

說明：請各位使用此 template 進行 Report 撰寫，如果想要用其他排版模式也請註明題號以及題目內容（請勿擅自更改題號），最後上傳至 github 前，請務必轉成 PDF 檔，並且命名為 Report.pdf，否則將不予計分。

中英文皆可，但助教強烈建議使用中文。

-----閱讀完以上文字請刪除-----

學號：R06921095 系級：電機碩一 姓名：吳睿哲

1. (1%) 請說明你實作的 RNN model，其模型架構、訓練過程和準確率為何？

(Collaborators:)

答：

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 40, 200)	49605600
gru_1 (GRU)	(None, 40, 256)	350976
gru_2 (GRU)	(None, 128)	147840
dense_1 (Dense)	(None, 32)	4128
dropout_1 (Dropout)	(None, 32)	0
activation_1 (Activation)	(None, 32)	0
dense_2 (Dense)	(None, 1)	33
activation_2 (Activation)	(None, 1)	0
Total params: 50,108,577		
Trainable params: 502,977		
Non-trainable params: 49,605,600		

在做完 parsing 以及 tokenizer 之後,我的第一層是放用 word2vec 做好的字典,我每次的 training 都會跑 100 個 epoch,一開始我只放了一層 128 個 cells 的 LSTM,但是 training accuracy 到 0.82 左右就有點快上不去的趨勢了,validation acc 大概也只能到 0.815 左右,應該是參數不夠的關西,於是我又多疊了一層,且把 cells 提升到 256 以及 128,這樣的作法,training accuracy 可以上到 0.88 左右,但是在跑了 15 個 epoch 以後會有很嚴重的 overfitting 的問題,於是我把 drop out 以及 recurrent drop out 都調到 0.4,且把 LSTM 換成 GRU,至於後面的 Dense 我原本只疊一層 dense 為 2,但是 2 層好像有稍微好一點,所以後來還是跌了 2 層,但我覺得這裡應該不是關鍵。以上的 RNN 的架構可以讓我的 training

accuracy 到達 0.845 左右,validation accuracy 可以到 0.827 左右,上傳 kaggle 最好的一次可以到 0.826。

2. (1%) 請說明你實作的 BOW model, 其模型架構、訓練過程和準確率為何?

(Collaborators:)

答：

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 1)	3001
activation_1 (Activation)	(None, 1)	0
Total params: 3,001		
Trainable params: 3,001		
Non-trainable params: 0		

Bag of word 的架構就簡單很多,只是單純用詞的頻率來作表達一個句子(我選了最常出現的 3000 個詞,因為再往上加會爆掉),一開始先用 parsing 好的 data,使用 tokenizer 的函式 text to matrix 轉換成 nn 可以吃進去的 input,mode 選了助教推薦的"tfidf",然後 nn 的架構我就只疊了一層 activation 為 sigmoid 的 layer,這樣的壞處是,他沒有辦法判斷一個句子中前後文的關聯性,但是他參數少,所以不會有 overfitting 的問題,且每個 epoch 只需要 2 秒(因為參數少),這樣的結果卻出乎意料之外的沒有很差,training accuracy 跟 validation accuracy 都可以到 0.78 左右,上傳 kaggle 結果也差不多,在沒有辦法判斷前後文的限制下,可以到 0.78 實在讓我深感意外。

3. (1%) 請比較 bag of word 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"這兩句的情緒分數, 並討論造成差異的原因。

(Collaborators:)

答：

	RNN(predict result)	BAG OF WORD(predict result)

Today is a good day,but it is hot	0.5312795	0.6302637
Today is hot,but it is a good day	0.99648213	0.6302637

由句子文意判斷,第一句話是負面的,第二句話是正面的,單純由結果判斷,rnn 是比較好的,由過程來判斷,rnn 有考慮到句子前後的關聯性,所以判斷的比較周全,bag of word 則無,所以 predict 出來的結果也不是這麼的正確。值得一提的是,bag of word 中,2 句的詞不考慮順序的話是一模一樣的,所以 predict 出來的結果也是一樣的

4. (1%) 請比較"有無"包含標點符號兩種不同 tokenize 的方式, 並討論兩者對準確率的影響。

(Collaborators:)

答：在使用同樣的架構之下,在保留標點符號的情況之下,predict 出來的準確率是比較高的,這樣的結果其實也是蠻合理的,因為有很多標點符號對文意都有很大的影響,舉例來說,在句尾加上”?” ,就可以讓一個句子變成疑問的語氣,加上”!” ,就可以讓一個句子成驚訝的語氣,所以有沒有保留標點符號我覺得是有差的,我有加標點符號的 model 大概會比沒加標點符號的 model 高 0.007 左右。

5. (1%) 請描述在你的 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。

(Collaborators:)

答：我的作法是把之前最好的 model 先 load 進來,predict “training nolabel.txt” 的句子,如果 predict 出來的結果大於 0.85 或是小於 0.15,我就把他假設為 1 或 0,也就是老師上課所提到的非黑即白的概念,接下來就直接拿原本的 training no label 及 predict 的結果來 train,然後用原本的 testing data 當作 validation,這樣的作法會讓 train accuracy 非常高,但是 validation accuracy 似乎沒什麼差,只比原來好一點點,幾乎沒有什麼差異。