學號：B06902042　系級：資工三 姓名：劉愷為

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)

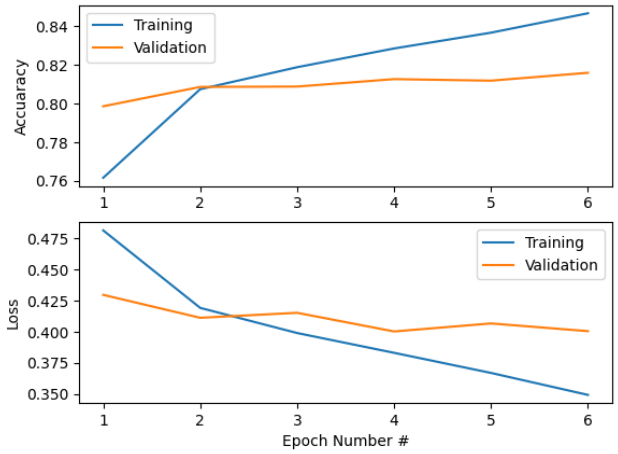
   我的RNN模型為

```
model = LSTM_Net(embedding, embedding_dim=250, hidden_dim=150, num_layers=3, dropout=0.5, fix_embedding=fix_embedding)
```

   Word embedding為

```
model = word2vec.Word2Vec(x, size=250, window=5, min_count=5, workers=12, iter=10, sg=1)
```

   Epoch = 6, lr = 0.001, batch_size = 128, sen_len = 30, optimizer = Adam




   準確率的部分在validation上為0.81592, 在kaggle上為0.82257

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+DNN | 0.6149 | 0.6149 |
| RNN | 0.1287 | 0.9832 |

   在BOW+DNN的模型下，兩句拿了相同的值。我認為是因為兩句所包含的但自完全一樣。而在RNN的模型下，第一句話相當負面，第二句話相當正面。像RNN這種會考慮單字的前後順序的模型才能拉高準確率，而BOW就不太適合來做文字語意的訓練。

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）

   我將sen_len調高到30，主要是因為句子的長度普遍都超過30，這樣就可以容納更多的文字，才能更精準的預測語意。但如果調太高的話，就有些會太小，有些會太大，反而會造成結果下降。我把LSTM的架構變得複雜(numOfLayer提高到3)，模型的可訓練參數也因此上升，使能得到更好的準確率。

   最後準確率: 0.82257

   初始準確率: 0.80219

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

   我取後1萬筆當作traing set, 前兩萬筆幫做validation set

   Epoch = 6, lr = 0.001, batch_size = 128, sen_len = 30, optimizer = Adam

| Before semi | | After semi | |
|---|---|---|---|
| Train_acc | Val_acc | Train_acc | Val_acc |
| 74.486 | 72.920 | 98.627 | 75.746 |

   我認為threshold設為0.9及0.1（即分數大於0.9者可被標示為1，小於0.1者可被標示為0，其餘資料則不採用）訓練出的效果較佳，最後將這些資料與原先的training set 合併後，再拿去train一次。當training set 小的時候，做了semi-supervised 後，training 和 validation acc 都有上升。Training 會這麼高是因爲semi-supervised的資料就是被一開始那一萬筆資料train之後選出來的。因此在training時這些資料一定會對，所以trainig acc 才會那麼高。而validation 提高的原因，我猜是因為資料量變大，模型因此也學到了更多東西。