學號：b04902053 系級：資工二 姓名：鄭淵仁

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。**

不適合。我使用的是 sigmoid。

原因是因為 softmax 會讓 predict 出來的值的總和是 1，但是這次作業的結果是 multi label 的，所以 predict 出來的各個 class 的機率應該要互相獨立，而不會有總和是 1 的限制。

2. **(1%)請設計實驗驗證上述推論。**

我使用 bag of word 以及同樣的 model 去訓練資料，只有最後的 activation function 分成 softmax 和 sigmoid 兩種，並且分別調整成最好的 threshold 來測試，結果如下表：

	sigmoid	**softmax**
threshold	0.4	0.2
validation set score	0.53386	0.50231
kaggle public score	0.51853	0.47662
kaggle private score	0.50022	0.46001

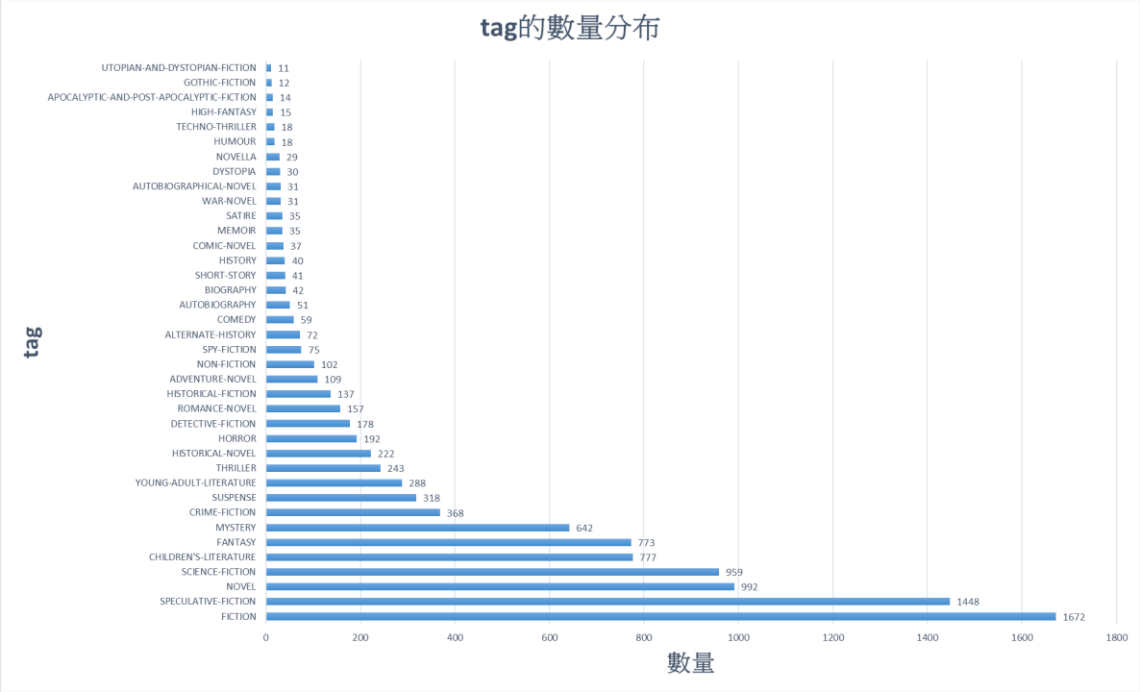
如果將上述實驗改成 RNN (GRU)來測試，結果如下表：

	sigmoid	**softmax**
threshold	0.4	0.2
validation set score	0.50519	0.48143
kaggle public score	0.50121	0.47966
kaggle private score	0.47602	0.45681

從上面兩個表格可以發現：sigmoid 的結果都比 softmax 的結果還要好。

3. **(1%)請試著分析 tags 的分布情況(數量)。**

tags 的數量分布如下圖：



從上圖可以很明顯的觀察到：所有 38 個 tag 中，所以只有少數 tag 的數量特別多，其它 tag 的數量特別少。

例如其中有 7 個 tag（FICTION、SPECULATIVE-FICTION、NOVEL、SCIENCE-FICTION、CHILDREN'S-LITERATURE、FANTASY、MYSTERY）的數量超過 400 個，總數量大約就占了總 tag 數量的 70%。相反的，有 20 個 tag 數量不到 100 個，總和只占了總數的 6.7%。

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

我的 word embedding 使用 glove 提供的檔案。

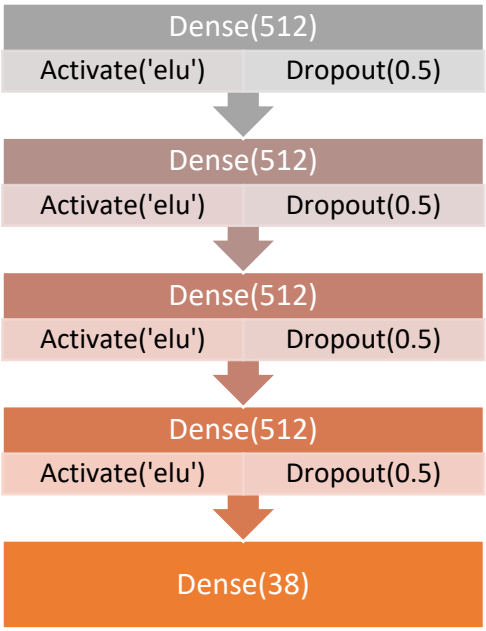
glove 是網路上生成好的 word vector。我選用的 glove 是使用 2014 年的 Wikipedia 和 Gigaword 5 做為訓練資料製作出來的。

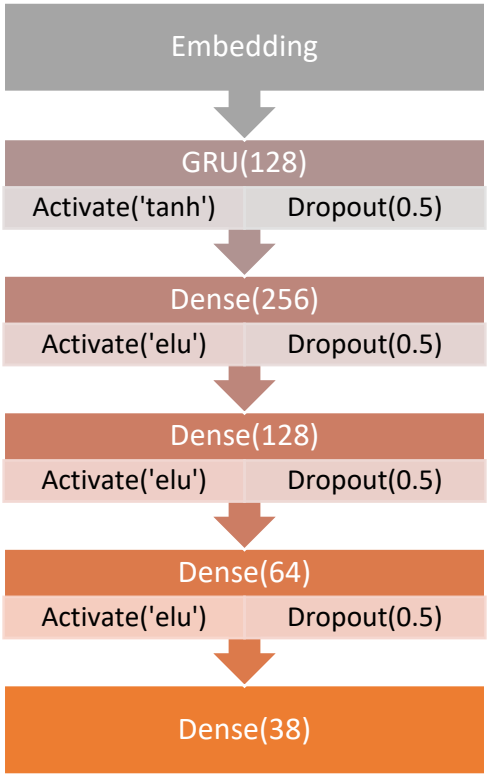
glove 訓練的方法是使用 Stanford tokenizer 造出一個大 word co-occurrence matrix，然後訓練出 word vector，目標是讓任何兩個 word vector 的內積結果盡量與兩個詞同時出現的頻率一致。

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

經過實驗比較，bag of word 比較好。實驗的方法是：我自行調整參數讓兩種方法都達到自己的最佳結果，兩者的模型架構如下：

bag of word 的模型架構 **RNN 的模型架構**





做出來的平均實驗數據如下：

	bag of word	**RNN**
kaggle public score	0.51853	0.50121
kaggle private score	0.50022	0.47602
validation set score	0.53386	0.50519

從上述數據可以看出來：bag of word 的效果比 RNN 還要好。

我想，會有這樣的結果是因為書本 summary 的寫法很有可能都是類似的，所以會多考慮文字次序的 RNN 反而可能考慮太多因素，而無法訓練出好的結果。