率並繪出訓練曲線*

   我RNN先用jieba切詞，再用word2vec train embedding 的向量（一個詞256維）且統計句長後，最後句子取40個詞為統一長度，並再加入<unk> <pad>的向量以做分句用，而後再用了2層的bidirectional LSTM，hidden layer = 256維來train所需的model，並用maxpooling與averagepooling取代只取output最後一維的sequence，再丟到DNN裡面展開。本身validation accuracy最高的大概能到76.1%左右。而後我再用ensemble10個model，在kaggle上有到0.7618的準確率。由training曲線可以看到大約在2-3 epoch就已經到達validation最高點，之後就overfitting了，故epoch太多反而效果不佳。



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*。

   我的會把所有詞都先算過總共出現幾次，再把詞全部換成出現次數來當training data丟進DNN內，並用四層dnn降到1維(256->256->128->1)。validation accuracy最高到75.218，但也可以看到在更早的epoch就有overfit的情況。最後kaggle accuracy 在0.74680，比RNN結果差，可能是因為bag of word只看字有沒有出現不管順序，故惡意字出現但句意非惡意的情況可能會導致辨識錯誤。



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。
   a. preprocess ： 我原本只取全部的中文去train，但因符號也有負面含意，取掉反而成效不佳，故後來還是要把全部符號一起切比較有效果。
   b. embedding: word2vec可以設定重複幾個字才能放到vector字典裡，我試出至少字出現3次效果較好。另外seq長度取40以在padding與刪去多餘字長做出取捨，不然若句子過長會吃到太多padding的無用資訊。另外若unknown與padding的向量也要分別設成uniform和zero丟到word2vec字典，不能合在一起，不然可能會斷句錯誤。
   c. 架構： RNN lstm,gru,或lstm,gru mix其實沒差太多，但有把output取maxpooling與avearge pooling能比取output最後一維更能代表整個句子(ie. seq中最大與最平均的特徵)，故我取這兩個vector cat起來送到DNN裡，此舉才真的有比較顯著的準確率提昇。

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。

有做斷詞時，由於詞才是中文意思的基本單位，故有斷詞時結果好很多，若只有一個一個字當作embedding，可能因一個詞中一個字造成詞意誤判，使model無法辨識正確語意。

	有斷詞(%)	沒斷詞(%)
kaggle accuracy	76.000	73.990
validation accuracy	76.126	75.277

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 output	0.5382	0.6235
BOW output	1	1

使用 BOW 表示時，兩句對於 BOW 的 model 來說是一樣的，且因為有"白痴"這個詞彙，讓 model 的 output 直接變成 1。 反觀 RNN 的 model，雖然兩句包含的詞彙相同，但由於順序不同，因此 output 的大小也不 相同，雖然兩句的 output 皆大於 0.5，但可由此看出 RNN 會根據詞彙的順序不同來判斷不同語句的意思。但在網路留言中，由於句子短且不是每個留言都遵循完整文法規則，有幾個惡意字出現基本上就可以斷定是惡意的，故BOW表現不一定比RNN差。