學號：b05705011　系級：資管四　姓名：楊子霖

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)

   word embedding 的方式是參照助教所提供的 sample code，唯一差別在於在訓練 word to vector 的 word embedding 時有涵蓋了 unlabel data。RNN 的模型架構，使用了 LSTM，embedding_dim為 250，hidden_dim 為 512，num_layers 為 1，最後在做 dnn，有開啟 dropout 設為 0.5，activate function 是使用 sigmoid，loss function 為 binary cross entropy loss。參數 epoch 為 7、learning rate 為 0.00055，拿九成的 data train 完後在 validation set 的 accuracy 達到 0.813 左右，接著再把 unlabel data 拿去做 testing，並將信心程度 >= 0.8 的資料取出再和 training data 重新一起 train 一遍，此時在 validation set 的表現提升至 0.825 左右。而此 model 在 kaggle 的 public 表現結果為 0.82242。

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。

   BOW+DNN 對於這二句所得出的結果為 0.80344425668522，而 RNN 對於這二句所得出的結果分別為 0.19415557384490967、0.9999994039535522，也就是說 BOW+DNN 判定二句結果皆為正面，而 RNN 判定結果前句為負面，後句為正面。會有此差異是因為 bag of words 只在意了出現什麼詞，而不會去考慮詞跟詞之間的前後關係，也因此二句的分析結果雷同，而 RNN 則會去考慮詞與詞之間的前後關係，也因此所得到的結果較正確。

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）

   首先是做 epoch 和 learning rate 的調整，將 epoch 增加、 learning rate 減少，發現當 epoch = 7、learning rate = 0.00055，在 validation set 的整確率最高可達 0.81，而且較其他參數值穩定。接著我去調整了 preprocess 的參數 sen_len，也就是定義句子的長度，發現當長度從 20 條成 40 之後，在 train 的過程 accuracy 會在更小的 epoch 就達到了 0.81，最高甚至可達 0.819，推測可能原因是因為一次看了很多個字，也就代表說看的範圍更加的全面，也因此提升了準確率，最後我去調整了 RNN 架構的 hidden_dim 將它調高至 512，那此調整會發現有時候準確率可達到 0.82，那可能原因應該就是單純的可調參數變多的關係導致結果提升。

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）。

   我另外寫了一個 function 裡面做的事情大致上和 testing 這個 function 相同，唯一的差別在於後者會去判斷值 >= 0.5、< 0.5 來去賦予值 1 或 0，而前者則是會去和參數 threshold 做比較來賦予值 1 或 0 或不變，最後我再將結果中不為 1 或 0 的部分去除，而去除的 y 對應到的 x 也一樣去除，最後再將得到的 labeled data 去和原本的九成的 labeled data 做合併並且重新 train 一遍，參數、模型架構都維持不變，最後在 validation set 的準確率提升至 0.825。這邊我的 threshold 的值設為 0.8，而準確率之所以提高可能原因是因為儘管我們自己標記 label 的 data 或許有誤，但整體上有極大部分的還是正確的，因為我們 threshold 設為 0.8 的關係，而因為如此我們能夠 train 的 data 大概增加了將近八十萬筆，也因此在 validation set 的準確率能夠提升。