

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)

RNN:

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
=====		
embedding_1 (Embedding)	(None, 32, 250)	13944750
=====		
bidirectional_1 (Bidirection	(None, 32, 256)	291072
=====		
bidirectional_2 (Bidirection	(None, 256)	295680
=====		
dropout_1 (Dropout)	(None, 256)	0
=====		
dense_1 (Dense)	(None, 1)	257
=====		
Total params: 14,531,759		
Trainable params: 587,009		
Non-trainable params: 13,944,750		
=====		

Embedding:

用 gensim 的 word2vec (sentence length=32, size=300, window=5, min\_cnt=5, iter=10)，把所有出現過的單詞轉成 300 維的 vector。(sample code 是 250 維)

Learning Curve:

在 validation set 上的表現差不多在 epoch=5 以下就達到最好了，接下來就不太會有什麼太大的波動了。

準確率：

最後的 testing accuracy 差不多在 0.827，不需 embedding 或 semi-supervised。同時 validation 表現最好的 training accuracy 大概有到 0.85, 0.86，因此是 overfit。

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。

	分數	原因
BOW+DNN	前者較低，後者較高	沒有明顯的答案，可能是擺後面的句子較具影響力
RNN	都一樣	因為個別字詞的出現次數都一樣

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 )

在訓練過程中，我認為影響最為顯著的兩個點是 word2vec model 的參數及 NN 的深度。在將字詞轉換為向量時，我查了網路的資料，有蠻多人包括 google 的研究都是用 300 維，因此我也由 250 維改成 300 維，而結果也好很多。另外原本我都架很多層 Dense，在經過反覆嘗試後，發現只要在兩層 Bidirectional GRU 之後直接輸出就夠了，再多都會造成更慘的 overfitting。

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-supervised 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-supervised learning 所帶來的幫助 )。

因為 model 所預測出的不一定是真的 label，所以我只讓 prediction 大於 0.9 或小於 0.1 的加入新的 training set。最後在實作上沒有顯著的進步 ( 甚至是更糟 )，可能是因為取 label 的方式雖然好像增加了 data，但那些都是 model 本來就知道的規則，因此沒辦法幫助 model 更認識資料。再加上新增的那些 pseudo label 有可能本身就是錯的，進而擾亂訓練的進行。