

1 (1%)請問softmax適不適合為本次作業的output layer? 寫出你最後選擇的output layer並說明理由。

這次作業是多標籤的分類問題，softmax 函數會使得最多只會有一個類別的機率大於 0.5，因此較為不適合。為了要模型能對於不只一個標籤預測出大於 0.5 的機率，輸出層應該是使用 sigmoid 函數會比較合理。

2 (1%)請設計實驗驗證上述推論

如果使用助教提供的樣本程式碼，F1 score 在自行切分的驗證資料集上，最高只能達到大約 0.29。而使用 sigmoid 的話，使用與樣本程式碼類似的架構的話，則可以得到 0.51 的 F1 score。這顯示了在輸出層使用 sigmoid 函數較為合理。

3 (1%)請試著分析tags的分布情況(數量)

AUTOBIOGRAPHICAL-NOVEL	31
DETECTIVE-FICTION	178
WAR-NOVEL	31
NOVELLA	29
NON-FICTION	102
CHILDREN' S-LITERATURE	777
HISTORICAL-FICTION	137
SUSPENSE	318
GOTHIC-FICTION	12
ROMANCE-NOVEL	157
HIGH-FANTASY	15
HORROR	192
CRIME-FICTION	368
TECHNO-THRILLER	18
ADVENTURE-NOVEL	109
DYSTOPIA	30
AUTOBIOGRAPHY	51
COMIC-NOVEL	37
ALTERNATE-HISTORY	72
UTOPIAN-AND-DYSTOPIAN-FICTION	11
HISTORY	40
NOVEL	992
SPECULATIVE-FICTION	1448
MEMOIR	35
THRILLER	243
SHORT-STORY	41
MYSTERY	642
APOCALYPTIC-AND-POST-APOCALYPTIC-FICTION	14
SCIENCE-FICTION	959
HUMOUR	18
SPY-FICTION	75
FICTION	1672
YOUNG-ADULT-LITERATURE	288
BIOGRAPHY	42
FANTASY	773
COMEDY	59
HISTORICAL-NOVEL	222
SATIRE	35

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

使用 GloVe 已經事先用 Wikipedia 2014 + Gigaword 5 訓練好的 100 維資料。GloVe 的訓練方式是先統計詞彙跟詞彙同時出現次數，計算出一個詞彙出現在另一詞彙的上下文的條件機率，並將這些機率寫成一個矩陣。然後再使用矩陣分解的方式，找出能最小化兩詞彙內積與前述的條件機率之誤差的向量表示法，即得到詞彙們的向量。

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

對於同一組訓練資料與驗證資料，使用 TfIdf 的方法處理 Bag of Word 後，再使用線性支撐向量機對每個類別分別做二元分類，最高可以在驗證資料上得到 0.52 的 F1 score，而使用 RNN 也可以得到 0.52 的 F1 score，因此我認為兩者效果差不多。