學號：B03901026 系級： 電機三 姓名：許凱傑

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。

在做 logistic classification 常用的 output layer 有兩種 activation：sigmoid 和 softmax，他們的公式如下，其中 $\bar{x}$ 為一個$(1 \times K)$vector 且 x_i 為第 i 個元素：

$$\text{sigmoid}(x_i) = \frac{1}{1 + e^{-x_i}}$$

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$$

通常如果是 multiclass 會採用 softmax，如果是兩類會用 sigmoid。但是因為這次的作業是 multi-label 的 classification，所以我希望可以把判斷每個 class 的機率越接近 0 或 1 越好，但 softmax 會使得機率總和=1，而不符合我的期望，所以我採用的是 sigmoid。

2.(1%)請設計實驗驗證上述推論。

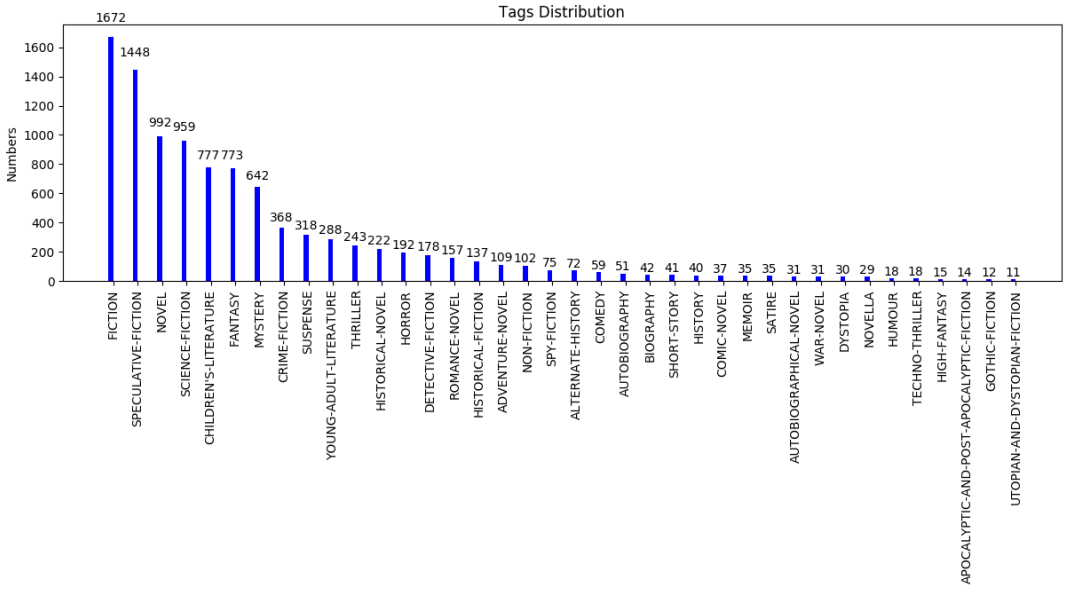
使用 sigmoid 的 kaggle 分數是：0.50384

使用 softmax 的 kaggle 分數是：0.45434

此外，我有把一些 test 的 output print 出來看，因為 softmax 有作歸一，所以 threshold 相對 sigmoid 來說比較難設成一個大家都通用的感覺，很容易只有一個 tag 產生，而非有多種 tag，不過這次的 data 感覺蠻多都只有 1-2 個 tag，所以分數還不會到特別差。

3.(1%)請試著分析 tags 的分布情況(數量)。

我覺得這次的作業 tag 分布蠻不平均的，對數量極少的 tag 來說，感覺很難使機器抓到辨認出該 tag 的特徵，使得大部分都是不屬於這個 tag，加上很多 tag 的分辨感覺很困難，例如對我們人來說要怎麼透過語意區分 fiction 跟其他種 fiction，所以我覺得這可能是這次作業很難 train 的很高的原因之一。



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

首先利用 keras 的 preprocessing.text 的 tokenizer 透過 fit_on_texts 找出 word-to-sequence 的 dictionary，並轉化 texts 成為 sequences。接著利用 glove.6B.200d.txt 當作 embedding vector set，建立好 word-to-vector 的 embedding_index 這個 vector set 後，再建立一個 sequence-to-vector 的 embedding_matrix 的 matrix，並把他寫入 keras 的 embedding layer，並使這些 weight 是 untrainable，這樣每個 seq 一定會對應到相同的 vector。

其中有試過不同 dimension，包括 50d、100d、300d，但 200d 比 50d、100d 效果都好，跟 300d 效果差不多，所以考慮 model 大小便採用 200d。

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

使用 bag of word 的 kaggle 分數是：0.36723

使用 RNN 的 kaggle 分數是：0.50384

從 kaggle 可以看出 RNN 的效果比較好，我覺得是因為 GRU 或 LSTM 會把上一個 unit 的結果傳到下一個 unit，也可以透過 return_sequence，相對來說有考慮到上下文的相對關係，但是 bag of word 只是計算出現的頻率，可能並無法達到較好的效果。