學號：B03901096 系級： 電機三 姓名：周晁德

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。

這次作業是 multi-label multi-class 的 classification，總共有 38 個 class，我的 network 架構是先 word embedding，然後是兩層每層有 256 個 neuron 的 GRU，接著是 DNN，DNN 中每個 neuron 的 activation function 皆為 elu，最後一層是 38 個 neuron 的 sigmoid，activation function 為 sigmoid，每個 neuron 代表一個 label，只要 neuron output > 0.4，則 text 屬於這個 class。(第五題有 network 架構圖！)。如果最後一層選擇用 softmax 作為 output layer，所有*class* 經過 $softmax$之後$ouput$的結果相加等於 1，又一筆 text 同時屬於多個 class，會使得每個 neuron 的 output 比較無法表達這筆 text 屬於這個 class (假設為 classA)的絕對機率，反而讓我們得知相對於其他 class，這筆 text 屬於 classA 的相對機率，此外一個使用 softmax 當最後一層的另一個問題我用一個例子來說明！

Input text	經過 softmax 前	經過 softmax 之後
Text1	2 個 neuron output = y 36 個 neuron output = 0	2 個 neuron output = 0.5 36 個 neuron output = 0
Text2	4 個 neuron output = y 34 個 neuron output =0	4 個 neuron output = 0.25 34 個 neuron output =0

由上面的例子，可以知道：

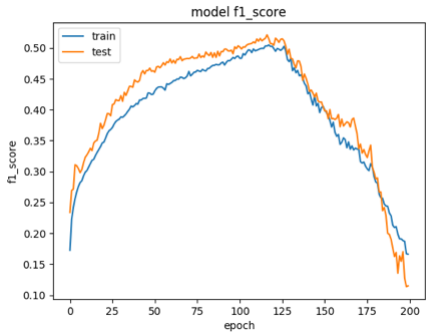
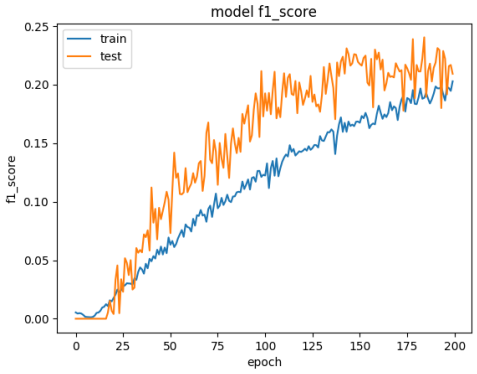
(1) Text1 應該會屬於 2 個 class
(2) Text2 應該會屬於 4 個 class

但是如果對 Text1 和 Text2 選擇使用相同 threshold，例如：threshold = 0.4 ，則 Text1 會屬於 2 個 class，但 Text2 會不屬於任何一個 class，所以 threshold 選多少對結果有很大的影響。

此外若當例子中經過 softmax 前 output = 0 的 neuron 現在 output ≠ 0，則 Text1 經過 softmax 之後數值為 0.5 的 neuron 數值將會下降，若下降到<0.4 且 threshold = 0.4，則 Text1 也就不屬於任一 class 了！

2. (1%)請設計實驗驗證上述推論。

表（一）

Sigmoid as last layer	Softmax as last layer
	

由上表我們可以發現在 threshold = 0.4 的時候，最後一層 layer 用的結果比最後一層用 softmax 來得好。

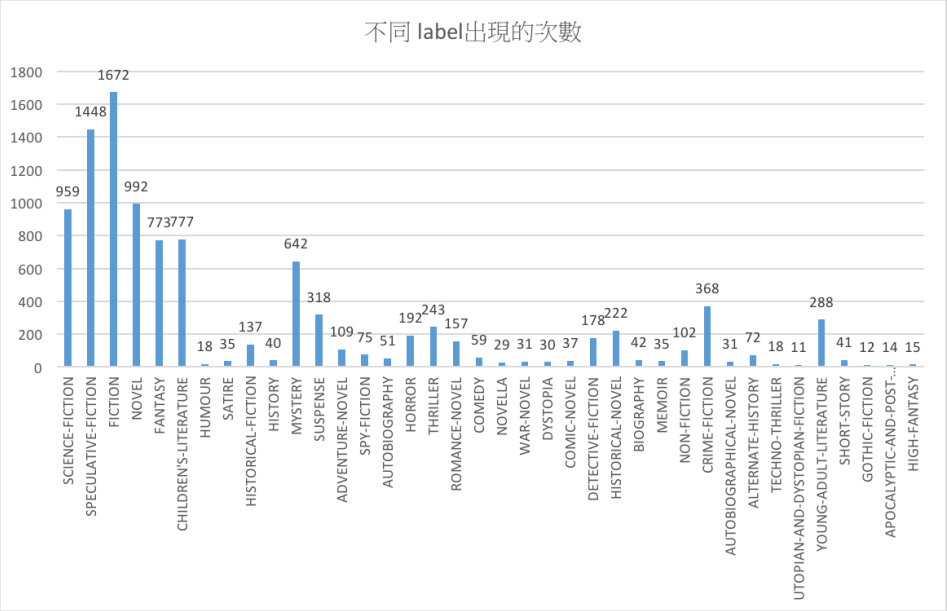
表（二）

	SCIENCE-FICTION	SPECULATIVE-FICTION	FICTION	NOVEL	FANTASY	CHILDREN'S-LITERATURE
True labels	1	1	0	0	0	0
Softmax	0.491871	0.335324	0.0986353	0.0257806	0.030802	0.00931461
Sigmoid	0.974386	0.866551	0.273753	0.0347275	0.169447	0.0452358

上表示我取 training data 最後 10 筆資料當 testing data 的其中一筆的結果，油表中我們可以發現，實際上這筆 text 的 label 是 SCIENCE-FICTION 和 SPECULATIVE-FICTION，如果最後一層用 sigmoid，則在對應位置的 neuron 輸出的值都> 我取的$thershold$ 0.4，但如果用 softmax 當最後一層 layer，因為 SCIENCE-FICTION 的數值最大，經過 softmax 之後只有這個位置的值 > 0.4，因此就會預測結果比較差！

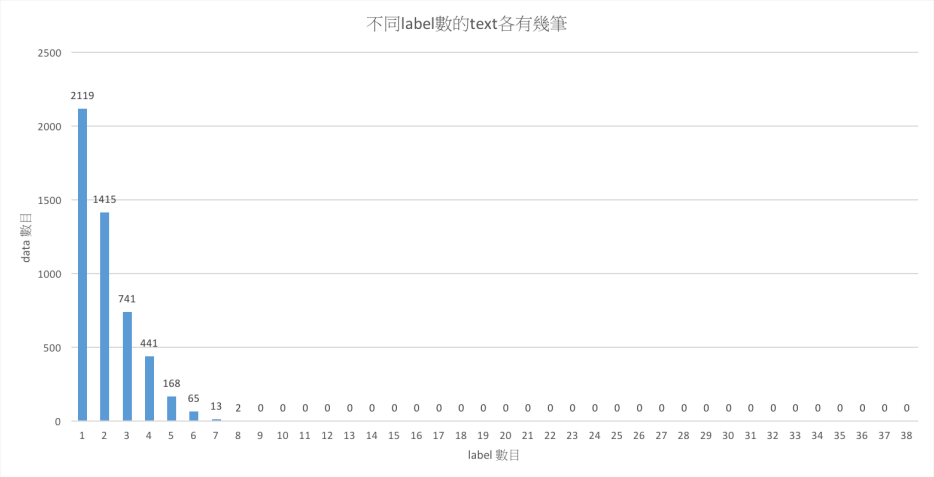
3. (1%)請試著分析 tags 的分布情況(數量)。

表（一）



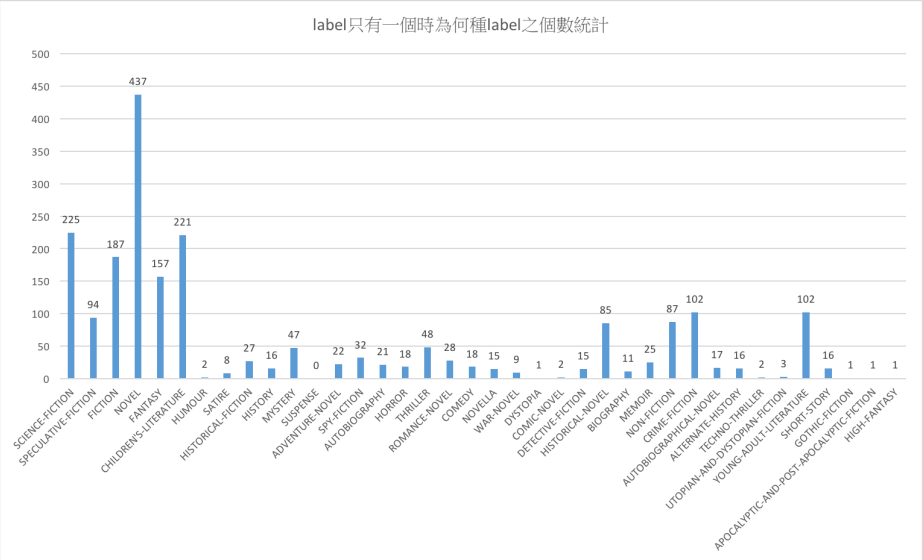
統計不同 labels 出現在不同 text 的次數，由上表中可以發現，Fiction 這個 labels 數目是出現最多的，而最少的是 UTOPIAN-AND-DYSTOPIAN-FICTION 只有 11 筆，此外我們可以發現不同 labels 出現次數的差距非常懸殊，即特定種類的 training data 很少，這會造成 training 出來的結果不是很好！

表（二）



因為每筆 text 有不同比 label 數目，所以統計不同 labels 數目的 text 各有幾筆，這個統計結果跟我預期的其實差蠻多的，居然是只有一個 label 的 text 最多！

表（三）



由於表（二）的結果應此我統計了，當一比 text 只有一個 label 時，這個 label 是什麼。從表中可以發現單一 label 最常出現的是 Novel，感覺蠻合理的，因為其實 novel 的定義還蠻廣泛的！

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

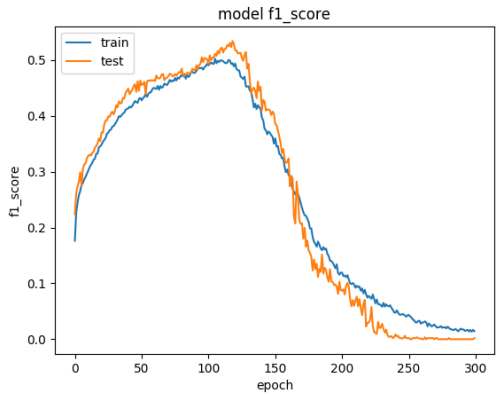
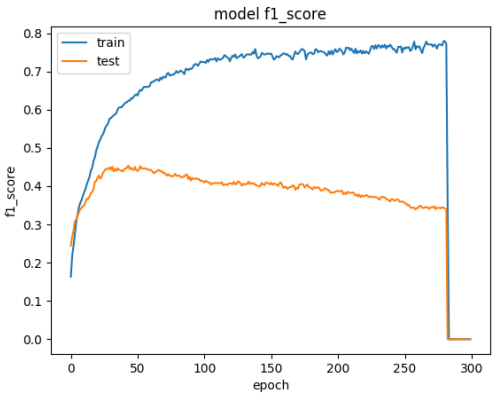
這次作業我是用 pre-trained 好的 Glove word embedding，這是屬於 count-based 的方法，先找一個 co-occurrence matrix X ($entries\ X_{ij}\ tabulated\ the\ number\ of\ times\ word\ j\ occursin\ the\ contextof\ word\ i$) ，接著去分解這個$matrix\ X$，而得到$word\ vector$和$context\ vector$，這個$model$用 2010 Wikipedia dump with 1 billion tokens and a 2014 Wikipedia dump with 1.6 billion tokens, Gigaword 5 with 4.3 billion tokens, a combination of Gigaword 5 and the 2014 Wikipedia dump totalling 6 billion tokens, and finally 42 billion tokens of web data from Common Crawl，相較於我們只用作業中的 words 來做 word bedding training ，直接使用 Golve 的 pre-trained 的 word embedding 可以有比較好的結果！

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

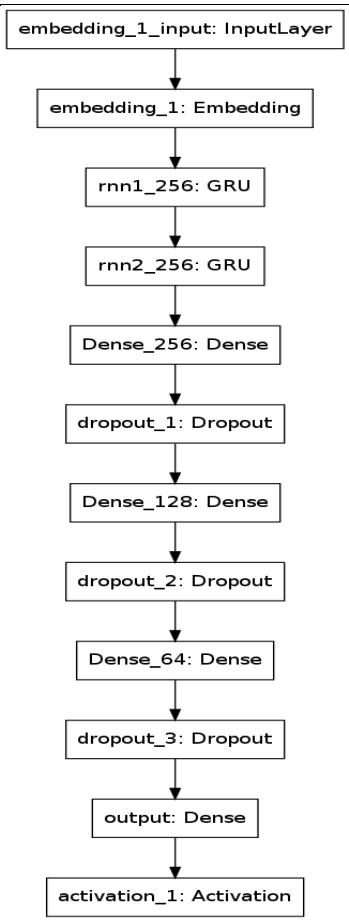
表（一）

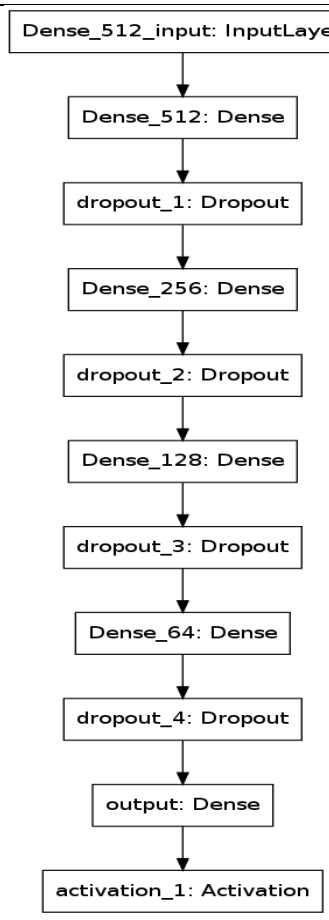
	時間(s/epoch)	Public leader board	# parameters
Bag of word	5	0.26912	27815782
RNN	35	0.51476	11241174

表（二）

RNN	Bag of word
	

表（三）





雖然上課時助教說 Word embedding 可以很輕鬆地過 simple baseline 但我試了，但我用 sequence_to_matrix(sequence, mode = “tfidf”)，試了幾次只得到以上的結果:

(1) 在表（三）中，bag of word 的 model ，$Dense_x$的x指的是那層 layer 的 neuron 數目，而每個 neuron 的 activation function 為 relu，只有 output layer 的 activation function 為 sigmoid。

(2) 在表（二）中，我們可以發現：

(a) RNN 在 validation set 上的表現，大致上跟在 training set 上的表現呈正相關，且最好的直都有超過 0.5，然而在過最高點之前 validation set 上的表現比 training set 來得好。

(b)bag of word 的表現，training set 上表現比 validation set 上來得好，training set 上最好的時候，validation set 上表現卻不是最好，此外 training set 可以達到快 0.8，然而在 validation set 最高大概只有 0.45，在圖中我也發現 bag of word 不論是 training set 或 validation set 的表現在某個點都突然急轉直下，我在其它次試驗中也都有看到相同的結果。

(3) 在表（一）中，即使 DNN 的 model parameters 數比 rnn 來得多，但 train 起來速度比較快，然而在 bublic leader board 的表現卻差很多。