

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。

不一定，這應該要取決於output是哪種類型，以這次的實驗來說，它的理想output是38維的0,1，predict的值介於0~1之間，如果predict出來的值大於特定的門檻就設為1，小於就設為0。然而，以softmax當作output layer的話每個維度的值都會很小，根據 $F(x_j) = \frac{e^{x_j}}{\sum_{i=1}^N e^{x_i}}$ 的公式，我們不知道要設定怎麼樣的門檻。然而，若有其他的方法先rescale，softmax也許是可行的，例如說從predictk的38維裡面找出最大的維度做基準，其他的37維對此normalize，並規定值> 0.5設為1，< 0.5就為0。

我是用sigmoid function，因為在訓練上有較佳的f1_score，再者，sigmoid function比較好調整門檻，通常f1_score的最佳值都落在門檻 = 0.5左右。(我自己突破strong baseline的做法也是在門檻上找最佳值逼近)

2.(1%)請設計實驗驗證上述推論。

我用同樣的架構去訓練，唯一的差別只在於output的activation一個是softmax一個是sigmoid。比較在training完之後的f1_score。

架構描述: 一層GRU(unit = 256)，dropout = 0.2，recurrent dropout = 0.2，epoch = 70

Activation	Accuracy
Sigmoid	Validating Score = 0.5109
Softmax	Validating Score = 0.4758

3.(1%)請試著分析tags的分布情況(數量)。

38個 Tags的分布數量如下：

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

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

tokenizer + pad_sequence + glove embedding

首先，我先用keras的tokenizer把文字轉換成對應的數字，每個字都有獨特的對應數字(ex: the = 1233, a = 488)，因為training + testing 的文字種類有差不多51000個，最後我編的數字就有51000種。接著，我用pad_sequence將每段文字都填補到相同的長度，較短的文字直接在前面補0。最後，我用網路上已經train好的100維glove embedding dict產生word embedding，這個glove在做的事情大致是把前後連

結、字意或用法相似的字轉成空間相關的向量，保存這些字的意涵跟關係，接下來，我們再把這個產生的embedding 放進RNN的model裡面，training的時候他就會找到對應的100維embedding 向量了。

5.(1%)試比較bag of word和RNN何者在本次作業中效果較好。

我訓練的結果是RNN(GRU)較好，在把4個最好的Model混合訓練之下，最高的Public Testing Set f1_score達到0.539，單一Model最高可到0.523。而bag of words 的分數最高只到0.508。

右圖是我成效最好的bag of words model架構，這三層的Unit數個別是512,256,128，每層都有加入Batch Normalization跟Dropout (0.3)。另外，我也有試過2層的model、調整drop跟layer units參數，但成效都沒有右圖這個結果好。

