

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。

我覺得 softmax 並不是合做本次作業的 output layer，因為 softmax 是每個值都是  $\geq 0$ ，並且其總和 1，在輸出時會有 normalize 的動作，會把高輸出的機率拉低，我認為可能是因為輸出有太多種類了，每個平均下來來判斷哪個最高可能會導致最後輸出沒有很準，最後我使用 sigmoid，他在輸出時不會有 normalize，讓原本的比較高機率的輸出不會被拉下來，在做 cross entropy 時會比較能夠真實反映 loss，會比使用 softmax 來得要好。

2. (1%)請設計實驗驗證上述推論。

如果我最後一層使用 softmax 的話，validation 跑出的結果大概是 0.2547

如果我最後一層使用 sigmoid 的話，validation 大概是 0.4922 多，上下這兩個都是用的架構都是一樣的，只有最後一層的 activation 不一樣了，從結果可知，這次作業的 output layer 可能比較適合使用 sigmoid。

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, None, 100)	5186800
bidirectional_1 (Bidirectional)	(None, 256)	175872
dense_1 (Dense)	(None, 128)	32896
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 128)	16512
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 128)	16512
dropout_3 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 64)	8256
dropout_4 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 38)	2470

Softmax	0.2547
sigmoid	0.4922

3. (1%)請試著分析 tags 的分布情況(數量)。

從下圖可以發現，在 train\_data 裡 NOVEL 是數量最多的大概到 400 個，其次是 SCIENCE-FICTION，在來是 FICTION、CHILDREN'S-LITERATURE、FANTASY、FICTION NOVEL、FANTASY SPECULATIVE-FICTION，後面的數量就都比較少了，有些都只有個位數而已



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，GloVe 是以矩陣分解的方式為基礎，通過對包含整個語料統計信息的矩陣進行分解，得到每個單詞對應的實際數量。以 ice 和 stream 為例，先對 ice 和 stream 對某個 K 字取他的 co-occurs 機率，可以找到他們分別對應的關係，可以看到下圖，他們對 solid 和 gas 有相近的機率，但對 water 和 fashion 卻有比較大的差別，我們可以透過對這個差距取 log 來得到相關的資料，再利用這個資料來去找到對應的單詞。

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k ice)$	$1.9 \times 10^{-4}$	$6.6 \times 10^{-5}$	$3.0 \times 10^{-3}$	$1.7 \times 10^{-5}$
$P(k steam)$	$2.2 \times 10^{-5}$	$7.8 \times 10^{-4}$	$2.2 \times 10^{-3}$	$1.8 \times 10^{-5}$
$P(k ice)/P(k steam)$	8.9	$8.5 \times 10^{-2}$	1.36	0.96

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

在架構不變的前提下，可以得到下面的數據，可以發現在同樣的架構下取 validation，RNN 會比 bag of word 表現得還要更好，

bag of word	0.3547
RNN	0.4872