

# HW3 Handwritten Assignment

Lecturer: Pei-Yuan Wu

TAs: Yuan-Chia Chang(Problem 1, 2), Chun-Lin Huang(Problem 3, 4, 5)

October 2023, First Edition

## Problem 1 (LSTM Cell)(0.5%)

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 the following equations :

- $z = w \cdot x + b$
- $z^i = w_i \cdot x + b_i$
- $z^f = w_f \cdot x + b_f$
- $z^o = w_o \cdot x + b_o$

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

$$c' = f(z^i)g(z) + cf(z^f)$$

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

Given an input sequence  $x^t$  ( $t = 1, 2, 3, 4$ ), please derive the output sequence  $y_t$ . The input sequence, the weights, and the activation functions are provided below.

$$\begin{array}{ll} w = [0, 0, 1, 0] & , b = 0 \\ w_i = [50, 50, 0, 0] & , b_i = -5 \\ w_f = [-50, -50, 0, 0], & , b_f = 120 \\ w_o = [0, 0, 200, 0] & , b_o = -30 \\ x^1 = [0, 0, 1, 3] & , x^2 = [0, 1, -1, 2] \\ x^3 = [2, 1, 3, 4] & , x^4 = [0, 1, 0, 0] \end{array}$$

The initial value in cell memory is 0. **Please note that your calculation process is required to receive full credit.**

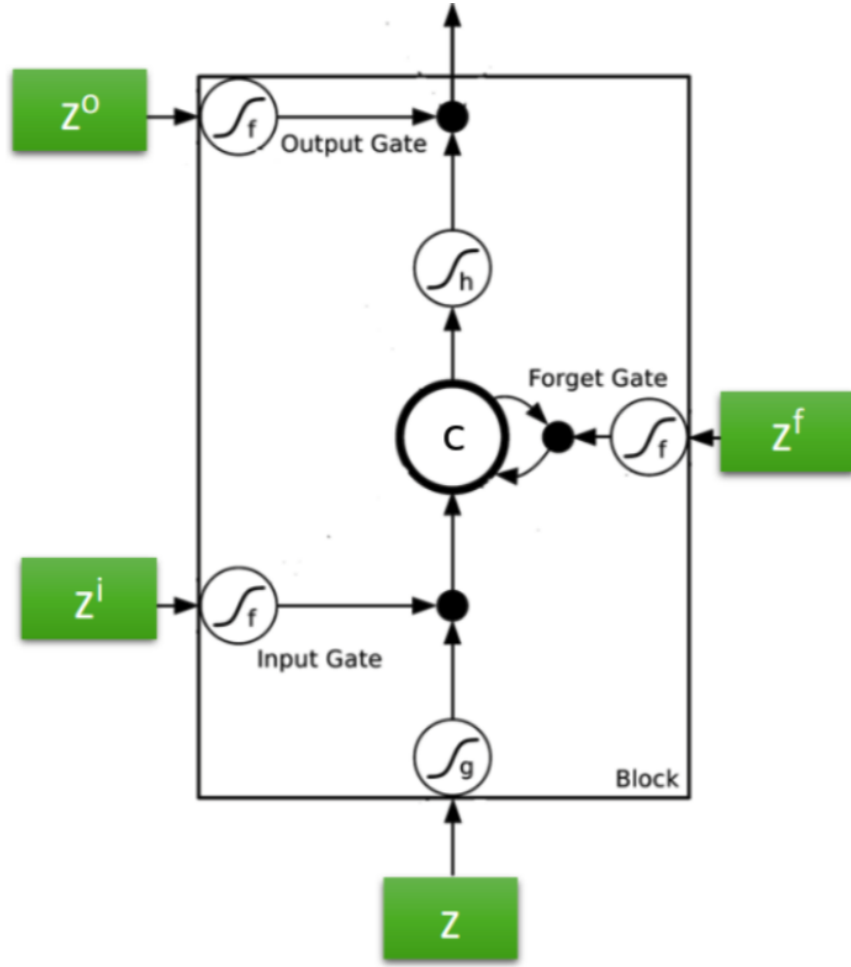


Figure 1: Problem 1 LSTM model

## Problem 2 (Laplacian Eigenmaps)(1.5%)

Consider an undirected connected graph  $G$ , which is shown below. We want to utilize Laplacian Eigenmaps method to reduce these 10 points to 3-dimensional space. Here, undirected graph means that edges in the graph do not have a direction, and connected graph means that there is a path from any node to any other node in the graph.

1. Write down the adjacency matrix  $W$
2. Write down the diagonal matrix  $D = \text{diag}(d_1, \dots, d_{10})$ , where  $d_i = \sum_{j=1}^{10} \frac{W_{ij} + W_{ji}}{2}$  and the Laplacian  $L = D - W$ .
3. By HW2 Problem 3, Neighbor Embedding Slide p.7-p.10 and programming tools(MATLAB, Python...), solve the optimization problem  
minimize  $\text{Trace}(\Psi^T L \Psi)$   
subject to  $\Psi^T D \Psi = I_3$   
variables  $\Psi \in \mathbb{R}^{10 \times 3}$   
Also, please plot the reduced points  $z_1, \dots, z_{10}$  in 3-D scatter plot.

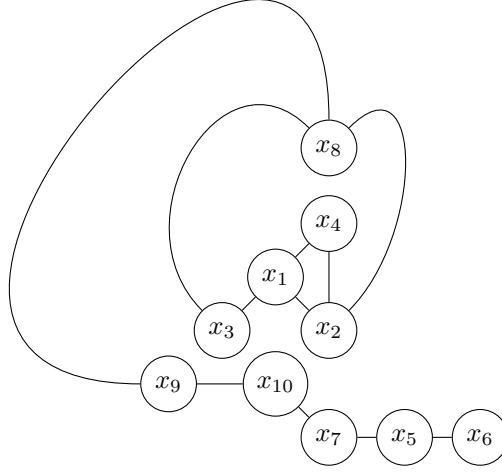


Figure 2: Problem 2 undirected connected graph  $G$

4. You may find that the minimal eigenvalue of  $\mathbf{L}$  is 0, and the corresponding eigenvector is

$$\begin{bmatrix} c \\ c \\ \vdots \\ c \end{bmatrix} \quad (1)$$

where  $c$  is a constant. Since all the points fall into a plane, the span of these points is  $\mathbb{R}^2$ . In order to construct  $\mathbf{z}_1, \dots, \mathbf{z}_{10}$  such that  $\text{span}\{\mathbf{z}_1, \dots, \mathbf{z}_{10}\} = \mathbb{R}^3$ , we need choose the second, third, fourth smallest eigenvalue and the corresponding eigenvectors. Please plot the reduced points  $\mathbf{z}_1, \dots, \mathbf{z}_{10}$  in 3-D scatter plot and verify that whether  $\text{Trace}(\mathbf{\Psi}^T \mathbf{L} \mathbf{\Psi}) = 0.726$  and  $\mathbf{\Psi}^T \mathbf{D} \mathbf{\Psi} = \mathbf{I}_3$ .

5. Show that for no matter the graph is, there is an eigenvector

$$\begin{bmatrix} c \\ c \\ \vdots \\ c \end{bmatrix} \quad (2)$$

where  $c$  is a constant, and the corresponding eigenvalue is 0.

6. By Neighbor Embedding Slide p.9, please show that

$$\forall f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} \in \mathbb{R}^N, f^T \mathbf{L} f = \sum_{1 \leq i, j \leq N} (f_i - f_j)^2.$$

7. Show that if  $f$  is an eigenvector which corresponds to eigenvalue 0, then  $f^T \mathbf{L} f = 0$ .
8. Show that if the graph is connected, the second smallest eigenvalue of  $\mathbf{L}$  will be nonzero.

### Problem 3 (Multiclass AdaBoost)(1.5%)

Let  $\mathcal{X}$  be the input space,  $\mathcal{F}$  be a collection of multiclass classifiers that map from  $\mathcal{X}$  to  $[1, K]$ , where  $K$  denotes the number of classes. Let  $\{(x_i, \hat{y}_i)\}_{i=1}^m$  be the training data set, where  $x_i \in \mathcal{X}$  and  $\hat{y}_i \in [1, K]$ . Given  $T \in \mathbb{N}$ , suppose we want to find functions

$$g_{T+1}^k(x) = \sum_{t=1}^T \alpha_t f_t^k(x), \quad k \in [1, K]$$

where  $f_t \in \mathcal{F}$  and  $\alpha_t \in \mathbb{R}$  for all  $t \in [1, T]$ . Here for  $f \in \mathcal{F}$ , we denote  $f^k(x) = \mathbf{1}\{f(x) = k\}$ , where  $\mathbf{1}(\cdot)$  is an indicator function, as the  $k$ 'th element in the one-hot representation of  $f(x) \in [1, K]$ . The aggregated classifier  $h : \mathcal{X} \rightarrow [1, K]$  is defined as

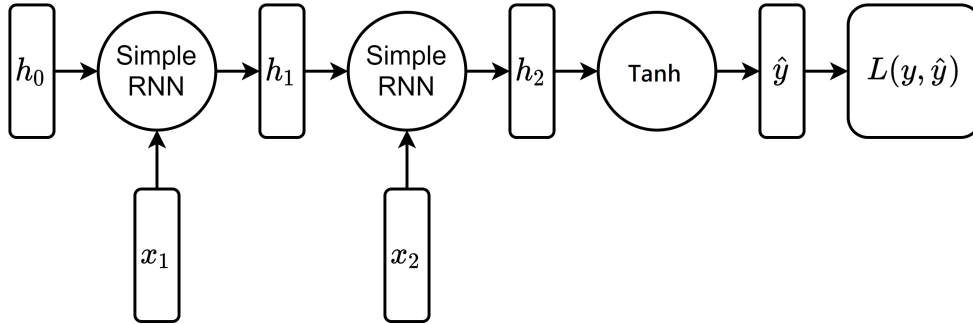
$$x \mapsto \operatorname{argmax}_{1 \leq k \leq K} g_{T+1}^k(x)$$

Please apply gradient boosting to show how the functions  $f_t$  and coefficients  $\alpha_t$  are computed with an aim to minimize the following loss function

$$L((g_{T+1}^1, \dots, g_{T+1}^K)) = \sum_{i=1}^m \exp \left( \frac{1}{K-1} \sum_{k \neq \hat{y}_i} g_{T+1}^k(x_i) - g_{T+1}^{\hat{y}_i}(x_i) \right)$$

### Problem 4 (Backpropagation through time via Simple RNN)(1%)

Backpropagation through time is a critical concept to know as we train a recurrent network. Here, we set a toy case of prediction problem. The Simple RNN module has two kinds of weights,  $w_i$  and  $w_h$ , such that



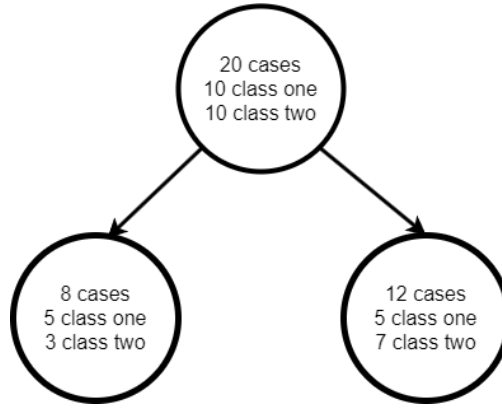
$h_t = \tanh(w_i x_t + w_h h_{t-1})$ , where  $t$  represents the index of steps. The module has the weight  $w_o$  such that  $\hat{y} = \sigma(w_o h_2)$ , where  $\sigma(w_o h_2) = \frac{1}{1 + \exp(-w_o h_2)}$ . The initial state  $h_0$  is set to be 0. The sequential input only contains  $\{x_1, x_2\}$ ; the label is  $y$ ; the loss function is MSE. Please derive  $\frac{\partial L(y, \hat{y})}{\partial w_o}$ ,  $\frac{\partial L(y, \hat{y})}{\partial w_h}$ ,  $\frac{\partial L(y, \hat{y})}{\partial w_i}$  in terms of  $x_1, x_2, h_0, h_1, h_2, w_i, w_o$ , and  $w_h$ .

### Problem 5 (Loss function of Decision tree) (1.5%)

It is known that decision tree is still a powerful classification model now a days. There are two different loss functions when it comes to entropy counting, which are Shannon information gain and Gini index. Following are their definition:

$$\text{Gini index} = \frac{N_{left}}{N} \left( 1 - \sum_{i=1}^c (p_{left}^i)^2 \right) + \frac{N_{right}}{N} \left( 1 - \sum_{i=1}^c (p_{right}^i)^2 \right)$$

$$\text{Shannon information gain} = \frac{N_{left}}{N} \left( - \sum_{i=1}^c p_{left}^i \log_2 p_{left}^i \right) + \frac{N_{right}}{N} \left( - \sum_{i=1}^c p_{right}^i \log_2 p_{right}^i \right)$$



$p_i^j$  := the proportion of class  $j$  in the node  $i$ .

$N_i$  := the number of cases in the node  $i$ .

Now we give a toy example. In this case  $N_{left} = 8, p_{left}^1 = \frac{5}{8}, p_{left}^2 = \frac{3}{8}, N_{right} = 12, p_{right}^1 = \frac{5}{12}, p_{right}^2 = \frac{7}{12}$

In the following questions we consider classification of two cases. Please calculate the entropy of the following questions using two above-mentioned loss functions.

- (a) (i) A 50/50 split with the first part containing 80% of positive examples and the second part containing 75% of positive examples.
- (ii) A 80/20 split with the first part containing 0% of positive examples and the second part containing 90% of positive examples.
- (iii) A 90/10 split with the first part containing 1% of positive examples and the second part containing 100% of positive examples.
- (b) However, now suppose that our case is to detect the covid-19. Thus, we want our entropy function can have a higher loss on (iii) than (ii). Please decide a function that fulfill this criteria and write down the loss.

## Version Description

1. First Edition: Finish Problem 1 to 5