Machine Learning 2017 HW5 Report

Multi-class & multi-label article classification

B03901156 Yu Xuan Huang 黃于瑄

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。

Structure:

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 306, 100)	5186700
gru_1(GRU)	(None, 256)	393984
dense_1 (Dense)	(None, 256)	65792
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
dropout_3 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
dropout_4 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 38)	2470
# Compile the model		
adagrad = Adagrad (lr=0.01, epsilon=1e-08, decay=0.0) model.compile (loss='categorical_crossentropy', optimizer=adagrad, metrics=[f1_score])		
Total params: 5,964,018 Trainable params: 777,318 Non-trainable params: 5,186,700		

```
### build model
print ('Building model.')
model = Sequential()
model.add(Embedding(num_words,
                    embedding_dim,
                    weights=[embedding_matrix],
                    input_length=max_article_length,
                    trainable=False))
model.add(GRU(256,activation='tanh',dropout=0.3))
model.add(Dense(256,activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(64,activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(38,activation='sigmoid'))
model.summary()

#adam = Adam(lr=0.001,decay=1e-6,clipvalue=0.5)
adagrad = Adagrad(lr=0.01, epsilon=1e-08, decay=0.0)
model.compile(loss='categorical_crossentropy',
              optimizer=adagrad,
              metrics=[f1_score])
```

```
earlystopping = EarlyStopping(monitor='val_f1_score', patience = 20, verbose=1, mode='max')
checkpoint = ModelCheckpoint(filepath='best.hdf5',
                             verbose=1,
                             save_best_only=True,
                             save_weights_only=True,
                             monitor='val_f1_score',
                             mode='max')
hist = model.fit(X_train, Y_train,
                 validation_data=(X_val, Y_val),
                 epochs=nb_epoch,
                 batch_size=batch_size,
                 callbacks=[earlystopping,checkpoint])
```

Considering you have a vector x of K outcomes:

Softmax function: $f(x_i) = \frac{exp(x_i)}{\sum_{j=0}^{K} exp(x_j)}, i = 0..K$

Sigmoid function: $f(x_i) = \frac{1}{1 + exp(-x_i)} \forall i$

在這個 model 中我選擇以 sigmoid function 作為 output layer 的 activation function.
由上式可知，sigmoid function 會將一個實數值(real value)映射到(0,1)的區間中，而 softmax 則將一個 k 維的實數向量(a1,a2,a3,a4…)映射成(b1,b2,b3,b4…)，其中 bi 是介於 0~1 的常數，而輸出神經元的總合為 1.0，bi 的數值相當於概率值，可根據 bi 的數值大小進行多分類(multi-class)的 task.
由於 softmax function 所進行的多分類(multi-class)任務 class 與 class 間是互斥的，亦即一個 input 只能被歸為一個 class，而 sigmoid function 的 class 和 class 間可以互相重疊，一個 input 可被歸類為多個 class；故考慮此次作業的目標，sigmoid function 比 softmax function 更適合 output layer 的 activation function.

2. 請設計實驗驗證上述推論。

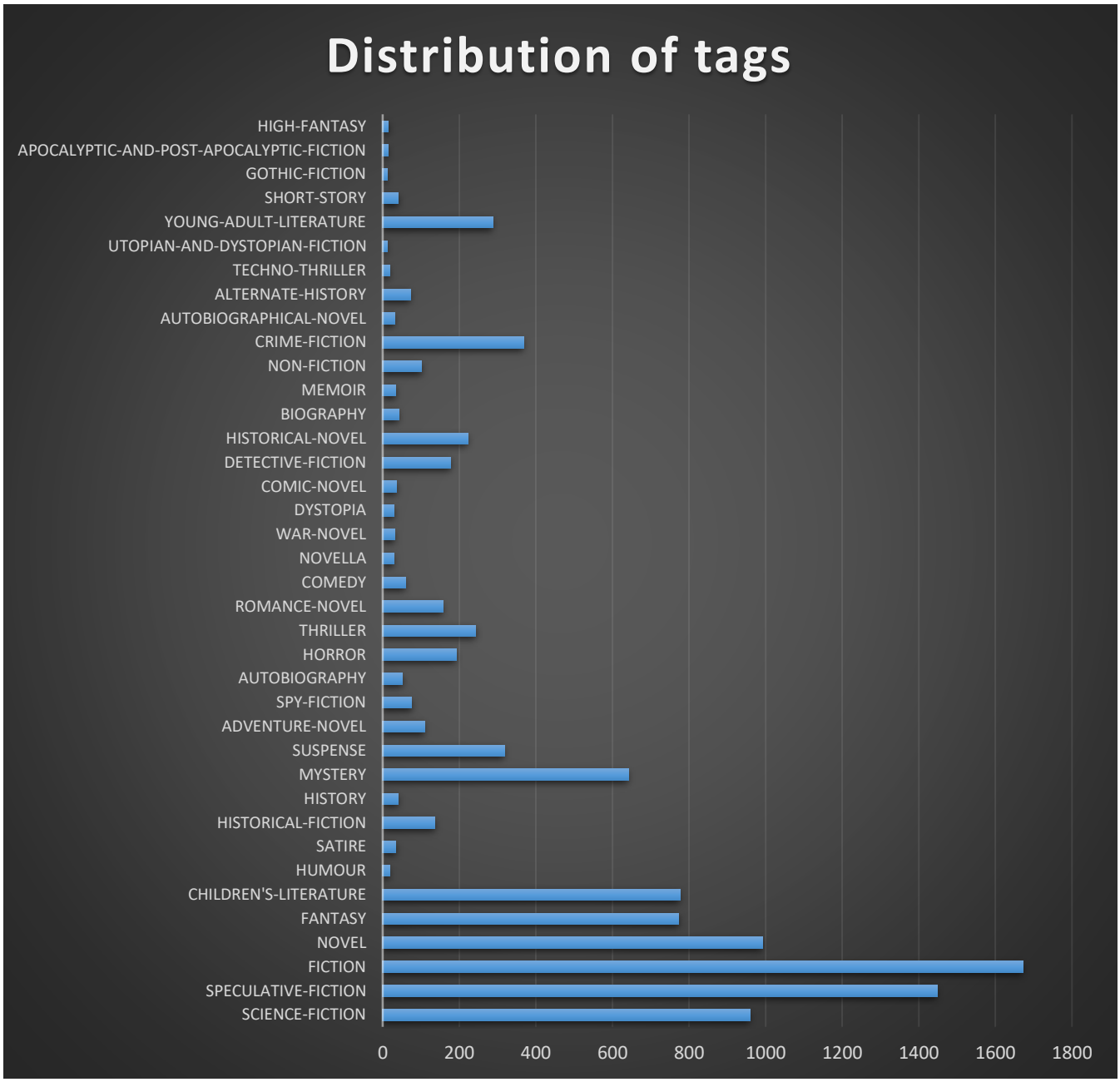
實驗設計: 兩組 model 皆採用 1.中的 structure，僅更改 output layer 的 activation function 分別為 softmax 及 sigmoid，分析在 training 過程中(前 10 個 epoch 中) validation set 的 f1 score 差異

實驗結果:

	softmax	**sigmoid**
Epoch 1	0	0.2431
Epoch 2	0	0.2772
Epoch 3	0	0.2883
Epoch 4	0	0.2945
Epoch 5	0	0.3080
Epoch 6	0	0.3203
Epoch 7	0	0.3382
Epoch 8	0	0.3422
Epoch 9	0.0010	0.3614
Epoch 10	0.0272	0.3529

由實驗結果可以看出在前 10 個 epoch 中，以 sigmoid 為 output layer activation function 的 model 其 validation set f1 score 優於以 softmax 為 output layer activation function 的 model；f 且後者於 epoch 9 時 f1 score 才開始緩慢成長，由此可驗證 softmax function 於此次作業中表現較 sigmoid function 來得差，較不適合做為本次作業 output layer 的 activation function.

3. 請試著分析 tags 的分布情況(數量)。



從上述分布圖中觀察可知在 training data 中數目較多的 tags 為 FICTION, SPECULATIVE-FICTION, SCIENCE-FICTION, NOVEL, FANTASY, CHILDREN'S-LITERATURE 和 MISTERY；而從 training data 的數個 articles 觀察可知這些數目較多的 tags 傾向於被同一 article 所標記；而在最後的 prediction(by RNN)結果中也可觀察出此現象。

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

在本次作業中，我使用了 glove.6B.100d.txt 作為 word embedding, 其做法簡單描述如下:
GloVe 是 unsupervised learning, log-bilinear model，用以觀察詞與詞的共現率以衡量對應的涵義並獲取詞向量表示；GloVe 會對語料進行詞與詞的矩陣分解從而得到詞表示的方法: 矩阵第 ii 行第 jj 列的值为词 vivi 与词 vjvj 在语料中的共现次数 xijxij 的对数，而該值與詞向量的空間差異相關聯。而 glove.6B.100d.txt 則是在 2014 年的英文维基百科上基於此共現矩陣分解所训练的詞向量，有 400k 个不同的词，每个词以 100 维向量表示。

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

RNN model with the structure the same as 1.: 0.48663 on kaggle
bag of word model: 0.51417 on kaggle (though post deadline)
從上述結果可得知， bag of word 的 model 可得到較好的結果，推測其因為本次作業的 article 其 tag 的標記可能與 tag 對應的 keyword 有關，當特定 keyword 出現在 article 中的頻率較高時，即判定該文章必須擁有與該 keyword 相對應的 tag. 而 bag of word model 正是利用詞頻的特徵來 predict testing data 所屬的 tag，故其相較於 RNN model 在本次作業中效果較好。