

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。

Softmax 並不適合作為本次作業的 output layer，因為本次作業為 multilabel，輸出為一維矩陣，對應每個 label 都有不同的機率，每個 label 的機率之間互相不影響，故選用 sigmoid 會比 softmax 結果的準確率上好很多。

2. (1%)請設計實驗驗證上述推論。

用同樣的程式，將最後的 output layer 分別設為 sigmoid 及 softmax，觀察結果的差別，切出 10% data 為 validation data，在 validation 上 sigmoid 的 f1score 結果可以到 0.5 附近，但若用 softmax 幾乎都在零附近，故明顯看出 softmax 適不適合作為本次作業的 output layer。

3. (1%)請試著分析 tags 的分布情況(數量)。

SCIENCE-FICTION [959.]

SPECULATIVE-FICTION [1448.]

FICTION [1672.]

NOVEL [992.]

FANTASY [773.]

CHILDREN'S-LITERATURE [777.]

HUMOUR [18.]

SATIRE [35.]

HISTORICAL-FICTION [137.]

HISTORY [40.]

MYSTERY [642.]

SUSPENSE [318.]

ADVENTURE-NOVEL [109.]

SPY-FICTION [75.]

AUTOBIOGRAPHY [51.]

HORROR [192.]

THRILLER [243.]

ROMANCE-NOVEL [157.]
COMEDY [59.]
NOVELLA [29.]
WAR-NOVEL [31.]
DYSTOPIA [30.]
COMIC-NOVEL [37.]
DETECTIVE-FICTION [178.]
HISTORICAL-NOVEL [222.]
BIOGRAPHY [42.]
MEMOIR [35.]
NON-FICTION [102.]
CRIME-FICTION [368.]
AUTOBIOGRAPHICAL-NOVEL [31.]
ALTERNATE-HISTORY [72.]
TECHNO-THRILLER [18.]
UTOPIAN-AND-DYSTOPIAN-FICTION [11.]
YOUNG-ADULT-LITERATURE [288.]
SHORT-STORY [41.]
GOTHIC-FICTION [12.]
APOCALYPTIC-AND-POST-APOCALYPTIC-FICTION [14.]
HIGH-FANTASY [15.]

可發現 SCIENCE-FICTION、SPECULATIVE-FICTION NOVEL 等出現機率較高，有大約五個 tag 出現機率不到 1%，發現 tag 出現得機率是極不公平的。

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

套用 glove 的 word embedding，每個字為 100dimension 的矩陣，因為 training data 中的總字數並沒有很多，因此採用 glove.6B.100d.txt，在 train 時發現效果很好，改用 glove.6B.200d.txt 對於結果並沒有太大差別。

以下為 glove model training summary:

Training

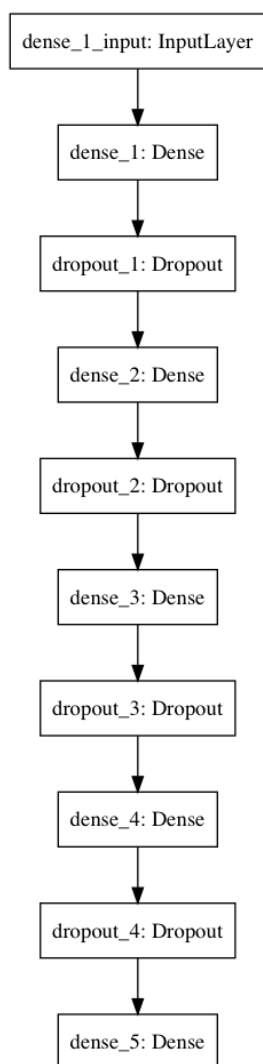
The GloVe model is trained on the non-zero entries of a global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus. Populating this matrix requires a single pass through the entire corpus to collect the statistics. For large corpora, this pass can be computationally expensive, but it is a one-time up-front cost. Subsequent training iterations are much faster because the number of non-zero matrix entries is typically much smaller than the total number of words in the corpus.

The tools provided in this package automate the collection and preparation of co-occurrence statistics for input into the model. The core training code is separated from these preprocessing steps and can be executed independently.

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

Bag of word 在 kaggle 上的 performance 較好，用 rnn+word embedding 的結果在 kaggle 上表線上大約為 0.47 0.48，而 bag of word 在 kaggle 表現上為 0.5 左右，比 rnn 稍微較好一些，我猜測可能原因是在 train data 中的某些字在 embedding 時無法轉換，會轉為一個都為 0 的矩陣，因此造成 rnn 成果較 bag of word 差。

Bag of word model



rnn model

