Homework 4 Report - Malicious Comments Identification

學號：R07922135　系級：資工碩一　姓名：顏百謙

Problem 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。

- 訓練曲線 (Training curve):顯示訓練過程的 loss 或 accuracy 變化。橫軸為 step 或 epoch,縱軸為 loss 或 accuracy。

Word Embedding方法

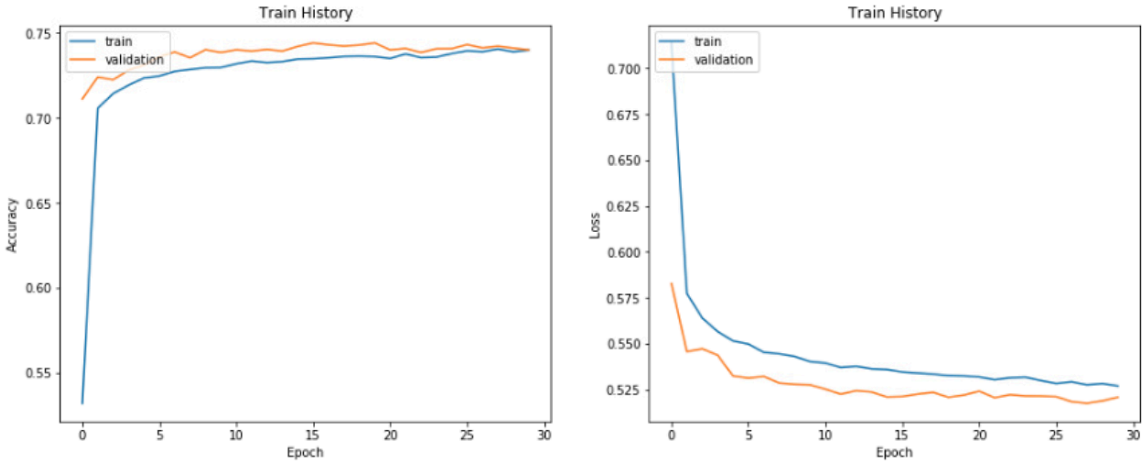
- Model架構
  - 有利用StopWord濾掉一些字詞
  - Embedding層將word index轉成vector再進入training

```
def build_LSTM_model():
    model = Sequential()
    model.add(Embedding(len(word2idx),vector_size,weights=[embeddings_matrix],input_length=word_len,trainable=False))
    model.add(BatchNormalization())
    model.add(LSTM(256, implementation=2, dropout=0.5, recurrent_dropout=0.5))
    model.add(Dense(64))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    model.add(Activation('relu'))
    model.add(Dense(2, activation='softmax'))
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
model = build_LSTM_model()
```

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 64, 256)	39848960
batch_normalization_1 (Batch	(None, 64, 256)	1024
lstm_1 (LSTM)	(None, 256)	525312
dense_1 (Dense)	(None, 64)	16448
batch_normalization_2 (Batch	(None, 64)	256
dropout_1 (Dropout)	(None, 64)	0
activation_1 (Activation)	(None, 64)	0
dense_2 (Dense)	(None, 2)	130

Total params: 40,392,130
Trainable params: 542,530
Non-trainable params: 39,849,600

- Training curve



- Kaggle Public result (0.74765)

- Result
  - Training和validation的Acc都卡在0.745左右上不去，可能要調整model結構或是v2w model或是stopword才可提升單一model的效果

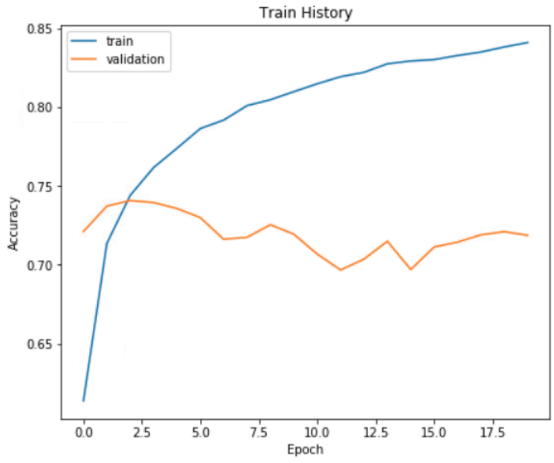
BOW+DNN Model

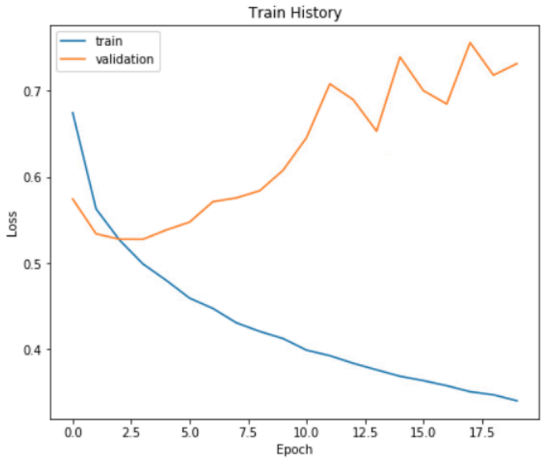
- Model架構
  - Bag結構取前20000個出現次數最多的字詞以及other和space共20002維的vector
  - 其中有用StopWord濾掉一些字詞

```python
def build_bow_model():
    model = Sequential()
    model.add(Dense(24, input_dim=bag_size+2))
    model.add(BatchNormalization())
    model.add(Dropout(0.6))
    model.add(Activation('relu'))
    model.add(Dense(30))
    model.add(BatchNormalization())
    model.add(Dropout(0.55))
    model.add(Activation('relu'))
    model.add(Dense(2, activation='softmax'))
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
model = build_bow_model()
```

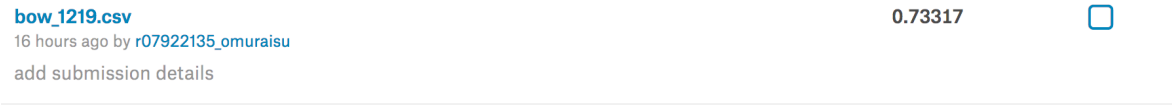
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 24)                480072
_________________________________________________________________
batch_normalization_1 (Batch (None, 24)                96
_________________________________________________________________
dropout_1 (Dropout)          (None, 24)                0
_________________________________________________________________
activation_1 (Activation)    (None, 24)                0
_________________________________________________________________
dense_2 (Dense)              (None, 30)                750
_________________________________________________________________
batch_normalization_2 (Batch (None, 30)                120
_________________________________________________________________
dropout_2 (Dropout)          (None, 30)                0
_________________________________________________________________
activation_2 (Activation)    (None, 30)                0
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 62
=================================================================
Total params: 481,100
Trainable params: 480,992
Non-trainable params: 108
_________________________________________________________________
```

- Training curve





- Kaggle Public result (0.73317)



- Result
  - 很明顯的Overfitting了，出來的結果也不盡理想

Problem 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。

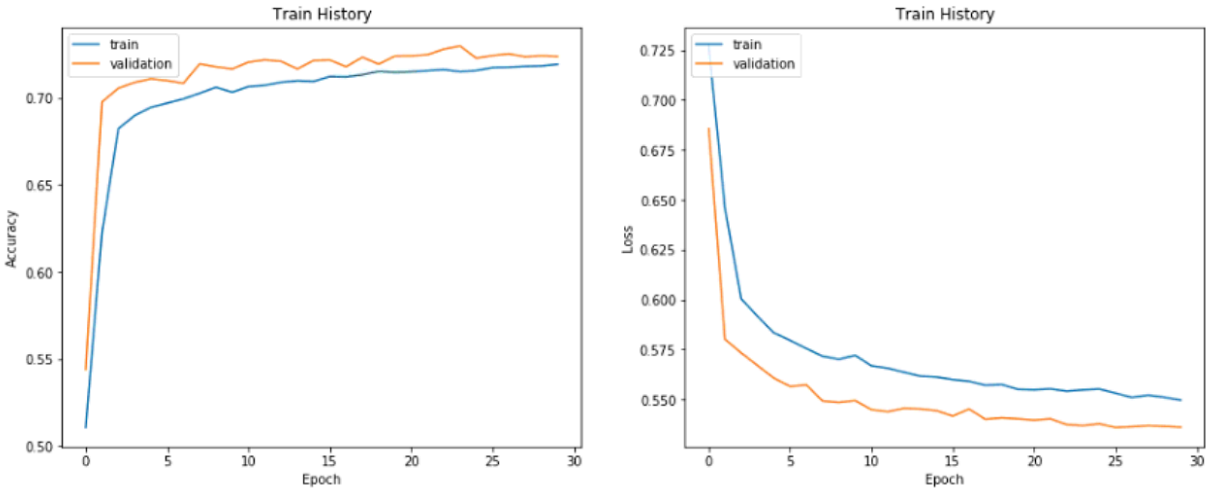
- 此次作業我選擇利用ensemble中的Bagging來提昇performance，選擇不同的「layer數」，「DNN node數」，「drop out比例」，「training data」來train出不同的model，再將其predict出的**機率相加取平均**以求獲得更好的結果
- 利用Bagging可以有效降低model的Variance以獲得較高的準確率
- Bagging結果：
  - Public : 0.75117

  - Private : 0.74950

Problem 3

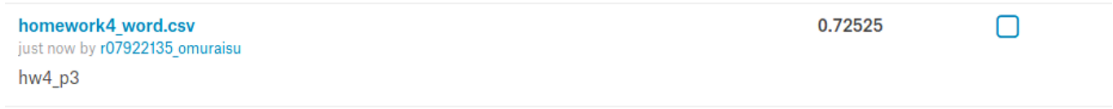
(1%) 請比較不做斷詞 (e.g., 以字為單位) 與有做斷詞,兩種方法實作出來的效果差異,並解釋為何有此差別。

以字為單位訓練出來的結果如下：



- validation的acc卡在0.725左右上不去，相較於斷詞的0.745~0.75效果差了些，原因有可能是中文的單字並不能完整表示其想表達的意義，故相較於斷詞的model少了一些information可以train。
- Kaggle result

- Public : 0.72525



- Private : 0.72485

Problem 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。

RNN Model Result:

- 判斷的結果（左:正面 右:負面）
- 可看見第二句的結果被判斷為較第一句負面（雖然結果還是判為正面...）

```
0
[[0.55854714 0.44145286]
 [0.5261509  0.4738491 ]]
```

BOW Model Result:

- 判斷的結果（左:正面 右:負面）
- 可看見第二句的結果和第一句結果一樣，且都離背叛為負面較遠
- 原因可能是BOW並沒辦法訓練出句子前後關係造成的語氣差異，故兩句的結果一模一樣

```
p
```

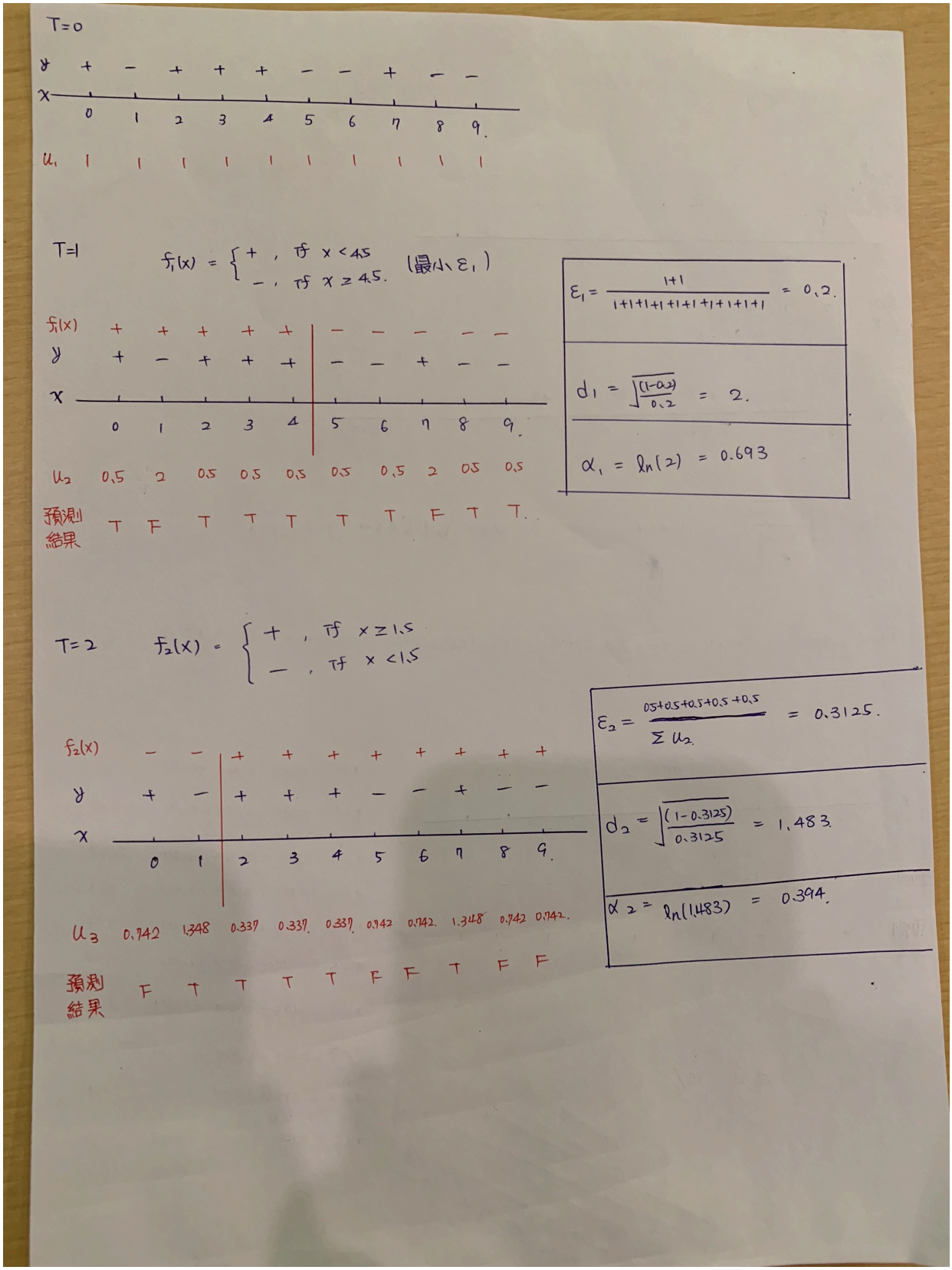
```
[array([[0.7179589, 0.2820411]], dtype=float32),
 array([[0.7179589, 0.2820411]], dtype=float32)]
```

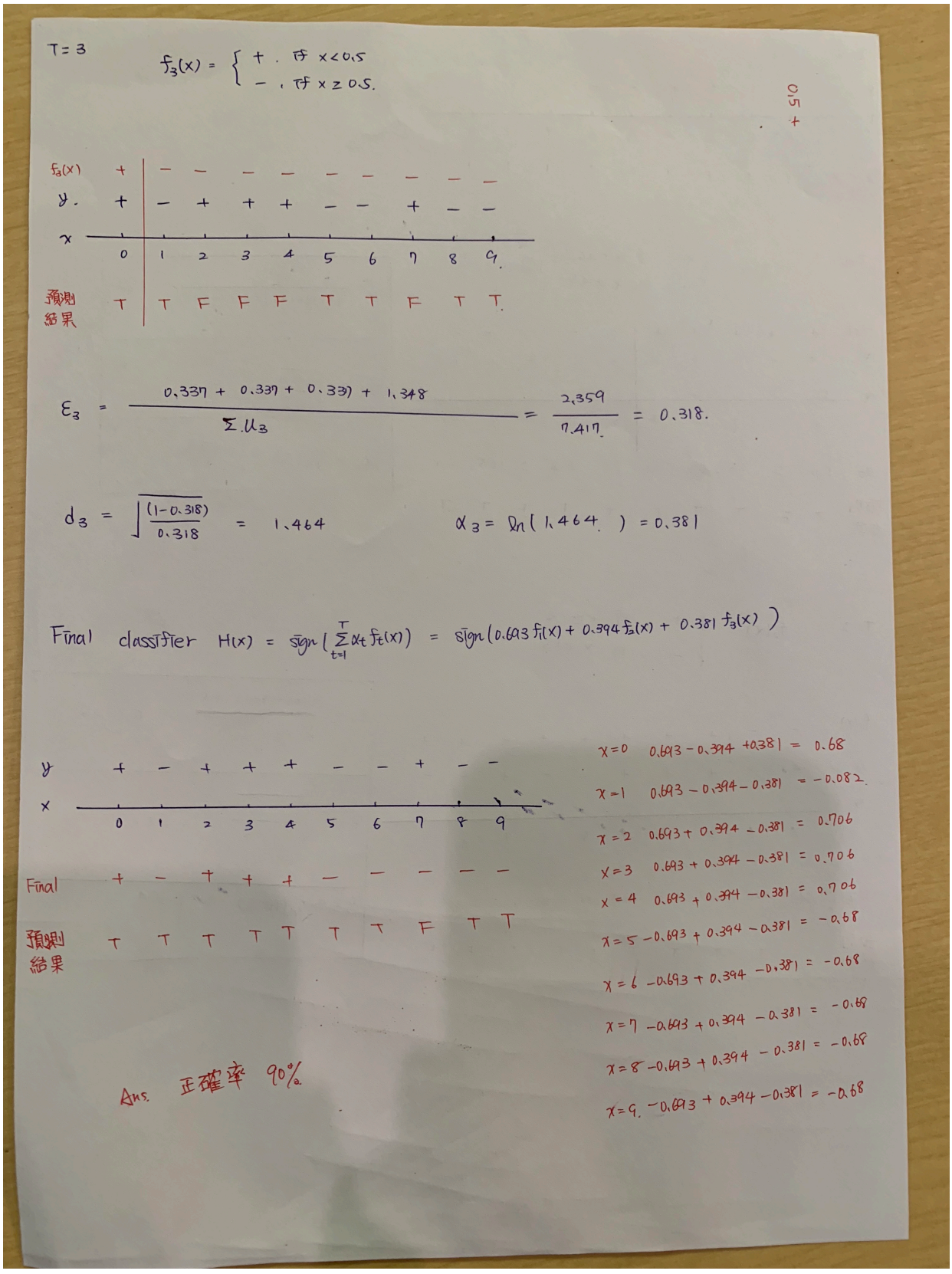
Problem 5

(1%) In this exercises, we will train a binary classifier with AdaBoost algorithm on the data shown in the table. Please use decision stump as the base classifier. Perform AdaBoost algorithm for T = 3 iterations. For each iteration (t = 1, 2, 3), write down the weights u_t^n used for training, the weighted error rate ϵ_t, scaling coefficient α_t, and the classification function $f_t(x)$. The initial weights u_1^n are set to 1 (n = 0, 1, ..., 9). Please refer to the course slides for the definitions of the above notations. Finally, combine the three classifiers and write down the final classifier.

x	0	1	2	3	4	5	6	7	8	9
y	+	-	+	+	+	-	-	+	-	-

Answer


T=0
y + − + + + − − + − −
x 0 1 2 3 4 5 6 7 8 9.
u_1 1 1 1 1 1 1 1 1 1 1
T=1
$f_1(x) = \begin{cases} +, & \text{if } x < 4.5 \\ -, & \text{if } x \geq 4.5. \end{cases}$ (最小 ε_1)
$f_1(x)$ + + + + + − − − − −
y + − + + + − − + − −
x 0 1 2 3 4 5 6 7 8 9.
u_2 0.5 2 0.5 0.5 0.5 0.5 0.5 2 0.5 0.5
預測結果 T F T T T T T F T T.
$\varepsilon_1 = \frac{1+1}{1+1+1+1+1+1+1+1+1+1} = 0.2.$
$d_1 = \sqrt{\frac{(1-0.2)}{0.2}} = 2.$
$\alpha_1 = \ln(2) = 0.693$
T=2
$f_2(x) = \begin{cases} +, & \text{if } x \geq 1.5 \\ -, & \text{if } x < 1.5 \end{cases}$
$f_2(x)$ − − + + + + + + + +
y + − + + + − − + − −
x 0 1 2 3 4 5 6 7 8 9.
u_3 0.742 1.348 0.337 0.337. 0.337. 0.742 0.742. 1.348 0.742 0.742.
預測結果 F T T T T F F T F F
$\varepsilon_2 = \frac{0.5+0.5+0.5+0.5+0.5}{\sum u_2} = 0.3125.$
$d_2 = \sqrt{\frac{(1-0.3125)}{0.3125}} = 1.483$
$\alpha_2 = \ln(1.483) = 0.394.$



Problem 6

(1%) In this exercise, we will simulate the forward pass of a simple LSTM cell. Figure.1 shows a single LSTM cell, where z is the cell input, z_i, z_f, z_o are the control inputs of the gates, c is the cell memory, and f, g, h are activation functions. Given an input x, the cell input and the control inputs can be calculated by

$$
\begin{aligned}
z &= w \cdot x + b \\
z_i &= w_i \cdot x + b_i \\
z_f &= w_f \cdot x + b_f \\
z_o &= w_o \cdot x + b_o
\end{aligned}
$$

Where w, w_i, w_f, w_o are weights and b, b_i, b_f, b_o are biases. The final output can be calculated by

$$
y = f(z_o)h(c')
$$

where the value stored in cell memory is updated by

$$
c' = f(z_i)g(z) + cf(z_f)
$$

Given an input sequence $x^t (t = 1, 2, \ldots, 8)$, please derive the output sequence y^t. The input sequence, the weights, and the activation functions are provided below. The initial value in cell memory is 0. Please note that your calculation process is required to receive full credit.

$w = [0,\ 0,\ 0,\ 1]$, $b = 0$
$w_i = [100,\ 100,\ 0,\ 0]$, $b_i = -10$
$w_f = [-100, -100,\ 0,\ 0]$, $b_f = 110$
$w_o = [0,\ 0,\ 100,\ 0]$, $b_o = -10$

t	1	2	3	4	5	6	7	8
x^t	0	1	1	0	0	0	1	1
	1	0	1	1	1	0	1	0
	0	1	1	1	0	1	1	1
	3	-2	4	0	2	-4	1	2

$$
f(z) = \frac{1}{1 + e^{-z}} \qquad g(z) = z \qquad h(z) = z
$$

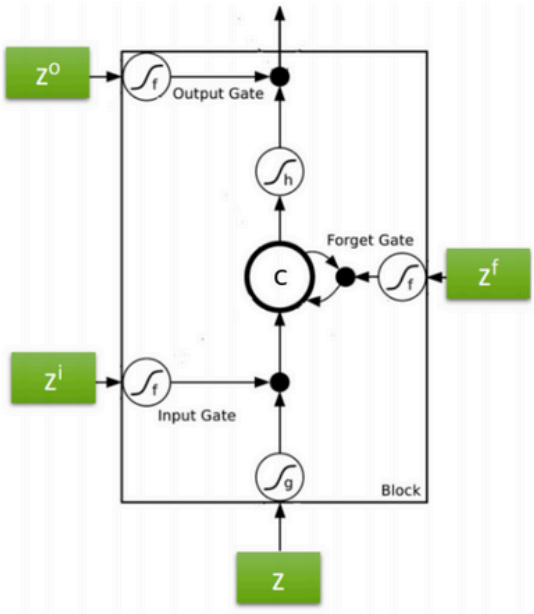


Figure 1: The LSTM cell

Answer:

$t = 0$ 時 $c = 0$

$t = 1$ 時

$$
z^1 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 3 \end{bmatrix} + 0 = 3
$$

$$
z_i^1 = \begin{bmatrix} 100 & 100 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 3 \end{bmatrix} - 10 = 90
$$

$$
\begin{bmatrix} 0 \end{bmatrix}
$$

$$z_f^1 = \begin{bmatrix} -100 & -100 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix} + 110 = 10$$

$$z_o^1 = \begin{bmatrix} 0 & 0 & 100 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 3 \end{bmatrix} - 10 = -10$$

$$c^{'} = \frac{1}{1+e^{-90}} * 3 + 0 * \frac{1}{1+e^{-10}} \approx 3$$

$$y^1 = \frac{1}{1+e^{10}} * 3 \approx 0.00013619360610730318$$

利用程式依照上面方式計算得到以下結果

$t = 2$ 時

$$z^2 = -2 \quad z_i^2 = 90 \quad z_f^2 = 10 \quad z_o^2 = 90$$

$$c^{'} = 0.9998638063938929$$

$$y^2 = 0.9998638063938929$$

$t = 3$ 時

$$z^3 = 4 \quad z_i^3 = 190 \quad z_f^3 = -90 \quad z_o^3 = 90$$

$$c^{'} = 4.0$$

$$y^3 = 4.0$$

$t = 4$ 時

$$z^4 = 0 \quad z_i^4 = 90 \quad z_f^4 = 10 \quad z_o^4 = 90$$

$$c^{'} = 3.9998184085251904$$

$$y^4 = 3.9998184085251904$$

$t = 5$ 時

$$z^5 = 2 \quad z_i^5 = 90 \quad z_f^5 = 10 \quad z_o^5 = -10$$

$$c^{'} = 5.999636825294247$$

$$y^5 = 0.00027237072485699856$$

$t = 6$ 時

$$z^6 = -4 \quad z_i^6 = -10 \quad z_f^6 = 110 \quad z_o^6 = 90$$

$$c^{'} = 5.999455233819437$$

$$y^6 = 5.999455233819437$$

$t = 7$ 時

$$z^7 = 1 \quad z_i^7 = 190 \quad z_f^7 = -90 \quad z_o^7 = 90$$

$$c^{'} = 1.0$$

$$y^7 = 1.0$$

$t = 8$ 時

$$z^8 = 2 \quad z_i^8 = 90 \quad z_f^8 = 10 \quad z_o^8 = 90$$

$$c^{'} = 2.9999546021312975$$

$$y^8 = 2.9999546021312975$$