

Homework 4 Report –

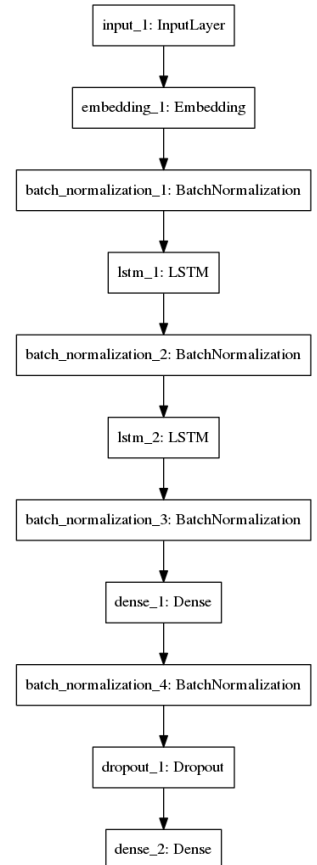
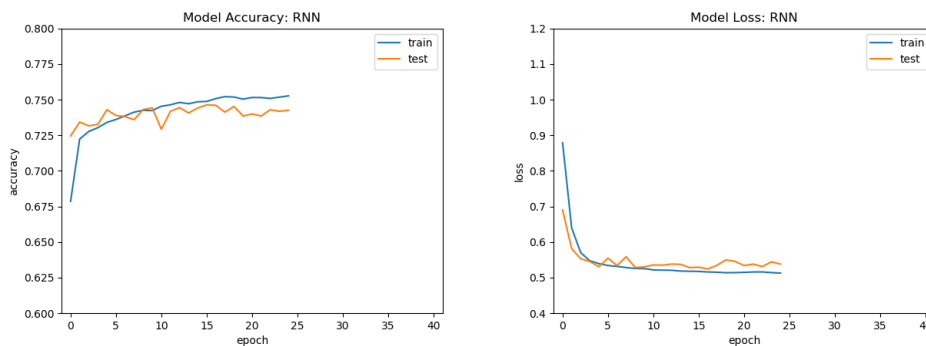
Malicious Comments Identification

學號：R06521504 系級：土木所交通組碩二 姓名：陳譽仁

Problem 1.

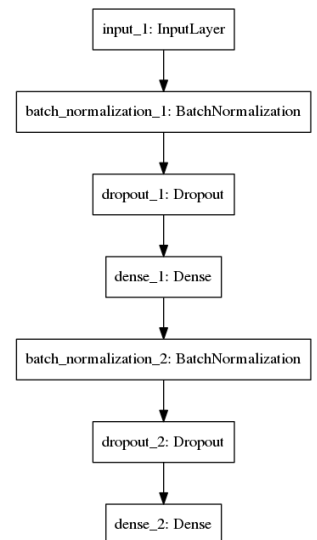
RNN:

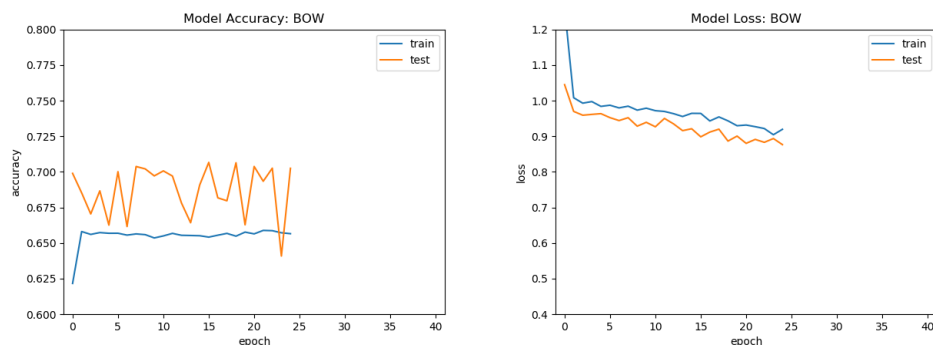
本次作業所建構出的 RNN model 架構如右圖，分別有一層關掉 trainable 的 embedding layer、兩層 512 單位的 LSTM、一層 256 個 nodes 的 dense 層以及一個輸出層，中間再穿插 BatchNormalization 與 Dropout 層，其中共有 13,420,545 個參數，而可訓練的有 3,808,769 個。處理文字的部分則是先讀入文字、使用 jieba 斷詞、去除停止詞、用 word2vec 訓練一個 word vector，其中選擇 skip-gram 的方式，而將句子輸入模式時使用 **PaddingLength = 60**。訓練模式時，選擇 adam(lr=0.002)做為 optimizer，batch_size 設為 512，validation set 為 training dataset 的 0.3，讓 model 自己去切。以這個架構在 Kaggle 上的分數為 **0.74367 (Private)**、**0.74450 (Public)**，訓練過程如下圖，可以發現過程中有一些 overfitting 的情況，可是並不明顯。



BOW+DNN:

本次作業所建構出的 BOW+DNN model 架構如右圖，分別有一層 64 個 nodes 的 dense 層以及一個輸出層，中間再穿插 BatchNormalization 與 Dropout 層，其中共有 2,552,697 個參數，而可訓練的有 2,477,501 個。處理文字的部分則是先讀入文字、使用 jieba 斷詞、去除停止詞、用 word2vec 訓練一個 word vector，其中選擇 skip-gram 的方式，而將句子輸入模式前選擇 **PaddingLength = 60**，將每個詞轉換為 index 之後再轉換為 BOW 的資料形式，總共記錄超過 37000 個詞。訓練模式時，選擇 adam(lr=0.002)做為 optimizer，batch_size 設為 512，validation set 為 training dataset 的 0.3，讓 model 自己去切。以這個架構在 Kaggle 上的分數為 **0.69982 (Private)**、**0.70117(Public)**，訓練過程如次頁圖，可以發現過程到第 2、3 個 epoch 就很難改善模式的 accuracy 了，模式的表現也沒有明顯的變化。





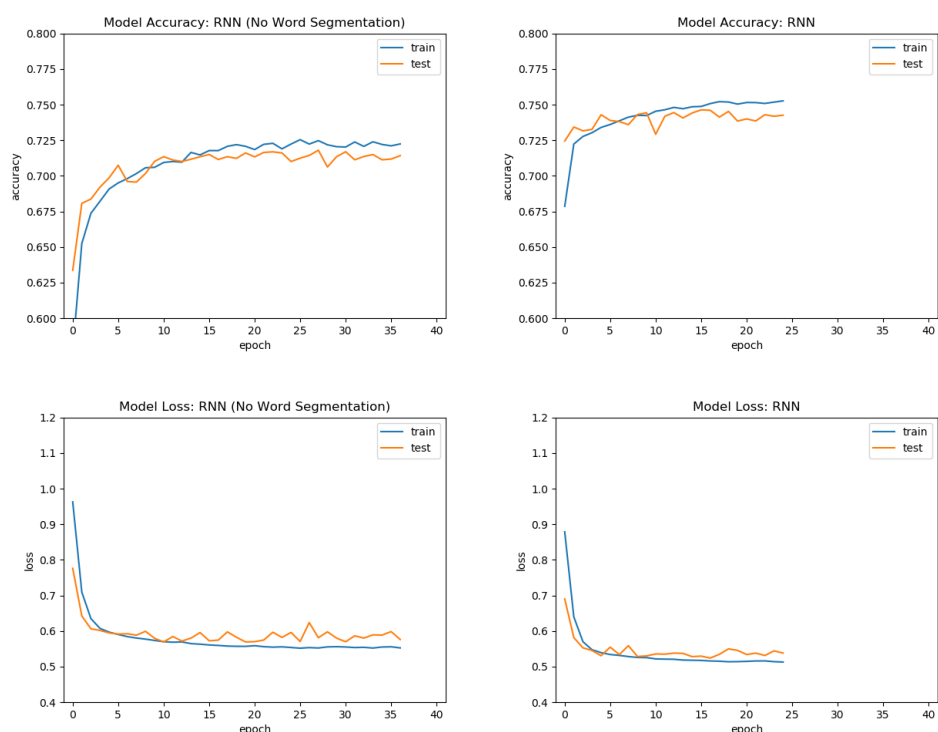
Problem 2.

進行本次作業時並沒有詳細記錄改善模式過程的結果，但是以 RNN 模式為例，在過程中發現以下可以改善模式表現的方式。

1. LSTM 與輸出層之間的 DNN 疊太多層會讓模式表現不好，原因可能是在訓練時，模式只會去調整比較接近輸出層的參數，讓 LSTM 的特性不明顯而使效果變差。
2. 兩層 LSTM 比一層 LSTM 的表現還好，原因可能是多層可以抓住更多跨字詞間的資訊。
3. 處理文字實有刪除一些比較沒有代表特定意義的停止詞、標點符號以及 Dcard 留言特有的樓層標籤，這樣比較能去除這些資訊的干擾。
4. 訓練 word2vec 的時候，發現 size 設太高(超過 300)也會降低模式表現，可能是因為這個模式 overfitting 了。

Problem 3.

本作業測試不做斷詞的 RNN 模式是直接使用第一題描述的 RNN 模式，但是將資料的句子斷成單一字元，將這個資料重新訓練一個 word2vec 模式套入第一題的架構之中，因此兩者的參數量是相同的，差別只在 embedding layer 的權重。訓練過程與第一題對比如下圖，左邊為不斷詞的結果，上傳到 **Kaggle** 的分數則為 **0.71020 (Private)**、**0.71037 (Public)**，可以發現還是比 BOW+DNN 好，但是因為不做斷詞沒有辦法反映中文多字為一個詞的特性，因此讓表現變差。



分別將兩句話輸入第一題描述的模式，結果如下：

可以發現 RNN 模式覺得兩句話都是惡意留言，可是看起來比較兇的第二句確實拿到比較高的預測分數，代表 embedding layer 與 LSTM 確實可以捕捉字詞順序的差異，雖然第一句好像還是不太算惡意留言。但是，由於 BOW+DNN 只記錄字詞出現頻率，因此對於 BOW+DNN 模式而言，這兩句話完全沒有差別，所以就得到一樣的預測分數，而且還都預測成非惡意留言。

按照上課投影片的計算方式進行計算，算式如下。

以下的表格就是計算過程，每次迭代所用的分類器都是手動挑錯誤率比較小的分類方式。

n	0	1	2	3	4	5	6	7	8	9
x^n	0	1	2	3	4	5	6	7	8	9
y^n	1	-1	1	1	1	-1	-1	1	-1	-1
$f_1(x^n) = \begin{cases} +1, x^n < 4.5 \\ -1, x^n \geq 4.5 \end{cases}$										
u_1^n	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
$f_1(x^n)$	1	1	1	1	1	-1	-1	-1	-1	-1
$\delta(f_1(x^n) \neq y^n)$	0.000	1.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000
$u_1^n * \delta(f_1(x^n) \neq y^n)$	0.000	1.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000
$\epsilon_1 = 0.200, d_1 = 2, \alpha_1 = 0.693$										
$f_2(x^n) = \begin{cases} +1, x^n > 1.5 \\ -1, x^n \leq 1.5 \end{cases}$										
u_2^n	0.500	2.000	0.500	0.500	0.500	0.500	0.500	2.000	0.500	0.500
$f_2(x^n)$	-1	-1	1	1	1	1	1	1	1	1
$\delta(f_2(x^n) \neq y^n)$	1.000	0.000	0.000	0.000	0.000	1.000	1.000	0.000	1.000	1.000
$u_2^n * \delta(f_2(x^n) \neq y^n)$	0.500	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.500	0.500
$\epsilon_2 = 0.313, d_2 = 1.483, \alpha_2 = 0.394$										
$f_3(x^n) = \begin{cases} +1, x^n < 0.5 \\ -1, x^n \geq 0.5 \end{cases}$										
u_3^n	0.742	1.348	0.337	0.337	0.337	0.742	0.742	1.348	0.742	0.742
$f_3(x^n)$	1	-1	-1	-1	-1	-1	-1	-1	-1	-1
$\delta(f_3(x^n) \neq y^n)$	0.000	0.000	1.000	1.000	1.000	0.000	0.000	1.000	0.000	0.000
$u_3^n * \delta(f_3(x^n) \neq y^n)$	0.000	0.000	0.337	0.337	0.337	0.000	0.000	1.348	0.000	0.000
$\epsilon_3 = 0.318, d_3 = 1.464, \alpha_3 = 0.381$										

三次迭代加總： $H(x^n) = \text{sign}(\sum_{t=1}^T \alpha_t f_t(x^n))$										
$\alpha_1 * f_1(x^n)$	0.693	0.693	0.693	0.693	0.693	-0.693	-0.693	-0.693	-0.693	-0.693
$\alpha_2 * f_2(x^n)$	-0.394	-0.394	0.394	0.394	0.394	0.394	0.394	0.394	0.394	0.394
$\alpha_3 * f_3(x^n)$	0.381	-0.381	-0.381	-0.381	-0.381	-0.381	-0.381	-0.381	-0.381	-0.381
$\sum_{t=1}^T \alpha_t f_t(x^n)$	0.680	-0.082	0.706	0.706	0.706	-0.680	-0.680	-0.680	-0.680	-0.680
$H(x^n)$	1	-1	1	1	1	-1	-1	-1	-1	-1
$\delta(H(x^n) \neq y^n)$	0	0	0	0	0	0	0	1	0	0
錯誤率：0.1										

Problem 6.

透過 LSTM 的方式計算，過程如下面的表格。

t	1	x^t					
c	0.00	[0 1 0 3]					
z	3	z_i	90	z_f	10	z_o	-10
$g(z)$	3.00	$f(z_i)$	1.00	$f(z_f)$	1.00	$f(z_o)$	0.00
c'	3.00	$h(c')$	3.00	y^t		0.00	

t	2	x^t					
c	3.00	[1 0 1 -2]					
z	-2	z_i	90	z_f	10	z_o	90
$g(z)$	-2.00	$f(z_i)$	1.00	$f(z_f)$	1.00	$f(z_o)$	1.00
c'	1.00	$h(c')$	1.00	y^t		1.00	

t	3	x^t					
c	1.00	[1 1 1 4]					
z	4	z_i	190	z_f	-90	z_o	90
$g(z)$	4.00	$f(z_i)$	1.00	$f(z_f)$	0.00	$f(z_o)$	1.00
c'	4.00	$h(c')$	4.00	y^t		4.00	

t	4	x^t					
c	4.00	[0 1 1 0]					
z	0	z_i	90	z_f	10	z_o	90
$g(z)$	0.00	$f(z_i)$	1.00	$f(z_f)$	1.00	$f(z_o)$	1.00
c'	4.00	$h(c')$	4.00	y^t		4.00	

t	5	x^t					
c	4.00	[0 1 0 2]					
z	2	z_i	90	z_f	10	z_o	-10
$g(z)$	2.00	$f(z_i)$	1.00	$f(z_f)$	1.00	$f(z_o)$	0.00
c'	6.00	$h(c')$	6.00	y^t		0.00	

t	6	x^t					
c	6.00	[0 0 1 -4]					
z	-4	z_i	-10	z_f	110	z_o	90
$g(z)$	-4.00	$f(z_i)$	0.00	$f(z_f)$	1.00	$f(z_o)$	1.00
c'	6.00	$h(c')$	6.00	y^t		6.00	

t	7	x^t					
c	6.00	[1 1 1 1]					
z	1	z_i	190	z_f	-90	z_o	90
$g(z)$	1.00	$f(z_i)$	1.00	$f(z_f)$	0.00	$f(z_o)$	1.00
c'	1.00	$h(c')$	1.00	y^t		1.00	

t	8	x^t					
c	1.00	[1 0 1 2]					
z	2	z_i	90	z_f	10	z_o	90
$g(z)$	2.00	$f(z_i)$	1.00	$f(z_f)$	1.00	$f(z_o)$	1.00
c'	3.00	$h(c')$	3.00	y^t		3.00	

所以算出來的 y 表列如下。

t	1	2	3	4	5	6	7	8
y^t	0	1	4	4	0	6	1	3