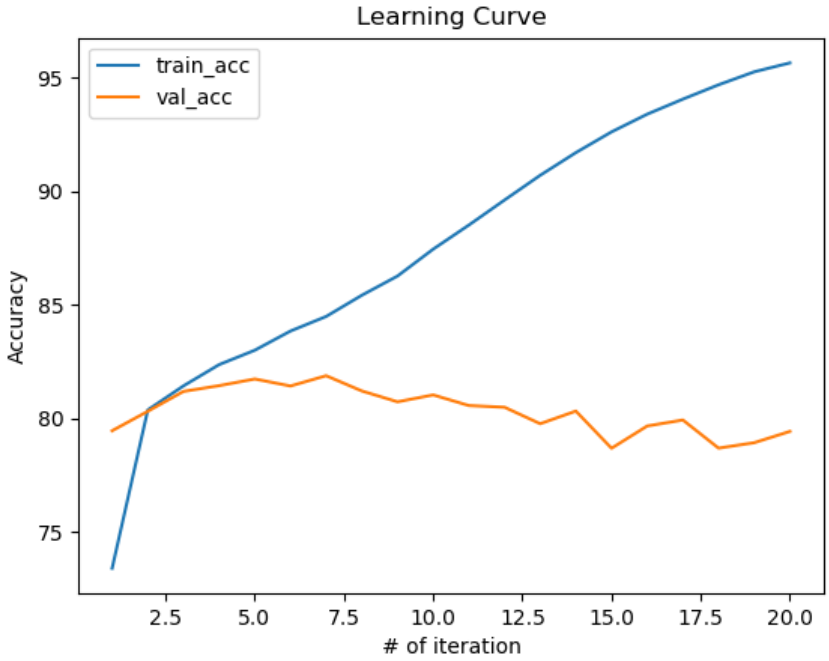
Machine Learning HW4

學號:B06902060　系級:資工三　姓名:鄒宗霖

1. (1%) 請說明你實作的 RNN 的模型架構、word embedding 方法、訓練過程 (learning curve) 和準確率為何？(盡量是過 public strong baseline 的 model)



在我實作的 RNN 模型架構中，input sentences 會先經過 preprocess，把參差不齊的句子都變成長度為 32 的 tensor，代表每個句子都是由 32 個單字組成，接著把 sentences 中的每一個單字依序通過 embedding layer，得到一個代表那個單字的 vector (dim = 250)，接著通過 LSTM (hidden_dim = 150 , num_layers = 1) 得到一個代表那句 sentence 的 vector (dim = 150)，最後再餵進一層 Linear 以及 sigmoid function (另外也實作了 self training，# unlabeled data = 200000)。Word embedding 的方法為利用 gensim.models 裡面的 Word2vec 函式把每一個單字轉成代表該單字的 vector，由於訓練資料量蠻大的，模型在第一個 # of iteration 更新完參數後 training set 以及 validation set 都有著不錯的表現，隨後 training accuracy 不斷上升、validation accuracy 趨於穩定，該模型準確率為 0.82347 (Kaggle 上的成績)。

2. (2%) 請比較 BOW + DNN 與 RNN 兩種不同 model 對於 "today is a good day, but it is hot" 與 "today is hot, but it is a good day" 這兩句的分數 (過 softmax 後的數值)，並討論造成差異的原因。

	today is a good day, but it is hot	**today is hot, but it is a good day**
BOW	0.5664	0.5664
LSTM	0.0763	0.9871

上表為 "today is a good day, but it is hot" 與 "today is hot, but it is a good day" 這兩個句子分別通過 BOW + DNN model 以及 LSTM model 的分數，可以看到這兩個句子在 BOW + DNN model 都得到了 0.5664，推測是因為 BOW 實作的方式不考慮句子裡面單字出現的先後順序，導致代表這兩個句子的 vector 一模一樣，此外，這兩個句子都是轉折句，BOW 自然分不出轉折句

是帶有正面還是負面意義，得到的分數 confidence 較低；然而 LSTM model 考慮了句子裡面單字出現的先後順序，因此得到了 confidence 較高且較準確的結果。

3. (1%) 請敘述你如何 improve performance (preprocess、embedding、架構等等)，並解釋為何這些做法可以使模型進步，並列出準確率與 improve 前的差異。(semi supervised 的部分請在下題回答)

   首先，我在 preprocess 的時候把 sen_len 調成 32，模型準確率從 0.80365 進步到 0.82188，可能的原因是在 data set 中有些句子較長，關鍵字出現的地方比較後面，原本 sen_len = 20 無法涵蓋到某些句子的關鍵字 ；再來我實作了 semi-supervised (self training , # unlabeled data = 200000)，模型準確率從 0.82188 進步到 0.82347，可能的原因在第四題中回答。

4. (2%) 請描述你的 semi-supervised 方法是如何標記 label，並比較有無 semi-supervised training 對準確率的影響並試著探討原因 (因為 semi-supervise learning 在 labeled training data 數量較少時，比較能夠發揮作用，所以在實作本題時，建議把有 label 的 training data 從 20 萬筆減少到 2 萬筆以下，在這樣的實驗設定下，比較容易觀察到 semi-supervise learning 所帶來的幫助)。

	iter_2	**iter_4**	**iter_6**	**iter_8**	**iter_10**
semi-supervise	57.1/61.5	75.4/73.8	96.0/75.5	96.9/76.1	97.7/76.4
supervise	59.1/70.3	77.2/75.8	78.7/75.7	80.5/75.3	82.4/75.1

   上表為 training set / validation set 在不同 iteration 中 semi-supervise learning 以及 supervise learning 的分數 (2 萬筆 labeled training data , 20 萬筆 unlabeled training data)，若是 unlabeled data 通過 sigmoid function 後得到的分數大於 0.9 就 label 為 1、小於 0.1 就 label 為 0，並把它們放入 labeled training data 參與之後 model 參數的更新。可以看到 10 個 iteration 後，semi-supervise learning model 在 training set 以及 validation set 上都有比較好的表現，可能的原因為 unlabeled data 的分布可能告訴了我們實際資料的分布應該長怎樣，雖然 unlabeled data 的 predicted label 不一定是正確的，但還是對 model 參數的更新有不少的幫助。