學號：B03502040 系級： 資工三 姓名：劉君猷

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。

答：我覺得這次作業不太適合用 softmax 作為output layer 的 acivation function。因為 softmax 的用意是要使被預測出的 classes 出現的機率限制在 0 ~ 1 裡面，也就是被設計用來預測只有單一一個 class 的情況。然而在這次作業的目標中，我們需要同時預測出多個 classes，這樣的話，應該要用 sigmoid 會比較好。我使用了 sigmoid，這樣的情況會像是針對每一個 class 看有或沒有，也就是每一個 class 之間的判斷是獨立的，互相不影響。若在某一個 class 輸出通過 sigmoid 後，機率超過 0.5，就會輸出 1 (這個 class 要選)；若不到 0.5，那就是 0 (這個 class 不要選)。

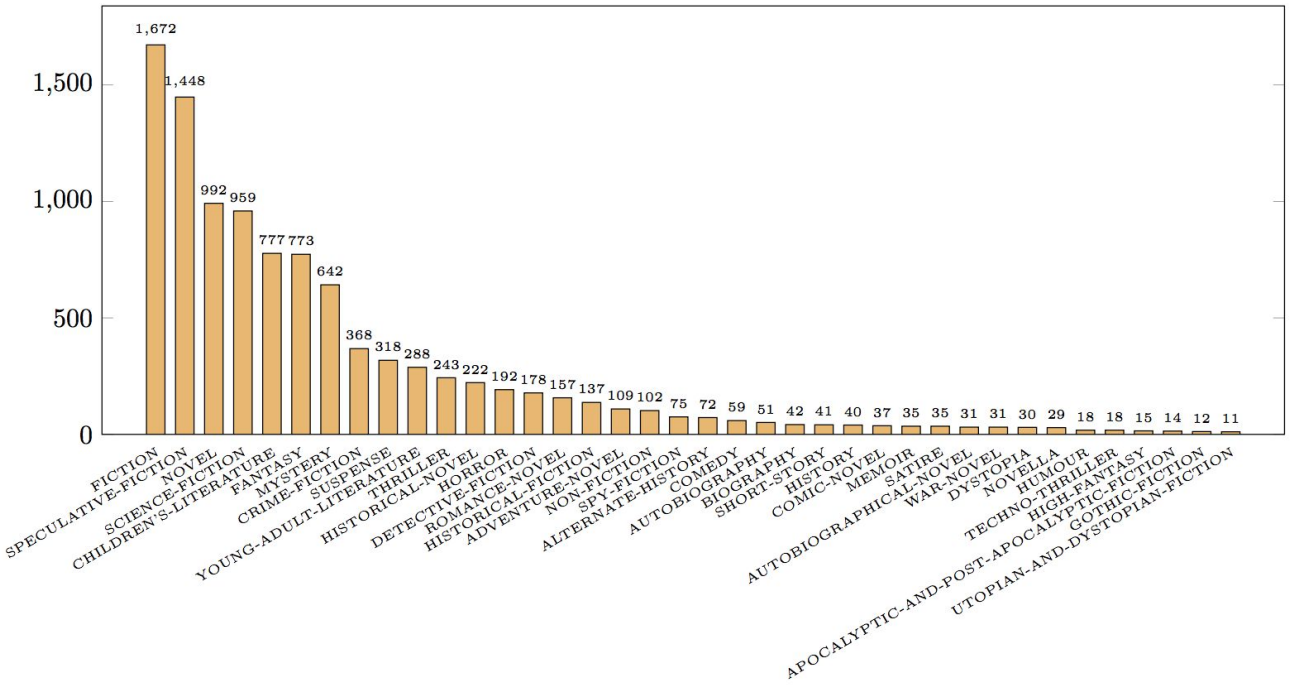
2. (1%)請設計實驗驗證上述推論。

答：我寫了兩份幾乎一樣的 code，差別在只改了 output layer 的 activation function。一個是 sofmax，另一個是 sigmiod，然後針對同一筆 validation set (前 400 筆) 的 f1 score 來比較，也上傳到 Kaggle 比較。同樣都跑 46 個 epoch，以下是實際測試的結果：

| Activation function | Validation set f1 score | Kaggle public f1 score |
|---|---|---|
| sigmoid | 0.5045 | 0.46583 |
| softmax | 0.1464 | 0.15321 |

3. (1%)請試著分析tags的分布情況(數量)。

答：以下這張圖是 tags 在 training data 上的分佈，可見不同 tags 的數量分布懸殊。



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

答：我使用 GloVe 的 pre-trained model 來做 word embedding。GloVe 是一種 count-based 的 word embedding，它的訓練方式是去吃一大堆文章當作 corpus，然後紀錄相互兩個字一起出現的次數，製成一個 co-occurence matrix。接著找到一組給每個字的 word vector，使得任意兩個字 wi, wj 它們各自的 word vector vi, vj 內積之後會最接近 co-occurence matrix 取 log 後的值。

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

答：bag of word 的部分，我使用 tfidf 來實作。在我比較結果下，bag of word的結果較 RNN 版本好。單純使用 tfidf 搭配 linearSVC 可以在不需要調參數的情況下直接讓 f1 score 來到 0.50 左右；然而，若使用 RNN 來做的話，我自己是需要兩層 GRU 和三層 Dense Layers 配上 0.4 的 dropout rate 才能在 public f1 score 上接近 0.5 左右，而且模型的訓練上的時間也要快一個半小時才能訓練出這樣準度的 model。在考慮到時間跟準度的情況下，bag of word 都比 RNN 來的好很多。