

# Homework 4 Report - Malicious Comments Identification

學號：B05505004 系級：工海三 姓名：朱信靂

## 1 (1%)

- (0.5%) 請說明你實作之RNN 模型架構及使用的word embedding 方法，回報模型的正確率並繪出訓練曲線。
- (0.5%) 請實作BOW+DNN 模型，敘述你的模型架構，回報正確率並繪出訓練曲線。

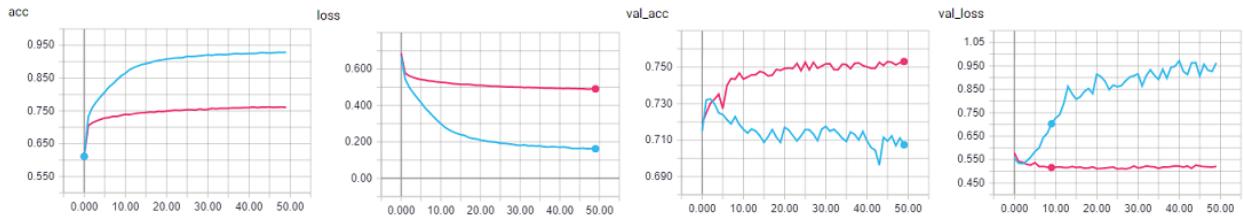
Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 64, 256)	30477056
gru_1 (GRU)	(None, 64, 256)	393984
batch_normalization_1 (Batch Normalization)	(None, 256)	1024
gru_2 (GRU)	(None, 512)	1181184
batch_normalization_2 (Batch Normalization)	(None, 512)	2048
dense_1 (Dense)	(None, 256)	131328
batch_normalization_3 (Batch Normalization)	(None, 256)	1024
leaky_re_lu_1 (LeakyReLU)	(None, 256)	0
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
batch_normalization_4 (Batch Normalization)	(None, 128)	512
leaky_re_lu_2 (LeakyReLU)	(None, 128)	0
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
batch_normalization_5 (Batch Normalization)	(None, 64)	256
leaky_re_lu_3 (LeakyReLU)	(None, 64)	0
dropout_3 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 1)	65
Total params:	32,229,633	
Trainable params:	1,750,145	
Non-trainable params:	30,479,488	

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	2505216
batch_normalization_1 (Batch Normalization)	(None, 256)	1024
leaky_re_lu_1 (LeakyReLU)	(None, 256)	0
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 512)	131584
batch_normalization_2 (Batch Normalization)	(None, 512)	2048
leaky_re_lu_2 (LeakyReLU)	(None, 512)	0
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 256)	131328
batch_normalization_3 (Batch Normalization)	(None, 256)	1024
leaky_re_lu_3 (LeakyReLU)	(None, 256)	0
dropout_3 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 128)	32896
batch_normalization_4 (Batch Normalization)	(None, 128)	512
leaky_re_lu_4 (LeakyReLU)	(None, 128)	0
dropout_4 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 64)	8256
batch_normalization_5 (Batch Normalization)	(None, 64)	256
leaky_re_lu_5 (LeakyReLU)	(None, 64)	0
dropout_5 (Dropout)	(None, 64)	0
dense_6 (Dense)	(None, 1)	65
Total params:	2,814,209	
Trainable params:	2,811,777	
Non-trainable params:	2,432	

### o Model

上圖左邊是我的RNN model，我的model是會先過兩層GRU，輸出的向量分別是256以及512。接著會在經過三層輸出分別是145、128、64的全連接層。另外在每層的中間我都有加上dropout以及batch\_normalization，並且全連接層都是使用leaky relu當作activation function。而word embedding則是用最基本的先jieba切完詞後把非英文、中文的詞濾掉，接著再丟進word2vec的model中訓練iter = 50，其他min\_count及windows都是採用預設值。

而上面右圖則是我的BOW+DNN的model架構。為了確保比較的公平性，我預處理是採用同樣的方法，斷詞後再把非英文、中文的詞濾掉，然而因為所占空間的關係，我只保留詞出現次數超過20次的詞去做BOW。model的架構的部分則是將原本RNN的embedding layer拔掉，並將兩層GRU換成全連接層。



### ◦ 訓練曲線

上面的訓練曲線中藍色是BOW+DNN，粉紅色的則是RNN，由左至右分別是training的accuracy、loss，以及validation的accuracy、loss。

可以看到相較於RNN，DNN他的training準確率是一直上升，甚至有高達95%的準確率，可是在validation的部分他大概在3個epoch左右就達到的最高峰。可以看到明顯的overfit在training data上，而且我的dropout都已經全部調到0.5還是如此嚴重。然而雖然DNN的參數比較多，可是只需要14分鐘的training時間，相較於RNN可說是少了許多，而且從validation上來說準確率也僅僅掉了2%左右。

## 2 (1%)

- 請敘述你如何improve performance (preprocess, embedding, 架構等)，並解釋為何這些做法可以使模型進步。

在改進我的model的上面我是從以下幾點來去改進的

### 1. 預處理

因為這個dataset其實有蠻多奇怪的資訊，像是表情符號以及通常回復別人會加的 `B01` 等等。雖然像表情符號或多或少會跟惡意留言有關係，可是其數量並不是太多，很容易會混淆model而學不到最重要的東西。而過濾掉這些和預測無關的輸入可以幫助model專注在有幫助的地方上，這也是為什麼這樣的預處理可以改善model。

### 2. 架構

在架構的部分呢我常識過LSTM、Bidirectional、不同的全連接層層數以及參數數量。最後才選定是現在這個model的架構，LSTM跟Bidirectional很容易會因為參數較多而產生overfit的問題，而全連接層的化則是三層比兩層好，三層的參數逐漸縮減又比三層的參數相同，或是中間比上下兩層有較少參數來的好。

RNN在前半部分我認為還是在扮演encoder的角色，encode出句子的意思。如果參數太多training data太少的話很容易會學不到零散、沒有意義的東西，而對於後面的全連接層也是同樣的道理。也因此選擇適合的架構、參數會對performance有根本上的幫助。

### 3. 初始化

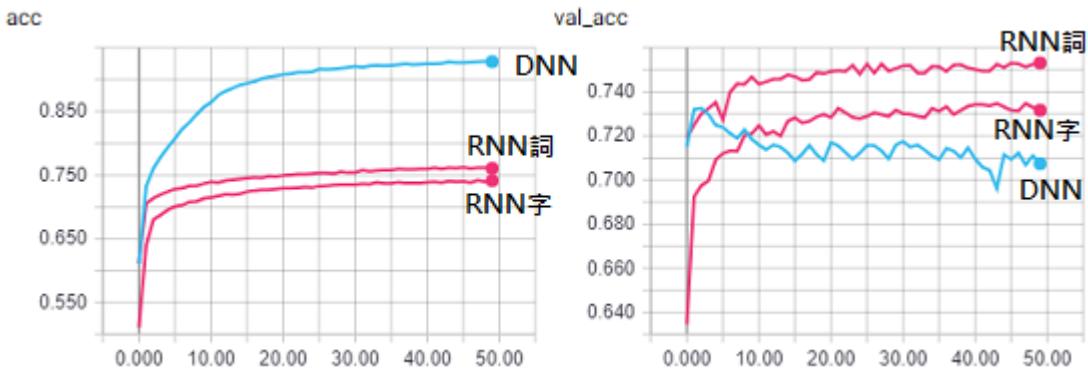
除了上述外我還嘗試了不同的初始化方式，包括GRU的weight用glorot\_normal、he\_normal或Orthogonal來初始化以及bias用0或1初始化。實驗的結果我最後使用Orthogonal，bias用1來初始化。而Orthogonal初始化的好處在於Orthogonal的weight matrix對於eigenvalue等會有一些特別的性質存在，而這個性質保障了RNN不會遇到gradient exploding或是gradient vanishing的問題。

而bias設成1相較於0最大的好處是他讓forget gate在一開始train會是打開的，它可以讓model記住跟多前面的資訊而不是因為初始化的關係全部都忘記。這也造成如果bias設成1，在訓練曲線上的第一個點都比設成0來的準確率高、loss低，並且也更快達到收斂。

綜合以上的敘述可以知道好的初始化可以幫助model更快達到收斂，並且可以避免掉一些潛在的問題。

## 3 (1%)

- 請比較不做斷詞(e.g., 以字為單位) 與有做斷詞，兩種方法實作出來的效果差異，並解釋為何有此差別。



上圖中RNN詞代表我是用jieba以詞為單位去做斷詞，而RNN字則是代表我直接用字去單位斷詞。可以看到兩者在經過同樣的預處理後與DNN一起做比較所話出來的training及validation的準確率圖。

可以看到相較於以詞為單位的RNN，以字為單位的RNN在兩個的準確率都較低，我認為是因為中文的語意單位其實是詞，縱使有兩個詞的開頭都是同樣的字，可是也很容易因為接下字的不同行程差別十萬八千里的意思，這就是為什麼以字為單位會有較低的準確率的原因。可是從我剛剛講的理由可以發現其實也有一點時續的觀念在裡面，就是一個字會因為後面出現的字的不同而有不同的意思。這也是為什麼以字為單位切的準確率較低，可是還是有維持一定的準確率。

#### 4(1%)

- 請比較RNN 與BOW 兩種不同model 對於“在說別人白痴之前，先想想自己”與“在說別人之前先想想自己，白痴”這兩句話的分數（model output），並討論造成差異的原因。

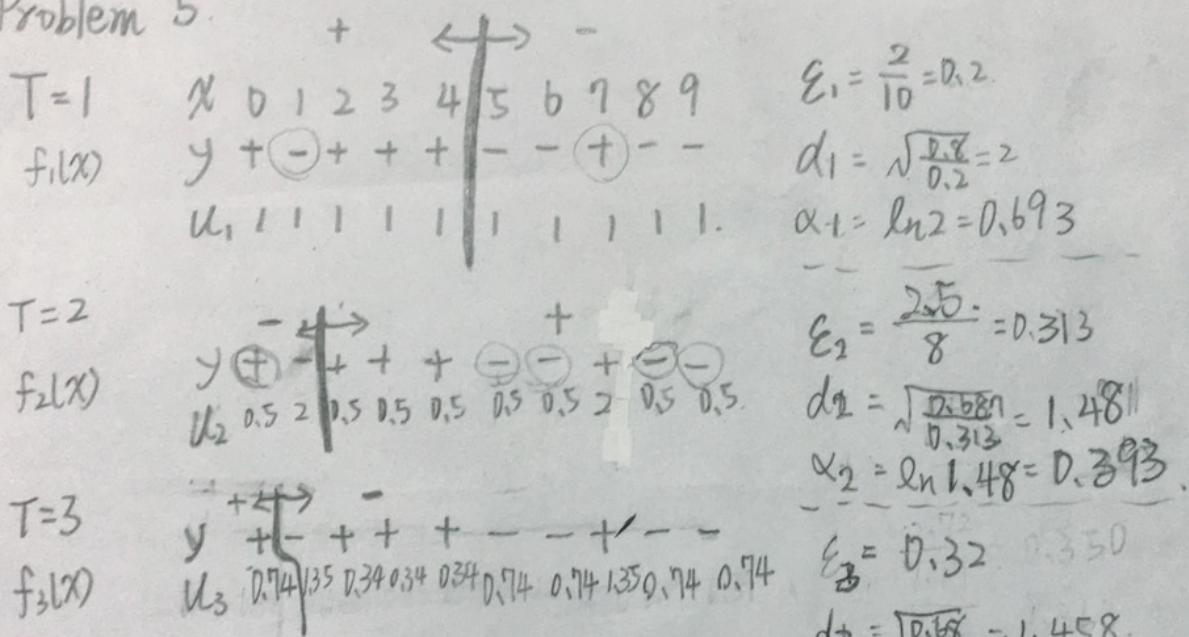
句子	RNN	DNN+BOW
在說別人白痴之前，先想想自己	0.38521773	0.4208312
在說別人之前先想想自己，白痴	0.5270581	0.4208312

上面的數字分別代表兩個model對於這兩個句子覺得是惡意留言的分數。可以看到對RNN來說她覺得第二個句子相較第一個句子具有較多的惡意成分在裡面，而DNN則因為是使用BOW的關係所以因為兩者詞相同而有了相同的分數。

會有這樣的差別我認為是因為RNN model其實有學到同一個詞在不同位置的意義，像是白癡在第二個句子就有指稱別人的意思。而BOW則完全不管詞的順序，而從分數來看DNN應該是有抓到白癡具有罵人的意思，所以分數較RNN的第一個句子高，可是因為沒有順序的觀念，所以也無法對不同順序的句子產生不同的分數。

#### 5(1%)

Problem 5.



final classifier:  $f(x) = 0.693f_1(x) + 0.393f_2(x) - 0.377f_3(x)$

6(1%)

Problem b.

$$t=1, z = 1 \times 3 + 0 = 3 \quad g(z) = 3 \quad C^1 = 1 \times 3 + 0 = 3$$

$$z_{\bar{x}} = 100 \times 1 - 10 = 90 \quad f(z_{\bar{x}}) = \frac{1}{1+e^{-90}} = 1 \quad y^1 = 0 \cdot 3 = 0 \quad *$$

$$z_f = -100 + 110 = 10 \quad f(z_f) = 1 \quad = 2+$$

$$z_0 = 0 - 10 = -10 \quad f(z_0) = 0$$

$$t=2, z = 1 \times (-2) = -2 \quad g(z) = -2 \quad C^2 = 1 \times (-2) + 3 \times 1 = 1$$

$$z_{\bar{x}} = 1 \times 100 - 10 = 90 \quad f(z_{\bar{x}}) = 1 \quad y^2 = 1 \times 1 = 1 \quad *$$

$$z_f = 1 \times (-100) + 110 = 10 \quad f(z_f) = 1$$

$$z_0 = 1 \times 100 - 10 = 90 \quad f(z_0) = 1$$

$$t=3, z = 4 \times 1 + 0 = 4 \quad g(z) = 4 \quad C^3 = 1 \times 4 + 1 \times 0 = 4$$

$$z_{\bar{x}} = 100 + 100 - 10 = 190 \quad f(z_{\bar{x}}) = 1$$

$$z_f = -100 - 100 + 110 = -90 \quad f(z_f) = 0$$

$$z_0 = 100 - 10 = 90 \quad f(z_0) = 1$$

$$t=4, z = 0 \quad f(z) = 0 \quad C^4 = 1 \times 0 + 4 \times 1 = 4$$

$$z_{\bar{x}} = 100 - 10 = 90 \quad f(z_{\bar{x}}) = 1$$

$$z_f = -100 + 110 = 10 \quad f(z_f) = 1$$

$$z_0 = 100 - 10 = 90 \quad f(z_0) = 1$$

$$t=5, z = 0 + 2 = 2 \quad f(z) = 2 \quad C^5 = 1 \times 2 + 4 \times 1 = 6$$

$$z_{\bar{x}} = 100 - 10 = 90 \quad f(z_{\bar{x}}) = 1$$

$$z_f = -100 + 110 = 10 \quad f(z_f) = 1$$

$$z_0 = 0 - 10 = -10 \quad f(z_0) = 0$$

$$t=6, z = -4 = -4 \quad f(z) = -4 \quad C^6 = 0 \times (-4) + 6 \times 1 = 6$$

$$z_{\bar{x}} = 0 - 10 = -10 \quad f(z_{\bar{x}}) = 0$$

$$z_f = 6 + 10 = 16 \quad f(z_f) = 1$$

$$z_0 = 100 - 10 = 90 \quad f(z_0) = 1$$

$$t=7, z = 1 = 1 \quad f(z) = 1 \quad C^7 = 1 \times 1 + 6 \times 0 = 1$$

$$z_{\bar{x}} = 100 + 100 - 10 = 190 \quad f(z_{\bar{x}}) = 1$$

$$z_f = 100 - 10 = 90 \quad f(z_f) = 1$$

$$z_0 = 100 - 10 = 90 \quad f(z_0) = 1$$

$$y_7 = 1 \times 1 = 1$$

$$z_f = -100 - 100 + 110 = -90 \quad f(z_f) = 0$$

$$z_0 = 100 - 10 = 90 \quad f(z_0) = 1$$

$$t=8, z = 2 \quad f(z) = 2 \quad C^8 = 1 \times 2 + 1 \times 1 = 3$$

$$z_{\tilde{x}} = 100 - 10 = 90 \quad f(z_{\tilde{x}}) = 1 \quad y_8 = 1 \times 3 = 3.$$

$$z_f = -100 + 110 = 10 \quad f(z_f) = 1$$

$$z_0 = 100 - 10 = 90 \quad f(z_0) = 1$$