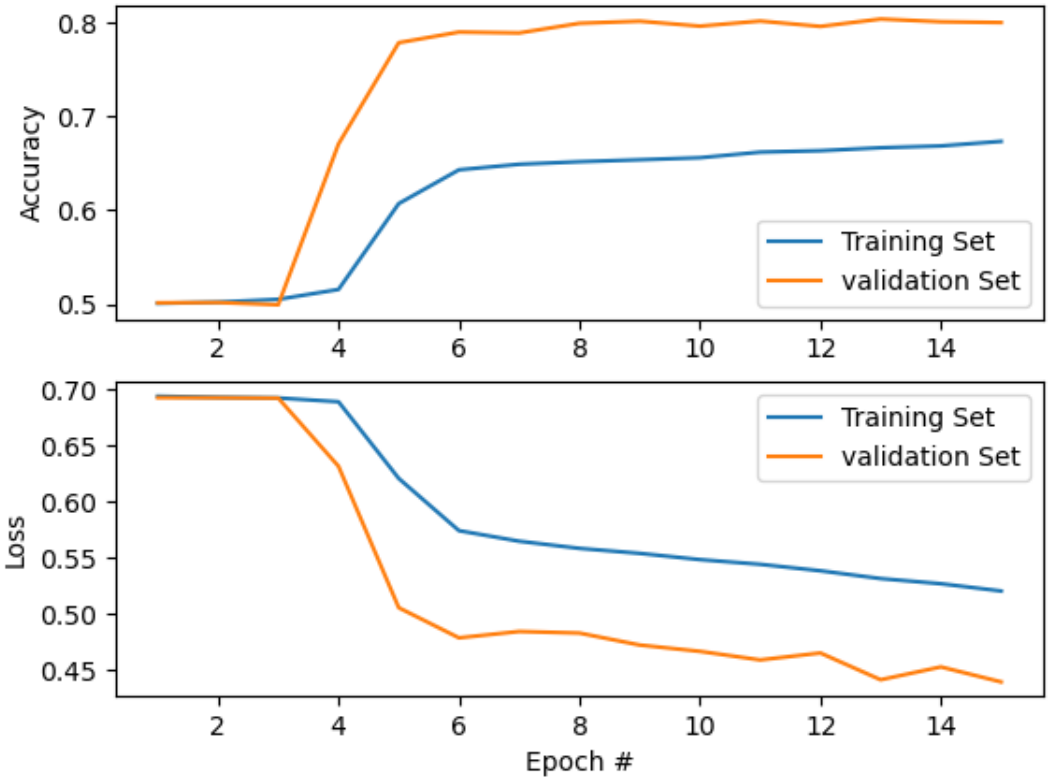
ML Hw4 report

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架構為LSTM組成，hidden dimension = 650、num_layer = 3，並使用了比率0.5的dropout，而DNN的部分只疊了一層。word embeding 的部份，我將window調成4，min count調成6，並將迭代次數調整為25次。準確率的部分，在validation set 上的準確率為0.822，而在kaggle上的準確率為0.82677。

   Learning Curve :



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"
RNN	0.1562	0.9873
BOW+DNN	0.5763	0.5763

在RNN的模型架構下，這兩句話的分數有著非常大的差別，第一句話相當的負面，而顯然的第二句話RNN模型也順利的判斷出正確的結果。而在兩句話中，BOW+DNN的模型都拿到了0.5763分，原因是看似語意完全不同的兩句話，所包含的單詞竟是完全一樣的，所以才導致了這兩句應該要很極端的語句拿到了一模一樣的分數。由此實驗可知，像BOW這種不考慮單詞順序及文法的訓練模式相當不適合文字方面的訓練，必須有如RNN這種會考慮到句子中前後字詞關係的模型才能成功拉高準確率。

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word embedding中的window參數調整至4，並將min count 參數調整至6。調整window參數可以讓我們決定多少距離內的字詞間可以互相影響，而我推測因為這些資料都是推特上的留言，這種網路上的留言或許通常比較"直接"，也就是不用看周圍太多字就可以抓準字詞的屬性。min count 改成6則可以篩選一些出現頻率較高的詞，通常社群軟體上的用詞常常會跟隨流行、趨勢、使用族群而導致某些字詞出現率較高，因此我認為將放入字典的門檻拉高可以使模型比較容易預測。

   另外，我把LSTM的架構變得複雜許多，如此一來模型的可訓練參數也上升，使我得到更好的準確率。最後我還參考了網路上的一些做法，實作ensemble training，使模型可以學習到更多原本可能沒學到或是學不夠的，也相當程度的提升了準確率。

   最後準確率： 0.822

   improve 前： 0.785

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）**

   以下訓練的training set 皆為將原本的training data 取前兩萬筆得到，並取第20000~第40000筆資料作為validation set。

No Semi-supervised	Semi-supervised
0.7561	0.7782

   在semi-superivse的部分，我將訓練好的model拿去餵給120萬筆的unlabeled data，並只取我有信心的data標籤化。經過測試後，我認為threshold設為0.75及0.25（即分數大於0.75者可被標示為1，小於0.25者可被標示為0，其餘資料則不採用）訓練出的效果較佳，最後將這些被標籤的資料與原先的training set 合併後再拿去train一次，進而得到最終的模型。由上表可發現實作semi-supervised的確對於提升準確率有幫助，而原因就是增加了許多資料（2,0000 $\rightarrow$ 55,0000），而這些增加的資料大部分是可信任的，因此模型理所當然的也能學到更多東西。