ML HW5 Report

學號：B04611015 系級： 資工二 姓名：陳佳佑

P1. 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 : 會壓縮到其他維度的表現，最後每筆資料只會有單一 label 突出。

Sigmoid : 各個 label 的 performance 不會壓縮到彼此，因此可以呈現出 Multilabel 的情況。

NN Model 的框架:

```
model = Sequential()
model.add( Embedding(num_words,100,weights=[embedding_matrix],input_length=max_article_length,trainable=False) )
model.add(Bidirectional(GRU(128,dropout=0.3,activation='tanh',return_sequences=True),merge_mode='ave'))
model.add(GRU(128,dropout=0.4,activation='tanh'))
model.add( Dense(output_dim = 256, activation='relu') )
model.add( Dropout(0.4))
model.add( Dense(output_dim = 128, activation='relu') )
model.add( Dropout(0.4))
model.add( Dense(output_dim = 64, activation='relu') )
model.add( Dropout(0.4))
model.add( Dense(output_dim = 38) )
model.add( Activation('softmax') )
```

因為我並不是分成 38 個 model 下去 train，所以最後選擇了 Sigmoid 為 output layer。

P2. 請設計實驗驗證上述推論

我以 P1 的 model 做實驗，分別將 output layer 設為 Softmax 與 Sigmoid

得到的結果

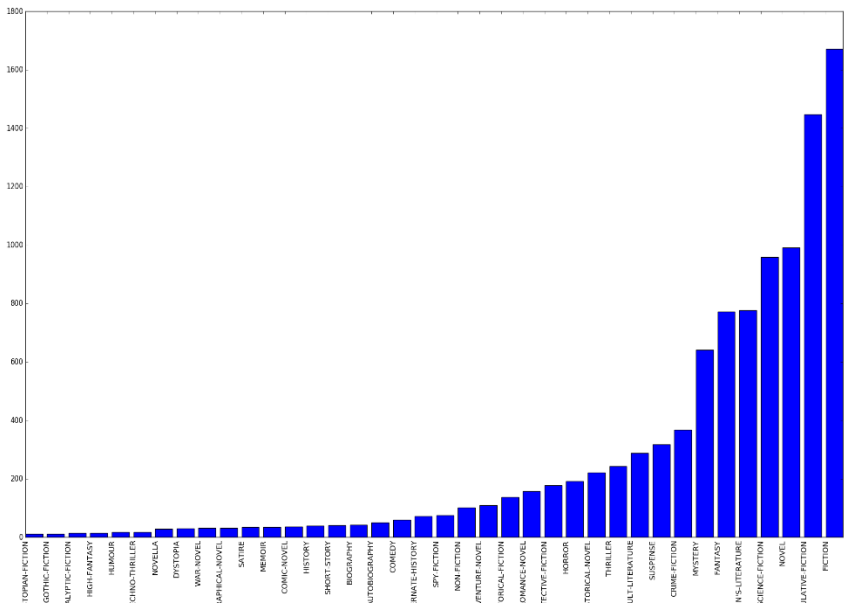
Softmax 的 predict 結果中，有 993 筆資料的 predict 為空，其餘都只有單一 label，這符合上述所敘的結果。

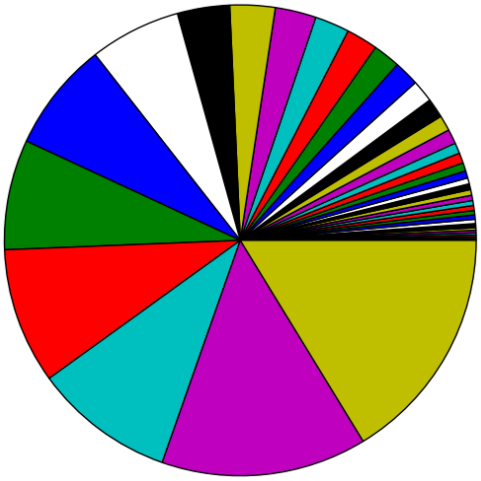
```
In [1]: string = open('out1.csv','r').readlines()

In [2]: dic ={0:0,1:0,2:0,3:0,4:0,5:0,6:0}

In [3]: for i in string:
    dic[len(i.replace('\n','').split(',',1)[1][1:-1].split())] += 1
   ...:

In [4]: dic
Out[4]: {0: 993, 1: 242, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0}
```

Sigmoid 的 predict 結果中，沒有欄位為空，大多數都有 label。

```
In [1]: string = open('4','r').readlines()

In [2]: dic = {0:0,1:0,2:0,3:0,4:0,5:0,6:0,7:0,8:0}

In [3]: for i in string:
   ...:     dic[len(i.replace('\n','').split(',',1)[1][1:-1].split
   ...: ())] += 1
   ...:

In [4]: dic
Out[4]: {0: 0, 1: 28, 2: 408, 3: 343, 4: 335, 5: 101, 6: 18, 7: 2, 8: 0}
```

P3. 請試著分析 tags 的分布情況(數量)





由上面兩張圖片，我們可以發現 tags 的分布是超級 imbalance 的，超過一半的 tag 出現次數不到 50，因此造成 training 上的瓶頸，大多數的 model 只能 fetch 到比例比較大的幾個 tag。

對於這種分布，如果貿然分成 38 個 model 下去 train 結果會非常的差。在此，我使用了三種對應方式。首先，我調整了不同的 tag 的 class weight，讓 Train NN 跟 LinearSVC 時，可以盡量減少不平衡。再者，對於數量較少的 tag，在 NN 內我調降了 activation 的 threshold，讓他較可能的被激發。最後，因為我有使用到 SVC，加上之前專題教授請我幫忙 review paper 的關係，我想到了 SMOTE，一種在 kernel space 裡 sample 新的 data 以解決 imbalance data 的問題。不過 public set 裡面似乎沒有像去年 final 一樣，有某些 training set 很少的 label 大量出現。

P4. 本次作業中使用何種方式得到 word embedding?請簡單描述做法

在本次作業中，我使用了 glove 為 embedding layer。

Glove (Global Vectors for Word Representation)，綜合了全局統計資訊與局部統計資訊來生成語言模型和詞向量。他利用一個全局 LOG-雙線性回歸模型，主要是 global matrix factorization 和 local context window methods。並且在效能的部分，因為只有利用到 word-word co-occurrence matrix 裡的非零元素，並不是計算整個稀疏矩陣，所以比 word2vec 快上不少。

P5. 試比較 bag of word 和 RNN 何者在本次作業中效果較好

在本次作業中，我的 RNN mode 的效果比 bag of word + SVC 好，分別是 50.5 與 49.3。可能的原因是因為 bag of word 對於字與字之間的連結程度考慮並不多，而 RNN 的 Model 中，因為 RNN 有紀錄的特性，所以可以承接上下的語意。如同老師上

課時的”好棒棒”跟”好棒”，對於 word bagging，兩者的紀錄方式是一樣的，但是對於 RNN，因為有 Memory 的存在，所以兩者的 summary vector 會是不同的。

雖然說二者之中 RNN 的表現比較好，但是兩者答案的表現其實不太一樣，因此透過 ensemble 後，可以獲得頗大幅的提升(53.3)。