學號：B03705006 系級： 資管三 姓名：侯舜元

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。

   softmax 並不適合。softmax 對向量進行歸一化，凸顯其中最大的值並抑制遠低於最大值的其他分量，且其輸出機率的合會是 1。這樣對於多 tags 的 article 的 performance 會很差。如果我有 3 個以上的可能 tags，threshold 設為 0.4，用 softmax 就無法都選到三個最有可能的 tags。

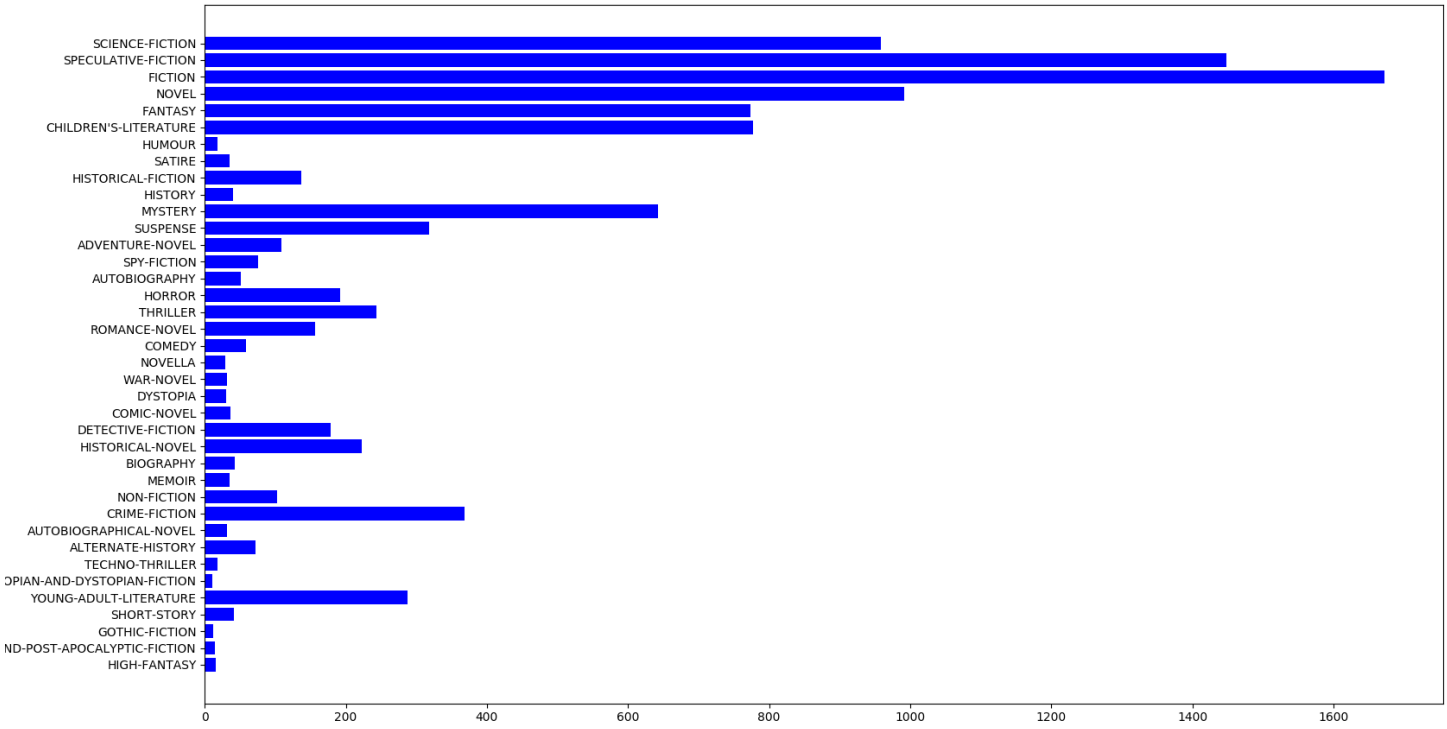
   我選擇使用 sigmoid，其 output 值一樣是 0~1，較能輸出的擁有每個 tags 合理的機率。

2. (1%)請設計實驗驗證上述推論。

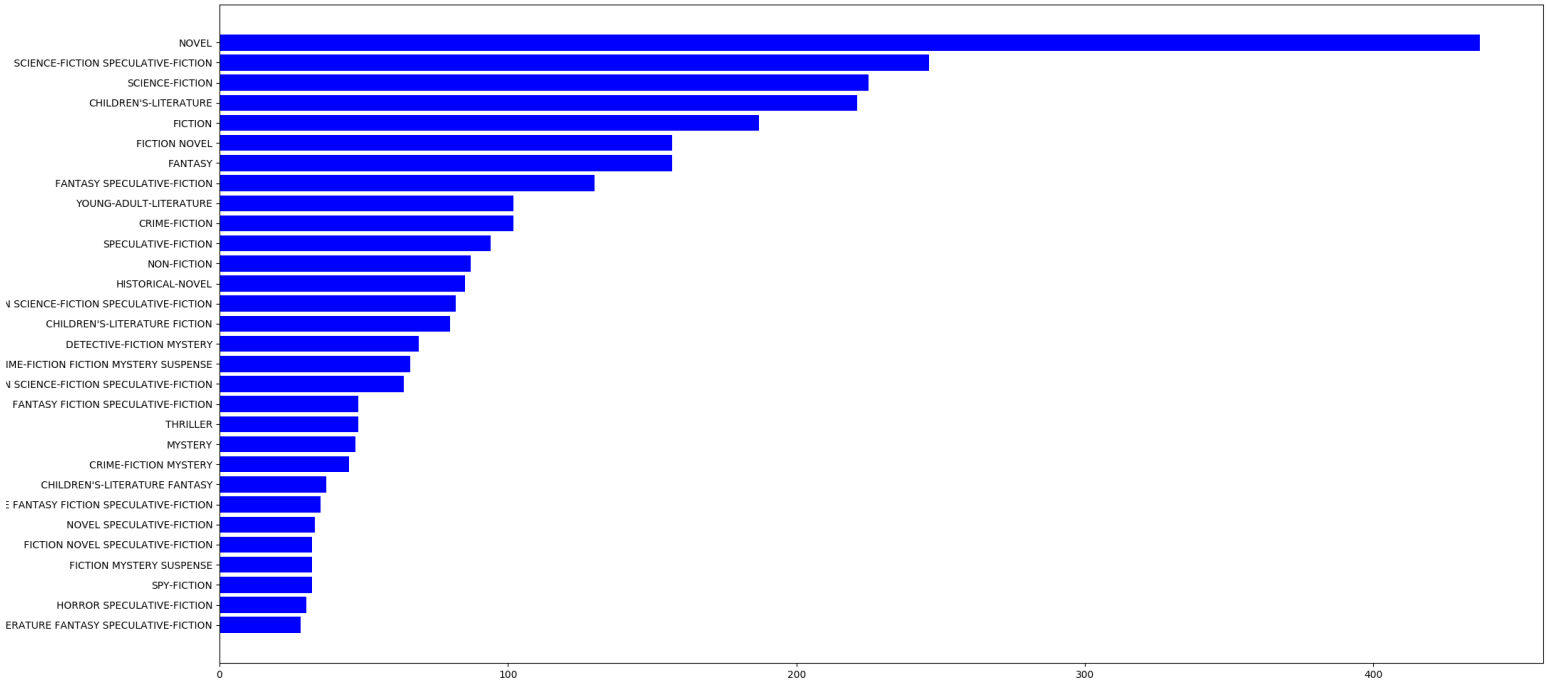
   將 threshold 設成 0.4 ，使用 softmax 當 output layer 的結果，的確沒有任何一篇有超過三個 tags，且絕大多數都是空白或是單一 tag，與前推論合理，機率被分給 38 個 tags 了。且得到的 performance f1 score 大概在 0.2~0.3 之間，不佳。

   而使用 sigmoid 當 output layer 的結果，performance 明顯教佳，也能夠同時選出好幾個高機率的 tags 了。得到的 performance f1 score 大概在 0.49~0.55 之間。

3. (1%)請試著分析 tags 的分布情況(數量)。



針對 38 個不同的 tag 做分析，可以看出’NOVEL’小說類和’FICTION’類別是最多的。（其中跟 FICTION 相似的 也有 SCIENCE-FICTION 科幻類等不同的想像的 article，數量也都不少）。其次就是 FANTASY，CHILDREN’s-LITERATURE 等。最少的為 UTOPIAN-AND-DYSTOPIAN-FICTION，烏托邦與反烏托邦的文章。



列出最常出現的前 30 種 tags 組合，除了原先常見的一些 tags 之外，可以看出 SCIENCE-FICTION 和 SPECULATIVE-FICTION 很常一起出現，算是關聯性很高的 tags。

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

   先用 Tokinizer 將 word sequence 轉換成 index sequence，並 padding 到一樣的長度。之後再用’ glove’做 word embedding 將 index sequences 都其轉換成一個 200 dimensional 的 array，最後得到 embedding_matrix

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 的 model 使用 freq 的 model，結構如下付圖。Output layer 一樣選擇 sigmoid，drop out rate 設成 0.4。其 performance 大概也在 0.5~0.53 之間，和 RNN 差不多（略差一點），但已足夠過 simple baseline 了。但是其難以 performance 再往上，故要處理 sequence 的 data 可能還是要用 RNN 略好一些（但也教難 train）

```
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 720)               72000720
_________________________________________________________________
activation_1 (Activation)    (None, 720)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 720)               0
_________________________________________________________________
dense_2 (Dense)              (None, 360)               259560
_________________________________________________________________
dropout_2 (Dropout)          (None, 360)               0
_________________________________________________________________
dense_3 (Dense)              (None, 180)               64980
_________________________________________________________________
dropout_3 (Dropout)          (None, 180)               0
_________________________________________________________________
dense_4 (Dense)              (None, 90)                16290
_________________________________________________________________
dropout_4 (Dropout)          (None, 90)                0
_________________________________________________________________
dense_5 (Dense)              (None, 38)                3458
_________________________________________________________________
activation_2 (Activation)    (None, 38)                0
=================================================================
Total params: 72,345,008
Trainable params: 72,345,008
Non-trainable params: 0
```