Homework 4 Report Problem Set

R07522814 陳俊翰

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。

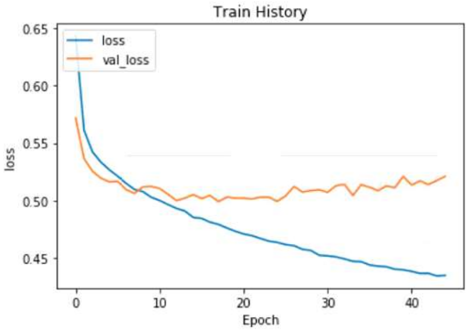
RNN+word embedding :
Word embedding 使用gensim Word2Vec套件，embedding size=256，利用一份training data sentences和兩分testing data sentences訓練。

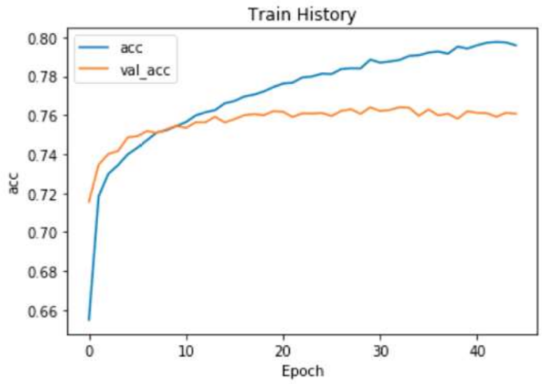
使用3層GRU配上適當深度的DNN

模型架構：

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, None, 256)	14258432
gru_1 (GRU)	(None, None, 128)	147840
gru_2 (GRU)	(None, None, 128)	98688
gru_3 (GRU)	(None, 128)	98688
batch_normalization_1 (Batch	(None, 128)	512
dense_1 (Dense)	(None, 256)	33024
dropout_1 (Dropout)	(None, 256)	0
batch_normalization_2 (Batch	(None, 256)	1024
dense_2 (Dense)	(None, 128)	32896
dropout_2 (Dropout)	(None, 128)	0
batch_normalization_3 (Batch	(None, 128)	512
dense_3 (Dense)	(None, 32)	4128
dropout_3 (Dropout)	(None, 32)	0
batch_normalization_4 (Batch	(None, 32)	128
dense_4 (Dense)	(None, 1)	33

訓練曲線：

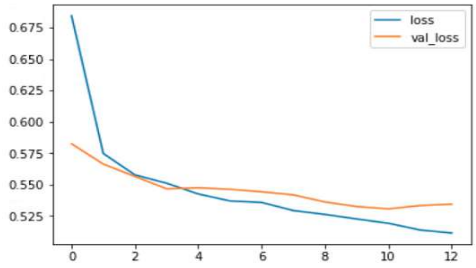


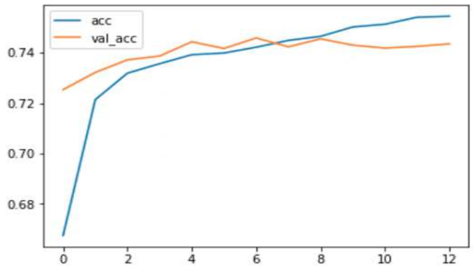


BOW+DNN：
Word embedding一樣使用word2vec訓練
BOW：計算個單字的出現次數，使用word embedding處理後加總，總和即代表該句留言，當作模型輸入。
最後搭配適當深度的DNN。
模型架構：

Layer (type)	Output Shape	Param #
dense_23 (Dense)	(None, 1024)	525312
dropout_21 (Dropout)	(None, 1024)	0
dense_24 (Dense)	(None, 512)	524800
dropout_22 (Dropout)	(None, 512)	0
dense_25 (Dense)	(None, 256)	131328
dropout_23 (Dropout)	(None, 256)	0
dense_26 (Dense)	(None, 128)	32896
dropout_24 (Dropout)	(None, 128)	0
dense_27 (Dense)	(None, 64)	8256
dropout_25 (Dropout)	(None, 64)	0
dense_28 (Dense)	(None, 1)	65

訓練曲線 :





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。

Preprocess : 將無用的符號和贅字清除留下能判斷正負面的詞彙，減少資料雜訊影響提高模型表現。

Embedding : 先經過jieba斷詞，在來使用word2vec訓練word embedding，訓練時為了增加特徵，我用了training data和兩倍的testing data的文字資料。

利用earlystopping避免overfitting，當validation accuracy不再上升時停止訓練。

如上課講義所述，word embedding比one-hot encoding能提高performance，因word embedding 將文字資料變得比較有意義且降維，使語意相近的字有關連性，且降維減少雜訊與垃圾資料，避免overfitting。

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。

有斷詞的BOW+DNN模型，kaggle正確率為0.74240，沒斷詞的BOW+DNN模型正確率下降到約62%，我認為是因為中文詞被拆成單字時意義會消失，導致模型錯誤，例如: “低能”模型可以判斷出是負面詞彙，然而”低”、”能”不容易被判斷為負面。

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。

Jieba斷詞的結果分別為： ['在', '說', '別人', '白痴', '之前', '，', '先', '想想', '自己']
以及['在', '說', '別人', '之前', '先', '想想', '自己', '，', '白痴']

RNN : 0.5235017, 0.57321037
BOW+DNN : 0.5447127, 0.5447127

RNN因為有考慮Time series關係，輸入順序會影響預測結果而BOW不會，RNN模型後面那句分數較高代表RNN認為後面那句比較負面，最後因為”白癡”是很明顯的負面言論所以預測分數都大於0.5，表示這兩個模型都成功預測這兩句話是負面言論。

Problem 5. (1%) In this exercise, 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	+	-	+	+	+	-	-	+	-	-

5.

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

$t=1$ $u_1^n = [0.1] \times 10$

$f_1(x) = \begin{cases} 1 & x < 4.5 \\ -1 & x > 4.5 \end{cases}$ $\varepsilon_1 = 0.2$ $\alpha_1 = \ln\sqrt{\frac{1-0.2}{0.2}} = 0.6931$

$d_1 = \sqrt{\frac{1-0.2}{0.2}} = 2$

$\frac{0.1}{2} = 0.05$ $0.1 \times 2 = 0.2$

$t=2$ $u_2^n = [\,0.05 \;\; 0.2 \;\; 0.05 \;\; 0.05 \;\; 0.05 \;\; 0.05 \;\; 0.2 \;\; 0.05 \;\; 0.05]$

$f_2(x) = \begin{cases} 1 & x > 1.5 \\ -1 & x < 1.5 \end{cases}$ $\varepsilon_2 = \frac{0.05 \times 4 + 0.05 \times 1}{0.05 \times 8 + 0.2 \times 2} = 0.3125$

$\alpha_2 = \ln\sqrt{\frac{1-\varepsilon_2}{\varepsilon_2}}$, $d_2 = \sqrt{\frac{1-\varepsilon_2}{\varepsilon_2}} = 1.4832$

$= 0.3942$

$\frac{0.05}{1.4832} = 0.0337$, $0.05 \times 1.4832 = 0.07416$, $\frac{0.2}{1.4832} = 0.1348$

$0.2 \times 1.4832 = 0.2966$

$t=3$ $u_3^n = [0.07416, 0.1348, 0.0337, 0.0337, 0.0337, 0.07416, 0.07416, 0.1348, 0.07416, 0.07416]$

$f_3(x) = \begin{cases} 1 & x < 0.5 \\ -1 & x > 0.5 \end{cases}$, $\varepsilon_3 = \frac{3 \times 0.0337 + 1 \times 0.1348}{0.07416 \times 5 + 0.0337 \times 3 + 0.1348 \times 2} = 0.3181$

$d_3 = \sqrt{\frac{1-\varepsilon_3}{\varepsilon_3}} = 1.4641$ $\alpha_3 = 0.3812$

$H(x) = \text{sign}(\alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x))$

A: [1 -1 1 1 1 -1 -1 -1 -1 -1]

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

$$z = w \cdot x + b$$

$$z_i = w_i \cdot x + b_i$$

$$z_f = w_f \cdot x + b_f$$

$$z_o = w_o \cdot x + b_o$$

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

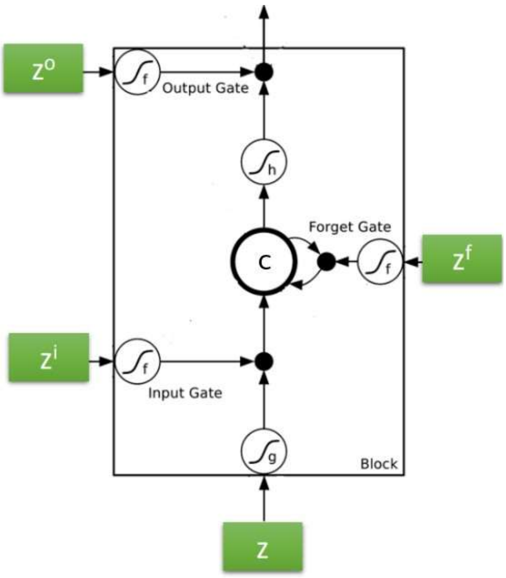
$$c' = f(z_i)g(z) + cf(z_f)$$

Given an input sequence x^t (t = 1, 2, ..., 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.

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

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

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



Answer 6 :

$$y_t = [0.001, 0.9999, 4.0000, 3.9998, 0.0003, 5.9998, 1.0000, 3.0000]$$

計算過程(by matlab) :

```
clc
clear
xx=[0 1 0 3;1 0 1 -2;1 1 1 4;0 1 1 0;0 1 0 2;0 0 1 4;1 1 1 1;1 0 1 2];
b=0; w=[0 0 0 1];
bi=-10; wi=[100 100 0 0];
bf=110; wf=[-100 -100 0 0];
bo=-10; wo=[0 0 100 0];
c=0;
y=[];
for i = 1:8
   x=xx(i,:);
   z=w*x'+b;
   zi=wi*x'+bi;
   zf=wf*x'+bf;
   zo=wo*x'+bo;
   cc=f(zi)*g(z)+c*f(zf);%cc = c'
   c=cc; %update cell
   ytmp=f(zo)*h(cc);
   y = [y ytmp];
end
fprintf('Answer:');
disp(y);
function fo = f(z)
fo=1/(1+exp(-z));
end
function go = g(z)
go=z;
end
function ho = h(z)
ho=z;
end
```