Homework 4 Report Problem Set

Professor Pei-Yuan Wu
EE5184 - Machine Learning

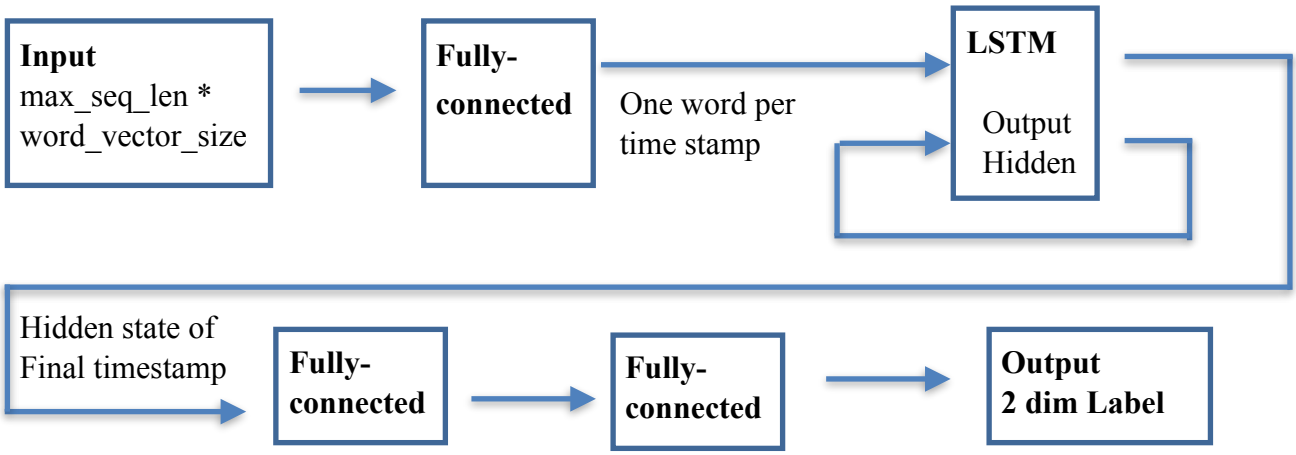
學號：B05902022
姓名：張雅信

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 + RNN]

word embedding: Gensim W2Vmodel trained by train+test text.

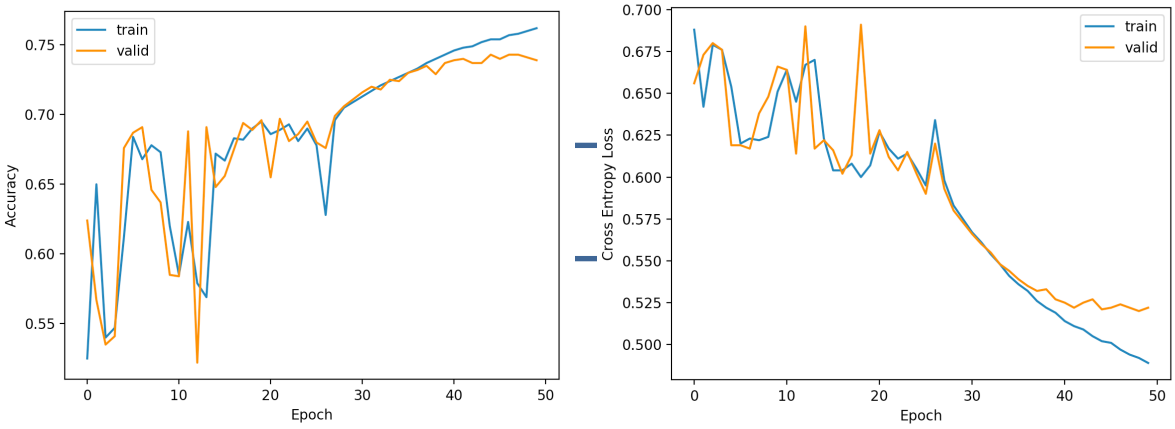
RNN model:



parameters:

max_seq_len = 128, LSTM input & output dim = 96,
word_vector_size = 128, batch_size = 512

training curve:



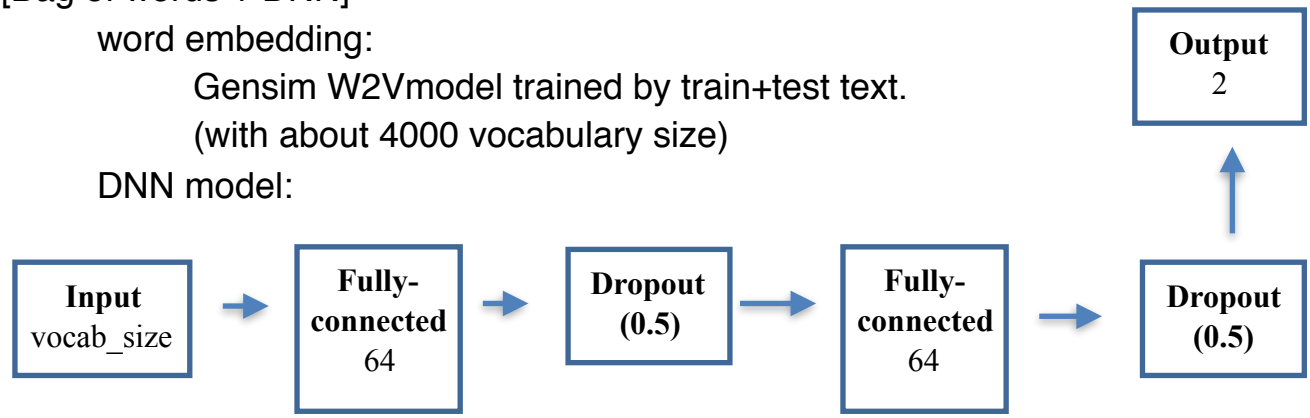
[Bag of words + DNN]

word embedding:

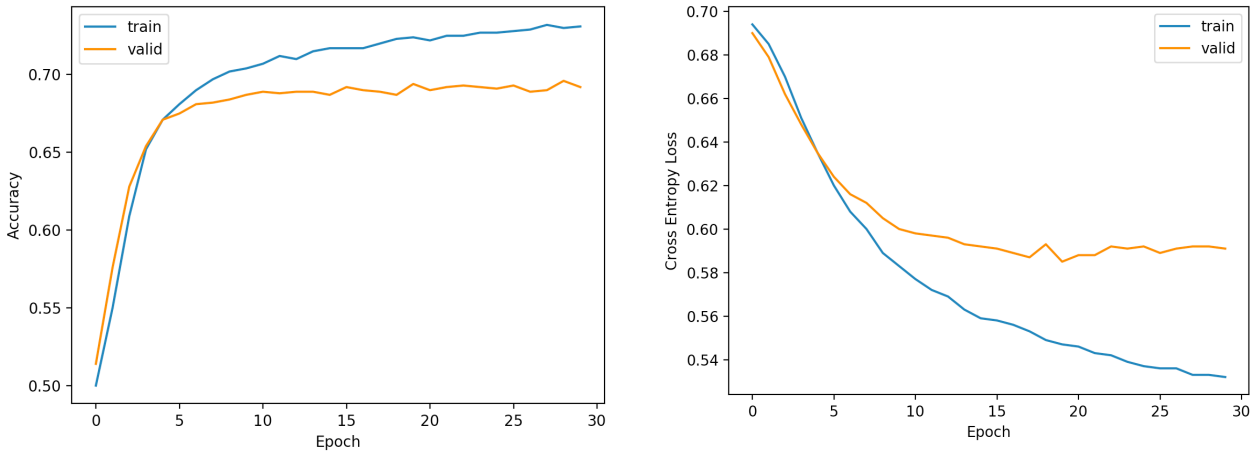
Gensim W2Vmodel trained by train+test text.

(with about 4000 vocabulary size)

DNN model:



training curve:



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。

[Word embedding + RNN]

使用最基本的 model 就可以超過 simple baseline(72%)，
在 LSTM 後加 Fully connected layer 以及 調整 hidden state 的維度，
可以有小幅進步，
(讓 model “消化” hidden 的意義，但減小 hidden dimension 避免 overfit)
在 ensemble 後即超過 strong baseline (74.6%)。

＊ Preprocessing 時把句子侷限在 max_seq_len 內，有丟掉一些資料，
若做更精細的處理，有機會使正確率再往上提升。

[Bag of words + DNN]

同樣使用最基本的 model 就可以超過 simple baseline(72%)。
但使用 BOW 主要的問題是資料量太大，
透過訓練 gensim model 時增加最小出現次數的限制，
減少 vocabulary size，直到資料轉換成 BOW 時機器裝得下。

BOW + DNN 很容易 overfit，
在每層 fully connected layer 後加了 dropout 即過 baseline。

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。

[Word Embedding + RNN]

	Training Loss	Training Acc	Valid Loss	Valid Acc
斷詞	0.514	0.752	0.597	0.704
不斷詞	0.473	0.777	0.533	0.741

兩者的 validation accuracy 都超過 0.7，算是有成功訓練起來的。
但在使用相同 model 參數量的狀況下，斷詞的 vocabulary 大大的增加：
[vocabulary size: 斷詞 >100000, 不斷詞 <8000]
所以有斷詞的 loss 與 accuracy 都略遜於不斷詞，
而將參數量調整至相近後兩者表現也非常接近。

此外，兩者都是使用 pytorch 的 nn.Embedding 訓練 word vector，
若改為使用 gensim word2vec，表現會再稍有進步。
(最後繳交 Kaggle submission 的也是使用 gensim)

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。

BOW
output: [[-0.3297, 0.0806], [-0.3297, 0.0806]]
predict: [1, 1]
RNN
output: [[0.4082, -0.5352], [-1.4979, 1.1341]]
predict: [0, 1]

由於 BOW 只能知道一句話裡有哪些詞，拋棄了一個句子 sequence 的特性，
因此從 BOW 來看，兩句是一模一樣的。

相對的，RNN 則保留了 sequence 的特性，因此可以對兩句話做出不同的評斷。
一般來說，這樣比較能真正了解一句話的意思，
以這兩句為例，正確答案應該是 RNN 產生的 [0, 1] 。

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	+	-	+	+	+	-	-	+	-	-

t = 1:

$$u_1^n = 1 for \quad all \quad data,$$

$$f_1(x) = sign(4.5 - x)$$

$$\epsilon_1 = \frac{\sum_n u^n \delta(f(x^n) \neq \hat{y})}{\sum_n u_1^n} = \frac{2}{10}$$

$$\alpha_1 = ln\sqrt{(1-0.2)/0.2} = 0.693$$

t = 2:

$$u_2^n = 2 for\ n = 1,7 \quad else \quad u_2^n = 0.5$$

$$f_2(x) = sign(x - 1.5)$$

$$\epsilon_2 = \frac{2.5}{10}$$

$$\alpha_2 = ln\sqrt{(1-0.25)/0.25} = 0.549$$

t = 3:

$$u_3^n = \{\frac{\sqrt{3}}{2}, \frac{2}{\sqrt{3}}, \frac{0.5}{\sqrt{3}}, \frac{0.5}{\sqrt{3}}, \frac{0.5}{\sqrt{3}},$$
$$\frac{\sqrt{3}}{2}, \frac{\sqrt{3}}{2}, \frac{2}{\sqrt{3}}, \frac{\sqrt{3}}{2}, \frac{\sqrt{3}}{2}\}$$

$$f_3(x) = sign(0.5 - x)$$

$$\epsilon_3 = \frac{2.02}{10}$$

$$\alpha_3 = ln\sqrt{(1-0.202)/0.202} = 0.687$$

Final Classifier: $H(x) = sign(\sum_{t=1}^{3} \alpha_t f_t(x))$

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

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_p 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

$$z = w \cdot x + b, \qquad z_i = w_i \cdot x + b_i, \qquad z_f = w_f \cdot x + b_f, \qquad z_o = w_o \cdot x + b_o$$

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 $y = f(z_0)h(c)$

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.

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

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

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

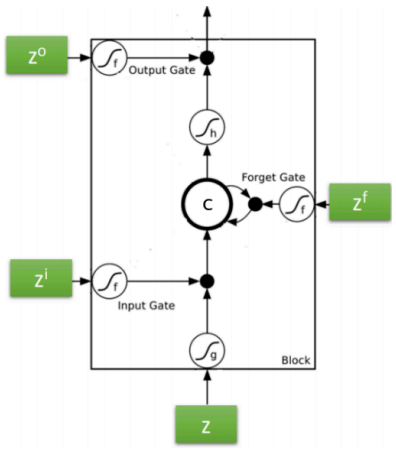


Figure 1: The LSTM cell

when $t = 1, \quad z = 3, \quad z_i = 90, \quad z_f = 10, \quad z_o = -10, \quad y = 0, \quad c' = 3$

when $t = 2, \quad z = -2, \quad z_i = 90, \quad z_f = 10, \quad z_o = 90, \quad y = 3, \quad c' = 0.9998$

when $t = 3, \quad z = 4, \quad z_i = 190, \quad z_f = -90, \quad z_o = 90, \quad y = 0.9998, \quad c' = 4$

when $t = 4, \quad z = 0, \quad z_i = 90, \quad z_f = 10, \quad z_o = 90, \quad y = 4, \quad c' = 3.9998$

when $t = 5, \quad z = 2, \quad z_i = 90, \quad z_f = 10, \quad z_o = -10, \quad y = 0.00018, \quad c' = 5.9996$

when $t = 6, \quad z = -4, \quad z_i = -10, \quad z_f = 110, \quad z_o = 90, \quad y = 5.9996, \quad c' = 5.9994$

when $t = 7, \quad z = 1, \quad z_i = 190, \quad z_f = -90, \quad z_o = 90, \quad y = 5.9994, \quad c' = 1$

when $t = 8, \quad z = 2, \quad z_i = 90, \quad z_f = 10, \quad z_o = 90, \quad y = 1, \quad c' = 2.9999$

輸出： $y^t = [\ 0, \ 3, \ 0.9998, \ 4, \ 0.00018, \ 5.9996, \ 5.9994, \ 1\ ]$