Machine Learning HW5

學號：R04522631 系級： 機械碩二 姓名：盧玄真

1. (1%)請問 softmax 適不適合作為本次作業的 output layer? 寫出你最後選擇的 output layer 並說明理由。

本次作業所使用的架構如下(Table 1):

Table 1 RNN 架構

RNN
GRU(256, tanh)
Fully-Connect(256, relu)
Fully-Connect(256, relu)
Fully-Connect(256, relu)
Output(38, sigmoid)

我認為 softmax 還是適合當做這次作業 ouput layer 的 activation function，因為 softmax 會去強化那些在前一個 layer 值原本就較大的 output，這會讓分類可以更明顯，只是有一個缺點是當我們在做 multilabel 的 task 時勢必要去設定一個 threshold 當做選擇的門檻，當使用 softmax 當做 output 時，因為其值域沒有一定範圍因此難以設定。但是使用 sigmoid 相對於 softmax 分類起來就未必有那麼明顯，但是好處是最後的 threshold 設定會比較直覺得可以設定在 0~1 之間。

2. (1%)請設計實驗驗證上述推論。

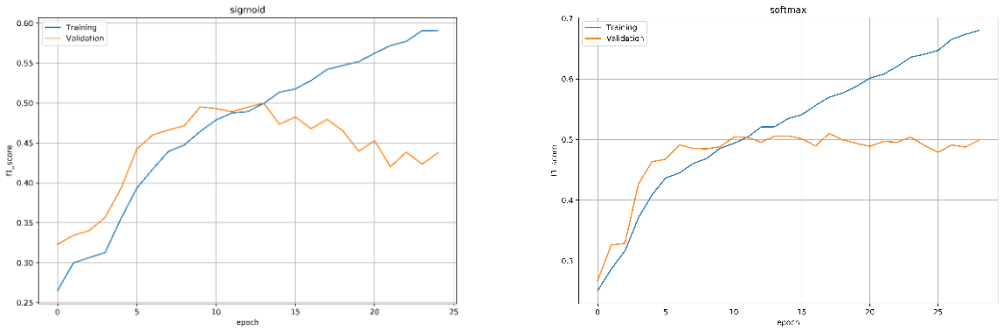

Figure 1 output layer activation function 比較(a)sigmoid 函數 (b)softmax 函數

從 Figure 1 output layer activation function 比較(a)sigmoid 函數 (b)softmax 函數 Figure 1 中可以看出 softmax 在最後 validation 的表現比較穩定，softmax 的 threshold 設定在 0.1，sigmoid 則設定在 0.5。因此 threshold 只要設定得當，softmax 仍然堪用。

3. (1%)請試著分析 tags 的分布情況(數量)。

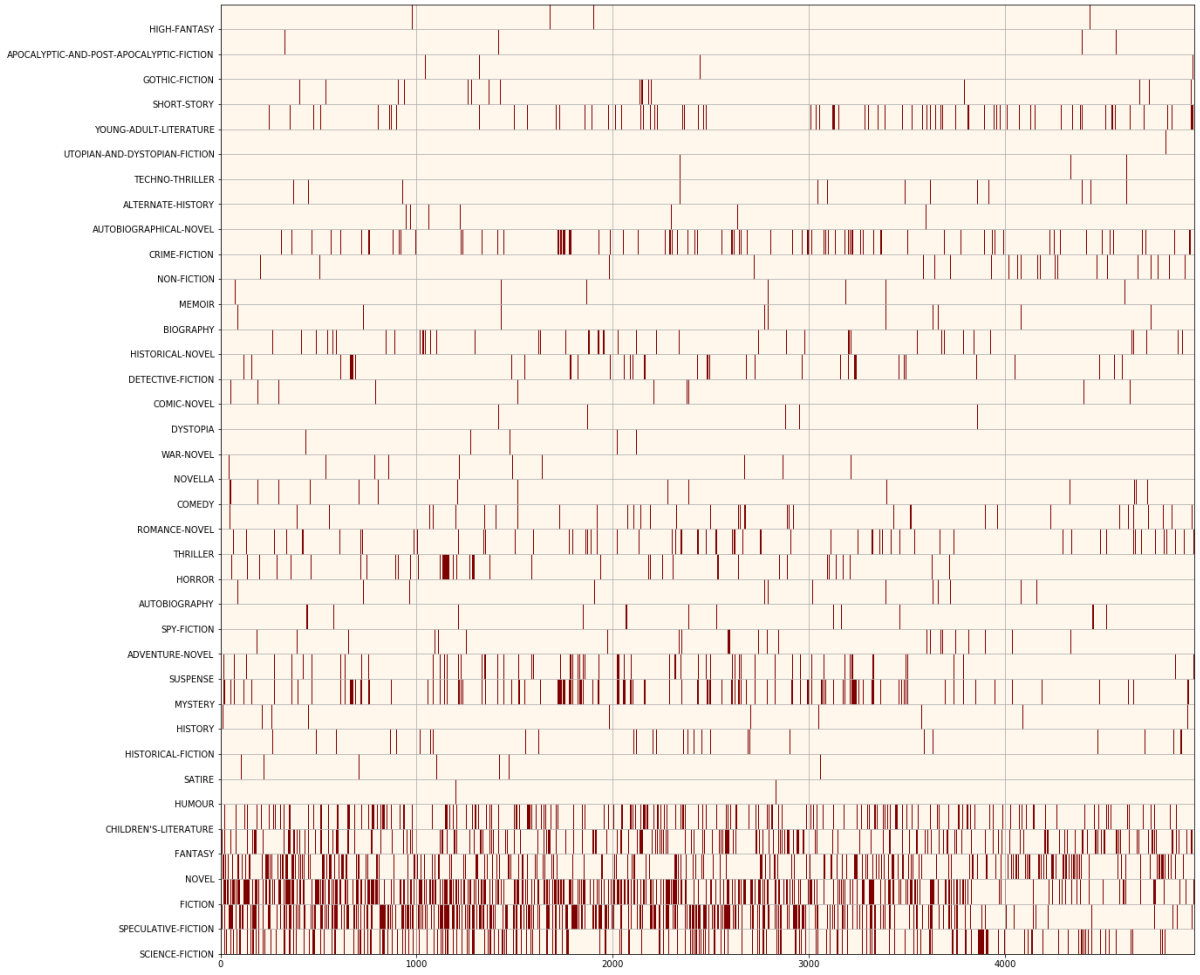

Figure 2 Tag 數量分布圖

如 Figure 2 為 Tag 數量分布圖，從圖中可以看出 FICTION 類的 tags 數量比較多，而且不管是哪一種 FICTION 類的文本，其後通常會跟著 FICTION 這個 tag，因此訓練的結果就會讓，當任何一種 FICTION 類出現的時候 FICTION 的 tag 出現機率也會比較高。而其他相關的類別也是如此。

4. (1%)本次作業中使用何種方式得到 word embedding?請簡單描述做法。

本次作業所使用的 word embedding 為 GloVe，其 model 主要的原理為假設兩詞向量 $w_{i,j}$ 在詞向量空間(word vector space)中的距離跟第三個詞 $\tilde{w}_k$ 向量的內積與這兩個詞 i, j 與第三個詞 k 的共同出現機率比有關，方程式如下:

$$F((\mathrm{w}_i - \mathrm{w}_j)^T \tilde{\mathrm{w}}_k) = \frac{P_{ik}}{P_{jk}} \qquad (1)$$

而詞共現機率 P_{ik} 的意思是詞 k 出現在詞 i 附近的次數除以所有的詞出現在詞 i 附近的次數。因此當兩個詞常常共同出現時其機率就會比較大。而兩個詞 i, j 與第三個詞 k 的共現機率比可以看出詞 k 與 i 跟 j 哪個比較常共同出現，與 i 較常共同則比值會大於 1，反之與 j 較常共同出現則比值會小於 1。經過一番推導後，可以就會得到 GloVe 的 cost function J 為:

$$J = \sum_{i,j=1}^{V} f\left(\mathrm{X}_{ij}\right)\left(\mathrm{w}_i^T \tilde{\mathrm{w}}_j + \mathrm{b}_i + \tilde{\mathrm{b}}_j - \log \mathrm{X}_{ij}\right)^2 \qquad (2)$$

其中 X_{ij} 為詞 j 在詞 i 附近出現的次數， $f\left(\mathrm{X}_{ij}\right)$ 是一個用來降噪的 weight function 當出現次數太少就會做某程度的削減，其圖大致如下。

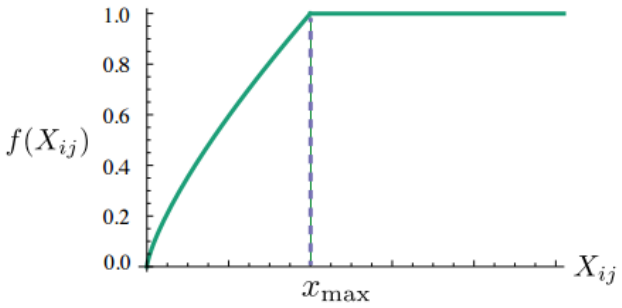

Figure 3 $f\left(\mathrm{X}_{ij}\right)$函數示意圖。

5. (1%)試比較 bag of word 和 RNN 何者在本次作業中效果較好。

Bag-of-word 模型架構如下(Table 2):

Table 2 Bag-of-Word 模型架構

Bag-of-word
Fully-Connect(32, tanh)
Fully-Connect(64, relu)
Fully-Connect(32, relu)
Output(38, sigmoid)

兩個模型的訓練過程如下 Figure 4、Figure 5 所示，可以看出在收斂速度上面，是 RNN 比較快，推測是因為 RNN 本身單次訓練資料量比較小參數較少再加上本身有防止 gradient vanish 的效果，因此可以快速達到收斂。再者，最後雖然兩個模型 training 的 f1_score 都差不多，但是 valiation 的 f1_score 卻是 RNN 比較高，因此看出 RNN 最後 overfitting 的情況比較不嚴重。所以最後結論就是，RNN 的訓練效果比較好!

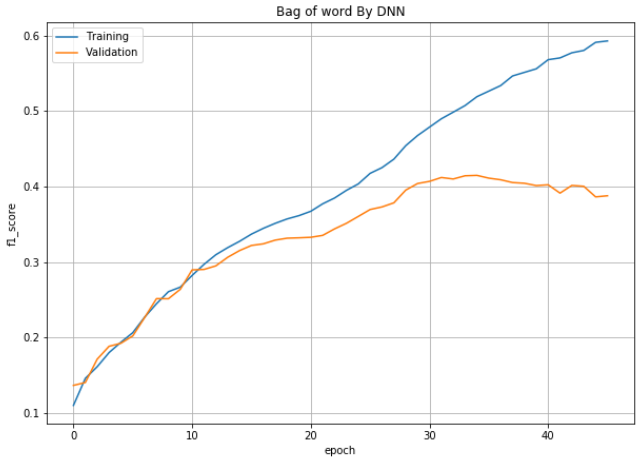

Figure 4 Bag of word 訓練過程

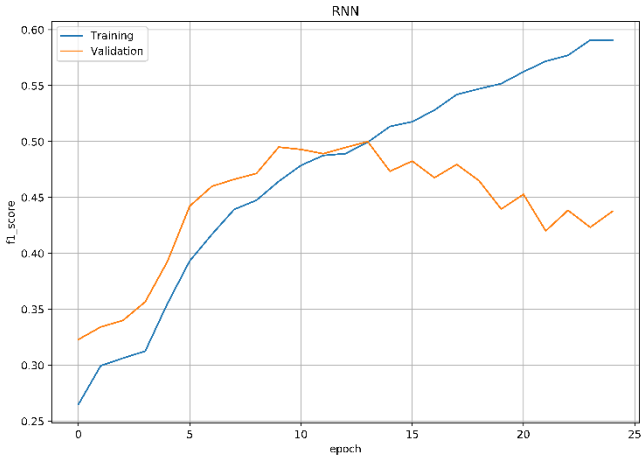

Figure 5 RNN 訓練過程