

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。

使用 softmax 會突顯預測 tag 結果之間的差異，強調最有可能的 tag 的值，這樣會造成被選擇到的 tag 可能只有一個，但是這次次的作業，預測的 tag 通常不只一個，所以顯然 softmax 不適用於此 output layer。

最後我選擇是用 sigmoid 作為我的 output layer，因為使用 sigmoid 的話，輸出的值會介於 0~1，可以將其視為每個 tag 可能的機率，因此我只要設定 threshold 值，就可以輸出多個可能的 tag 當成預測結果。

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 306, 100)	5186700
lstm_1 (LSTM)	(None, 128)	117248
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 512)	66048
dropout_2 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 512)	262656
dropout_3 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 512)	262656
dropout_4 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 512)	262656
dropout_5 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 38)	19494
Total params: 6,177,458		
Trainable params: 990,758		
Non-trainable params: 5,186,700		

此為我所使用的 model 架構

2. (1%)請設計實驗驗證上述推論。

在同樣的 model 架構下，僅改動 output layer 的結果：

id	tags	id	tags
1	0 CHILDREN'S-LITERATURE	1	0 SPECULATIVE-FICTION FICTION FICTION FANTASY CHILDREN'S-LITERATURE
2	1 NOVEL	2	1 FICTION NOVEL
3	2 NOVEL	3	2 FICTION NOVEL
4	3	4	3 FICTION NOVEL HISTORICAL-FICTION HISTORICAL-NOVEL
5	4 NOVEL	5	4 FICTION NOVEL
6	5 NOVEL	6	5 FICTION NOVEL
7	6 NOVEL	7	6 FICTION NOVEL
8	7 SPECULATIVE-FICTION	8	7 SCIENCE-FICTION SPECULATIVE-FICTION FICTION FANTASY
9	8 SCIENCE-FICTION SPECULATIVE-FICTION	9	8 SCIENCE-FICTION SPECULATIVE-FICTION
10	9	10	9 SCIENCE-FICTION NON-FICTION
11	10 SCIENCE-FICTION	11	10 SCIENCE-FICTION SPECULATIVE-FICTION FICTION
12	11 FICTION	12	11 FICTION NOVEL
13	12 NOVEL	13	12 FICTION NOVEL
14	13 SCIENCE-FICTION	14	13 SCIENCE-FICTION SPECULATIVE-FICTION
15	14 NOVEL	15	14 FICTION NOVEL ROMANCE-NOVEL
16	15	16	15 FICTION NOVEL
17	16 NOVEL	17	16 FICTION NOVEL
18	17	18	17 SCIENCE-FICTION NOVEL
19	18 SPECULATIVE-FICTION	19	18 SCIENCE-FICTION SPECULATIVE-FICTION FICTION FANTASY
20	19 SPECULATIVE-FICTION	20	19 SPECULATIVE-FICTION FANTASY
21	20 SCIENCE-FICTION	21	20 SCIENCE-FICTION SPECULATIVE-FICTION CHILDREN'S-LITERATURE
22	21 SCIENCE-FICTION	22	21 SCIENCE-FICTION SPECULATIVE-FICTION FICTION
23	22 NOVEL	23	22 FICTION NOVEL
24	23 NOVEL	24	23 FICTION NOVEL

左圖為使用 softmax，右圖為 sigmoid。

很明顯可以看到，使用 softmax 的預測的結果，大多只有一個 tag，所以造成與預期多個 tags 的結果有落差。

1	FICTION	1672
2	SPECULATIVE-FICTION	1448
3	NOVEL	992
4	SCIENCE-FICTION	959
5	CHILDRENS-LITERATURE	777
6	FANTASY	773
7	MYSTERY	642
8	CRIME-FICTION	368
9	SUSPENSE	318
10	YOUNG-ADULT-LITERATURE	288
11	THRILLER	243
12	HISTORICAL-NOVEL	222
13	HORROR	192
14	DETECTIVE-FICTION	178
15	ROMANCE-NOVEL	157
16	HISTORICAL-FICTION	137
17	ADVENTURE-NOVEL	109
18	NON-FICTION	102
19	SPY-FICTION	75
20	ALTERNATE-HISTORY	72
21	COMEDY	59
22	AUTOBIOGRAPHY	51
23	BIOGRAPHY	42
24	SHORT-STORY	41
25	HISTORY	40
26	COMIC-NOVEL	37
27	SATIRE	35
28	MEMOIR	35
29	WAR-NOVEL	31
30	AUTOBIOGRAPHICAL-NOVEL	31
31	DYSTOPIA	30
32	NOVELLA	29
33	HUMOUR	18
34	TECHNO-THRILLER	18
35	HIGH-FANTASY	15
36	APOCALYPTIC-AND-POST-APOCAL	14
37	GOTHIC-FICTION	12
38	UTOPIAN-AND-DYSTOPIAN-FICTION	11

3. (1%)請試著分析 **tags** 的分布情況(數量)。

全部共有 38 種不同的 tags

Tag 出現最多次的前三名分別為：

FICTION：出現 1672 次，約佔全部的 34%

SPECULATIVE-FICTION：出現 1448 次，約佔全部的 29%

NOVEL：出現 992 次，約佔全部的 19%

最少的為：

UTOPIAN-AND-DYSTOPIAN-FICTION：出現 11 次，約佔全部的 0.2%

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

我是使用 GloVe，而它是 count-based 的 model。

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

我使用 Bag of word 的效果比 RNN 來得顯著，推測原因可能是因為每筆資料所給的文章長短不一，對於 RNN 來說，有些比較短的文章無法得到足夠的前後文來判斷。

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 480)	24896640
dropout_1 (Dropout)	(None, 480)	0
dense_2 (Dense)	(None, 512)	246272
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 512)	262656
dropout_3 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 512)	262656
dropout_4 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 512)	262656
dropout_5 (Dropout)	(None, 512)	0
dense_6 (Dense)	(None, 38)	19494
Total params: 25,950,374		
Trainable params: 25,950,374		
Non-trainable params: 0		

以上是我在 bag of word 使用的 model，在 kaggle 中最好的分數為：

Public：0.51942，private：0.49662。

Bag of word 單純統計各種單字出現的次數，某種程度上比較不像 RNN 受限於文章的長短，所以 train 起來效果比較顯著，但準確率大多落在 0.50 左右，有點難以突破。