學號：B04902112 系級： 資工二 姓名：張凱捷

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。

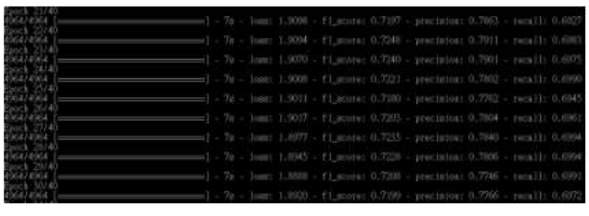
- 不適合
- 使用sigmoid
  - 因為softmax對應到的output是相加為1，而sigmoid是每一個對應到0~1，本題為multi label multi class，每一個label被選到的機率為0~1比較make sense，若使用softmax的話threshold會比較難調(每篇文章的label數量不一樣)。
  - 若文章X的label數量為K個，則softmax threshold必須<1/K才有可能準確預測
  - 不設一個很小的threshold的原因: 在預測時並不會那麼精準，若threshold設太小可能會造成一些錯誤的label被輸出

```
model = Sequential()
model.add(Embedding(num_words,
                    embedding_dim,
                    weights=[embedding_matrix],
                    input_length=max_article_length,
                    trainable=False))
model.add(GRU(128, recurrent_dropout=0.25, dropout=0.25))
model.add(Dense(256,activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(64,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(38,activation='sigmoid'))
```

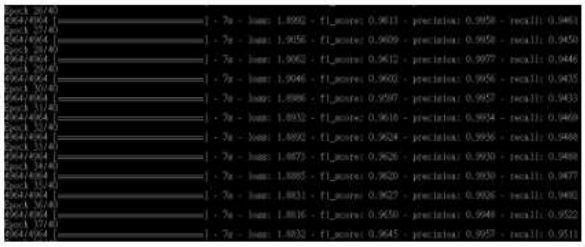
2. (1%)請設計實驗驗證上述推論。

- 下表為Label數量統計，由於softmax是總和為一，所以在不知道數量的情況下沒辦法設出一個好的threshold

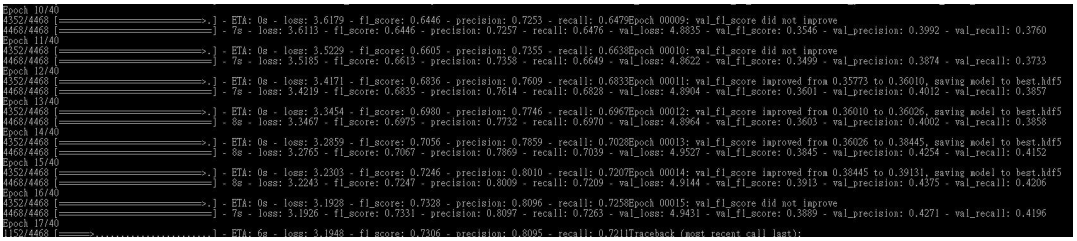
- threshold : 0.4



- threshold : 0.2

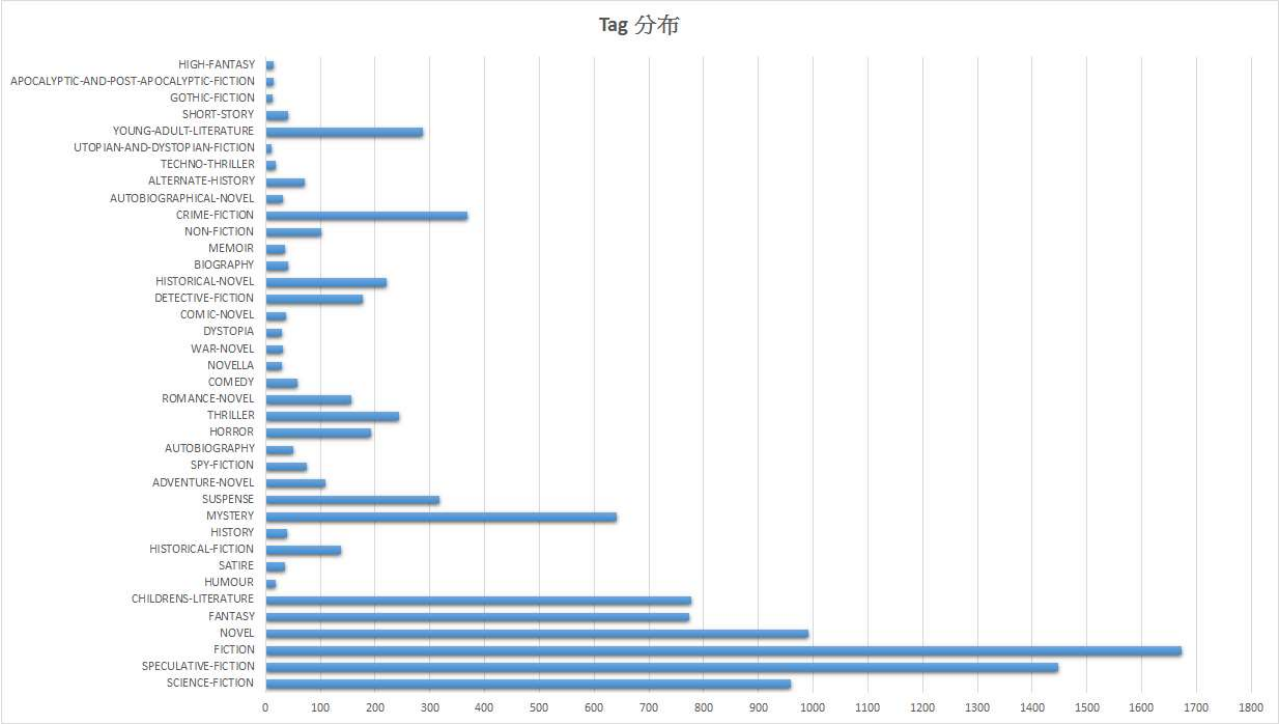


- threshold 0.2 validation



- ❖ 由上圖可以發現，threshold必須越小才越能預測出一些label數量多的文章
- ❖ 然而就算threshold設小一點validation的performance也不好

3. (1%)請試著分析tags的分布情況(數量)。



- ❖ 分布非常不平均，38個class中就有18個數量小於100
- ❖ 觀察預測出來的檔案後，的確這18類幾乎沒有出現，於是我嘗試將這18個class挑出另外train一個不同參數得model再將其append入答案之中

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

- ❖ 一開始使用word2vec套件，將training data & testing data當作corpus適用CBOW model進行訓練，獲得dim 100的word vector
- ❖ 最後在RNN的部分選擇使用glove的dim 100的檔案，因為我認為只用training data & testing data沒辦法有效的分類文字

- ❖ glove訓練概念
  - The GloVe model is trained on the non-zero entries of a global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus.
  - GloVe is essentially a log-bilinear model with a weighted least-squares objective. The main intuition underlying the model is the simple observation that ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning.

reference: https://nlp.stanford.edu/projects/glove/

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

- ❖ **模型設計:**
  - 先將stopword去除，tokenizer後將每篇文章裡的單字設成1，獲得一個長度為單字總數的list
  - 接著丟給DNN train一波

```
model = Sequential()
model.add(Dense(512, input_dim=num_words, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(256,activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.7))
model.add(Dense(len(tag_list),activation='sigmoid'))
```

❖ **數據:**

平均1個epoch所需時間

- RNN: 10s
- bag of word: 4s

Kaggle public score

- RNN: 0.50685
- bag of word: 0.51440

訓練速度

❖ **分析:**

- 我認為bag of word的表現比較好適合理的，因為這一題是文章分類，而不同總類的文章單字會差很多，比如說自傳有些裡面就直接包含biography了，或是戰爭通常有war weapon等等。
- 不過由於資料量不夠多，這題用bag of word也沒有非常理想，比如說有同樣兩個字苗速同一個意思，但是training的時候沒有出現過這個單字的話，就無法有效辨別，這是比word vector差的一個點。

- code: bag_of_word.py