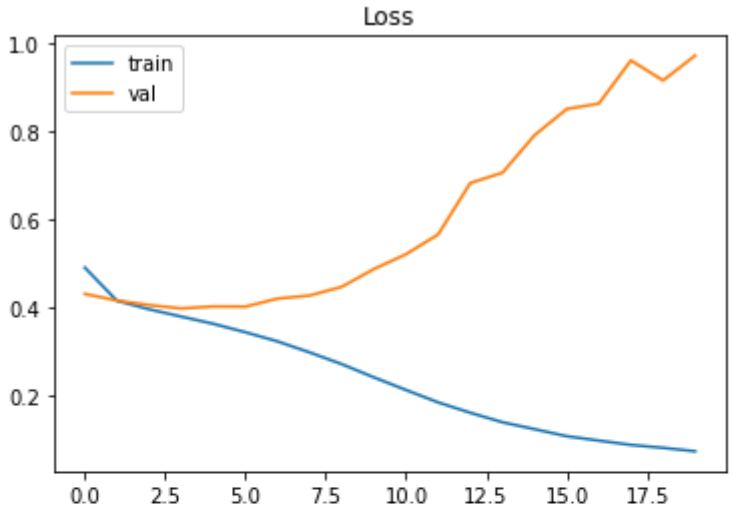
學號：b06901153 系級：電機三 姓名：林瑩昇

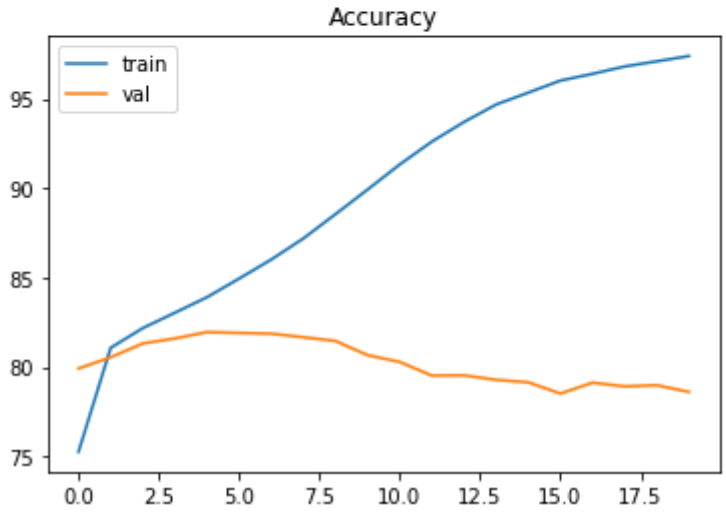
1. (1%) 請說明你實作的RNN的模型架構、word embedding 方法、訓練過程 (learning curve)和準確率為何？ (盡量是過public strong baseline的model)

   (1) RNN的模型架構：我用的是兩層bidirectional LSTM，hidden_dim=150，並且在FC layer中加入dropout=0.8

   (2) word embedding：我是把所有data都拿去做w2v，其中我把sen_len調到30，因為原先在助教的sample code中，sen_len=20其實會去掉許多有用的資訊，調整sen_len後，acc有顯著的提升

   (3) learning curve：基本上train的很快，val_acc很快就開始下滑





   (4) val_acc = 0.821

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。

   (1) BOW本身是計算一個句子中出現的word的次數，所以並沒有前後文的關係，而如果用word2vec的方式，便可以保留句子中前後文的關係。再加上RNN能夠提取時間序列中的信息，所以比DNN更適合NLP。

   (2) 由上述可以想見"today is a good day, but it is hot"與"today is hot, but it is a good day"兩個句子對BOW+DNN來說是沒有區別的(因為embedding之後是一樣的，所以餵進DNN model後的輸出也會一樣，都是5.384)。反之，RNN則可以比較出差異。以我在第一題中的model為例，第一句的分數會第二句稍高0.003左右，雖然符合預期，但我個人認為這其實是在誤差內，可能沒有解釋這個現象。

   (3) 我認為雖然語意上第二句確實相較第一句比較負面，但我的model可能沒有train的那麼好，能夠分辨這細微的差異，再來就是這兩個句子都偏短，在做embedding的時候又被一起補成sen_len=30的句子，更會使原本的差異變的更小。

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）

   (1) preprocess：一開始在train的時候，val_acc到0.802左右就會停滯不前，我去看了一下data，發現說其實很多sentense的長度都超過20，如果把sen_len設為20，可能就會截掉很多有用的資訊，所以把sen_len調整成30後，正確率提升了2%

   (2) model：因為之前修過李琳山教授的專題，那時看到的論文很多都有用BiLSTM，試過以後正確率會提升1%左右，效果沒有到很明顯，最主要還是sen_len的調整

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）。
   (1) 我的方法是先用label data訓練出一個model，然後用這個model對no label data做預測，出來的結果如果大於0.8設為1，小於0.2設為0。最後再把所有data拿下去train(大概會多3倍)。
   (2) 一開始沒有減少label data，threshold用上述方法的情形下，其實是會愈train愈糟的，validation set的正確率會不停下降。
   (3) 之後我把training data減少到20000筆，其餘作為testing set，再用上述方法再train一次，如果只用labeled data，testing正確率大概會在78%上下震盪，不過因為training data很少的緣故，很快就overfitting (expoch 到4左右)。
   (4) 之後我把新label的data與舊的資料concatenate起來再train一次，但是不知道為甚麼，不管跑多少epoch，結果仍然不好，有時候甚至會低於原先(78%)10%以上。我猜測可能不能用這種方法，原先用少量data訓練出來的model在預測時可能會有偏差，這樣有偏差的data在第二次training時至少比正確 data多10倍，當然也就訓練不好。