學號 :R05921086 系級：電機碩一 姓名 :邱名彥

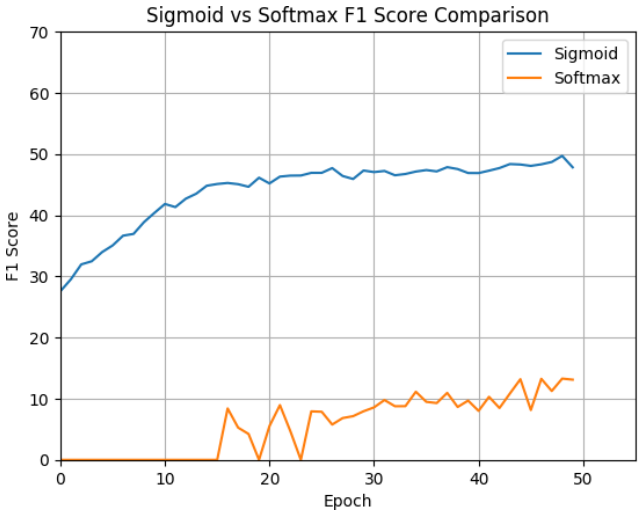
1. (1%)請問 softmax 適不適合作為本次作業的 output layer? 寫出你最後選擇的 output layer 並說明理由。

A softmax function is more suited for binary cases where there is only one output node. It gives a probability distribution centered around ‘n’ number of classes as output. For multi-class and multi-label article classification, a softmax function would function poorly, if at all, and give significantly low accuracy. A sigmoid activation layer would function better as it operates independently on each node. This allows multi-class and multi-label classification which gives significantly improved results over a softmax activation layer.

Shown below are the relevant activation functions

$P(y=j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\mathsf{T}\mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\mathsf{T}\mathbf{w}_k}}$	$S(x) = \frac{1}{1+e^{-x}}.$
Softmax Activation Function	Sigmoid Activation Function

2. (1%)請設計實驗驗證上述推論。



Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 190, 100)	5173100
gru_1 (GRU)	(None, 128)	87936
dense_1 (Dense)	(None, 256)	33024
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
dropout_3 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 38)	2470

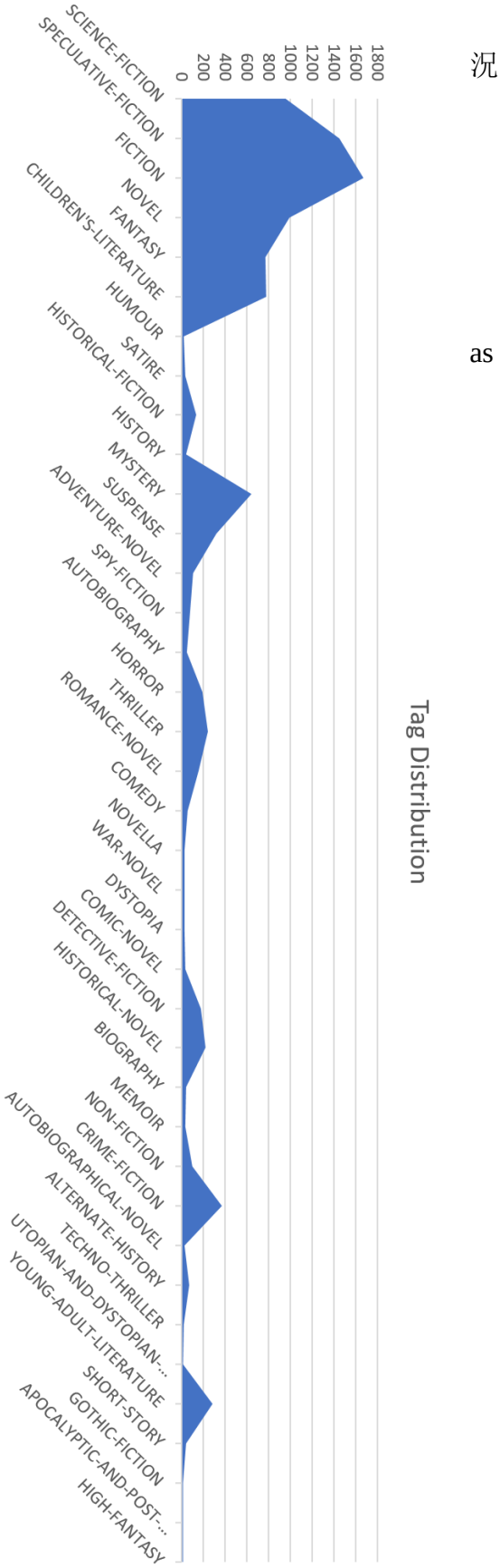
Total params: 5,337,682
Trainable params: 164,582
Non-trainable params: 5,173,100

From the graph shown in the above left, we can see that using a sigmoid activation layer provides significantly better accuracy in terms of validation F1 Score. It can be observed that using a sigmoidal activation layer gives an maximum accuracy of 50% while the softmax activation layer gives an accuracy of 14% at best. The RNN model summary is given in the top right.

F1 Score has been scaled up by 100

3. (1%)請試著分析 tags 的分布情況(數量)。

The tags number 10273 in total. There are 38 unique tags in total. The tags with the most count are ‘Speculative-fiction’, ‘Science-Fiction’ and ‘Children’s Literature’ which number 1448, 959, and 777 respectively. The genres with the lowest count include tags such as ‘Gothic Fiction’, High-Fantasy’, and ‘Humour’ which do not number past 20.



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

Word Embedding converts text into numbers. This allows machine learning algorithms (such as natural language processing) to map words or phrases to a corresponding vector of real numbers. This also reduces total dimensionality as words which must be represented by multiple characters can be expressed by a single integer value.

The GloVe algorithm is a method that converts text to a vector representation which can be generalized in 3 steps:

1. Collect word occurrences in a co-occurrence matrix X. Each element in the word co-occurrence matrix represents how often a word *i* appears in context of word *j*. For each term, we search for context terms defined by a window_size before and after the term. A decay function is used to give less weight for more distant words, usually taking some form of the following:

$$decay = 1/offset$$

2. Define soft constraints for each word pair

$$w_i^T w_j + b_i + b_j = log(X_{ij})$$

3.

W_i is the vector for the main word

W_j is the vector for a context word

b_i and b_j are biases for the main and context words

4. Define a cost function where f is the weighting function to prevent the model from learning common words taking the general form below:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - \log X_{ij})^2$$

GloVe usually uses the weighting function shown below:

$$f(X_{ij}) = \begin{cases} (\frac{X_{ij}}{x_{max}})^{\alpha} & \text{if } X_{ij} < XMAX \\ 1 & \text{otherwise} \end{cases}$$

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

The graph below compares the F1 Score Validation (Scaled up by 100) between an RNN network (with GloVe Word Embedding) vs a Bag of Words network.

Bag of Words operate similarly to RNN networks in that Bag of Words tries to simplify a file/sentence, which is considered a bag, of its words. Word order and grammar is ignored. Input words are sampled and are compared against repositories of other bags to see if it is a likely fit.

Recurrent Neural Networks (RNN) uses filters such as LSTM or GRU layers in addition to word embedding. It tries to create word associations based on the word embedder and the layers in the RNN try to create a prediction.