HW3 Handwritten Assignment Solution

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

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

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.

$$\begin{array}{ll} w = [0, 0, 1, 0] & , b = 0 \\ w_i = [50, 50, 0, 0] & , b_i = -5 \\ w_f = [-50, -50, 0, 0], & , b_f = 120 \\ w_o = [0, 0, 200, 0] & , b_0 = -30 \\ x^1 = [0, 0, 1, 3] & , x^2 = [0, 1, -1, 2] \\ x^3 = [2, 1, 3, 4] & , x^4 = [0, 1, 0, 0] \end{array}$$

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

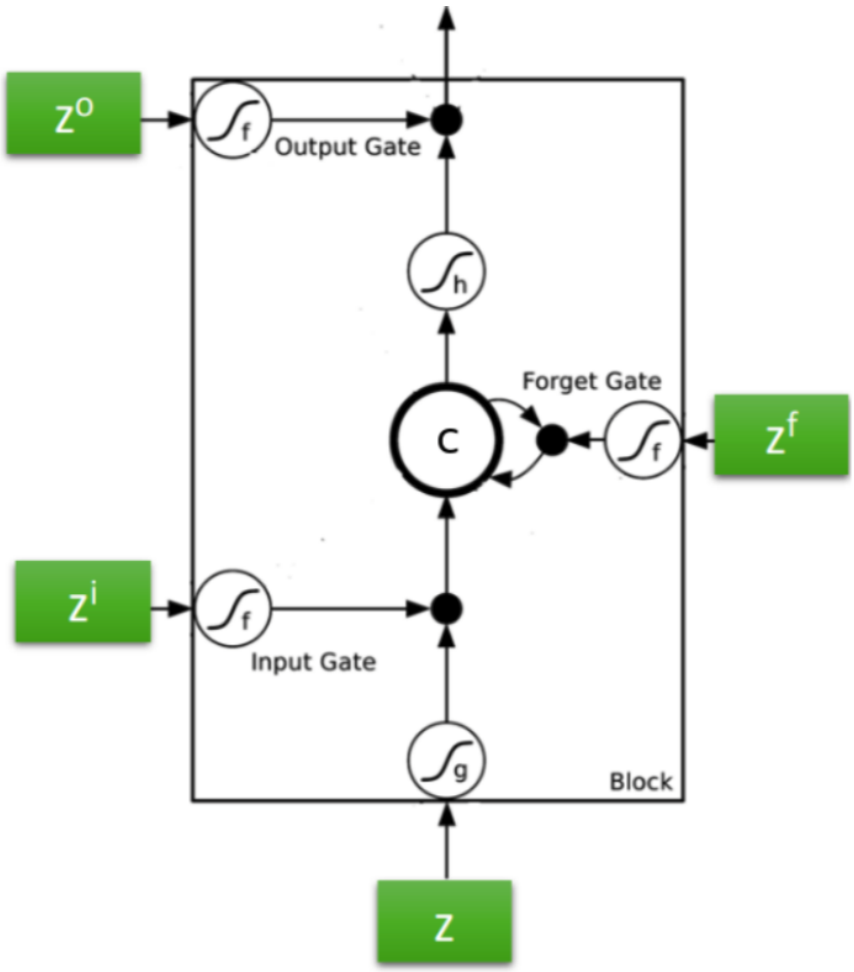


Figure 1: Problem 1 LSTM model

Solution: when $t = 1,\ z_1 = 1, z_1^i = -5, z_1^f = 120, z_1^o = 170, c_1 = 0.0067, y_1 = 0.0067.$
$t = 2,\ z_2 = -1, z_2^i = 45, z_2^f = 70, z_2^o = -230, c_2 = -1, y_2 = 0.$
$t = 3,\ z_3 = 3, z_3^i = 145, z_3^f = -30, z_3^o = 570, c_3 = 3, y_3 = 3.$
$t = 4,\ z_4 = 0, z_4^i = 45, z_4^f = 70, z_4^o = -30, c_4 = 0, y_4 = 0.$
Hence, $y_1 = 0.0067, y_2 = 0, y_3 = 0, y_4 = 0.$

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.

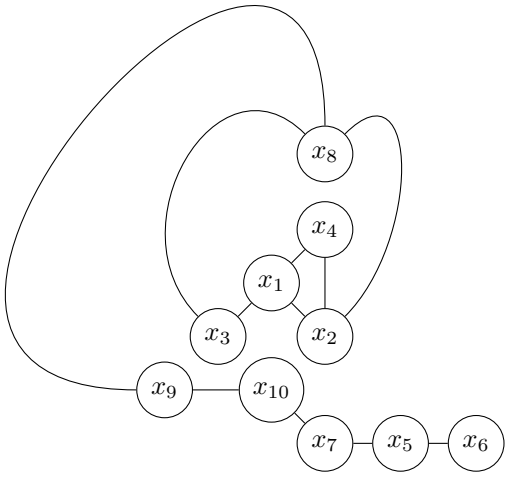


Figure 2: Problem 2 undirected connected graph G

1. Write down the adjacency matrix $\boldsymbol{W}$

Solution: $\boldsymbol{W}$ is shown below.

$$\begin{bmatrix}
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0
\end{bmatrix}$$

Figure 3: $\boldsymbol{W}$

2. Write down the diagonal matrix $\boldsymbol{D} = diag(d_1, ..., d_{10})$, where $d_i = \sum_{j=1}^{10} \frac{\boldsymbol{W_{ij}} + \boldsymbol{W_{ji}}}{2}$ and the Laplacian $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$.

Solution: $\boldsymbol{D}$ and $\boldsymbol{L}$ is shown below.

```
[[3. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 3. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 2. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 2. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 2. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 2. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 3. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 2. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 2.]]
```

Figure 4: $\boldsymbol{D}$

```
[[ 3. -1. -1. -1.  0.  0.  0.  0.  0.  0.]
 [-1.  3.  0. -1.  0.  0.  0. -1.  0.  0.]
 [-1.  0.  2.  0.  0.  0.  0. -1.  0.  0.]
 [-1. -1.  0.  2.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  2. -1. -1.  0.  0.  0.]
 [ 0.  0.  0.  0. -1.  1.  0.  0.  0.  0.]
 [ 0.  0.  0.  0. -1.  0.  2.  0.  0. -1.]
 [ 0. -1. -1.  0.  0.  0.  0.  3. -1.  0.]
 [ 0.  0.  0.  0.  0.  0.  0. -1.  2. -1.]
```

Figure 5: $\boldsymbol{L}$

3. By HW2 Problem 3, Neighbor Embedding Slide p.7-p.10 and programming tools(MATLAB, Python...), solve the optimization problem
   minimize $Trace(\boldsymbol{\Psi}^T \boldsymbol{L} \boldsymbol{\Psi})$
   subject to $\boldsymbol{\Psi}^T \boldsymbol{D} \boldsymbol{\Psi} = \boldsymbol{I}_3$
   variables $\boldsymbol{\Psi} \in \mathbb{R}^{10 \times 3}$
   Also, please plot the reduced points $\boldsymbol{z_1}, ..., \boldsymbol{z_{10}}$ in 3-D scatter plot.

Solution: the scatter plot is shown below.

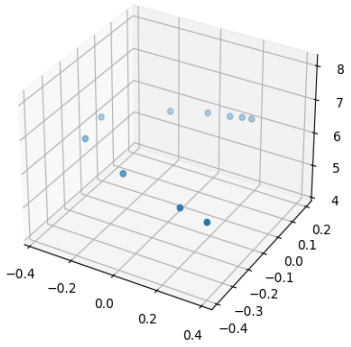


Figure 6: Scatter plot

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

$$\begin{bmatrix} c \\ c \\ \vdots \\ c \end{bmatrix} \tag{1}$$

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 $\boldsymbol{z_1}, ..., \boldsymbol{z_{10}}$ such that $\text{span}\{\boldsymbol{z_1}, ..., \boldsymbol{z_{10}}\} = \mathbb{R}^3$, we need choose the second, third, fourth smallest eigenvalue and the corresponding eigenvectors. Please plot the reduced points by the updated $\boldsymbol{z_1}, ..., \boldsymbol{z_{10}}$ in 3-D scatter plot and verify that whether $Trace(\boldsymbol{\Psi}^T \boldsymbol{L} \boldsymbol{\Psi}) = 1.098$ and $\boldsymbol{\Psi}^T \boldsymbol{D} \boldsymbol{\Psi} = \boldsymbol{I}_3$.

Solution: $\boldsymbol{\Psi}$ and the scatter plot of adjusted reduced points is shown below. It is true that $Trace(\boldsymbol{\Psi}^T \boldsymbol{L} \boldsymbol{\Psi}) = 1.098$ and $\boldsymbol{\Psi}^T \boldsymbol{D} \boldsymbol{\Psi} = \boldsymbol{I}_3$.

```
[[ 0.09069099  0.14189576  0.18831754]
 [ 0.09972695  0.09357859  0.17464619]
 [-0.30654733 -0.00046639  0.15764162]
 [ 0.29471481  0.18210673  0.19459486]
 [-0.09353918  0.25247422 -0.3627844 ]
 [-0.28954605  0.39050752 -0.38899742]
 [ 0.22910952 -0.06404401 -0.28767812]
 [-0.28875405 -0.14249883  0.10571999]
 [-0.0730296  -0.36950084 -0.03650001]
 [ 0.24156895 -0.33528676 -0.17380083]]
```

Figure 7: Psi

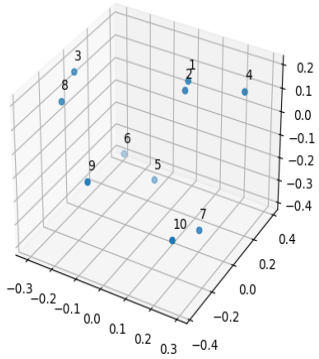


Figure 8: Scatter plot

5. Show that for no matter the graph is, there is an eigenvector of $\boldsymbol{L}$

$$\begin{bmatrix} c \\ c \\ \vdots \\ c \end{bmatrix} \tag{2}$$

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

Solution: Let $\boldsymbol{v} =$

$$\begin{bmatrix} c \\ c \\ \vdots \\ c \end{bmatrix} \tag{3}$$

.
$\boldsymbol{L}\boldsymbol{v}_i = \sum_{k=1}^{N} \boldsymbol{L}_{ik} c = c \sum_{k=1}^{N} \boldsymbol{L}_{ik} = c(\sum_{k=1}^{N} \boldsymbol{D}_{ik} - \sum_{k=1}^{N} \boldsymbol{D}_{ik}) = c(\text{deg of node i} - \text{deg of node i}) = c \times 0 = 0.$
Hence, $\boldsymbol{L}\boldsymbol{v} = \boldsymbol{0} = 0\boldsymbol{v}$, which indicates that $\boldsymbol{v}$ is an eigenvector of $\boldsymbol{L}$ that corresponds to eigenvalue 0.

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

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

Solution: Under the knowledge on Neighbor Embedding Slide p.9 that

$$\text{Trace}(\boldsymbol{\Psi}^T \boldsymbol{L} \boldsymbol{\Psi}) = \frac{1}{2} \sum_{i,j=1,\ldots,N} w_{ij} ||z_i - z_j||_2^2$$

. Let $\boldsymbol{f} = \boldsymbol{\Psi} \in \mathbb{R}^{N \times 1}$, $\text{Trace}(\boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f}) = \boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f}$, $\sum_{i,j=1,\ldots,N} w_{ij} ||z_i - z_j||_2^2 = \sum_{i,j=1,\ldots,N} w_{ij(f_i - f_j)^2}$.
Hence we get $\boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f} = \frac{1}{2} \sum_{1 \le i,j \le N} w_{ij}(f_i - f_j)^2$.

7. Show that if $\boldsymbol{f}$ is an eigenvector of $\boldsymbol{L}$ which corresponds to eigenvalue 0, then $\boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f} = 0$.

Solution: $\boldsymbol{L}\boldsymbol{f} = 0\boldsymbol{f} = \boldsymbol{0}$, hence $\boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f} = 0$.

8. Show that if the graph is connected, the second smallest eigenvalue of $\boldsymbol{L}$ will be nonzero.

Solution: By 6, we know that $\boldsymbol{L}$ is positive semidefinite matrix, which indicates that all the eigenvalues of $\boldsymbol{L}$ is non-negative. From the above result and 5. we deduced that 0 is the smallest eigenvalue, which corresponds to eigenvector $\boldsymbol{v_1} =$

$$\begin{bmatrix} c \\ c \\ \vdots \\ c \end{bmatrix} \tag{4}$$

Also, because $\boldsymbol{L}$ is a symmetric matrix, it its diagonalizable, which means that $\boldsymbol{L}$ has N independent eigenvectors. Suppose $\boldsymbol{v_2}$ is an eigenvector corresponding to the second smallest eigenvalue λ_2. From the above facts, $\boldsymbol{v_2}$ can not be the multiply of $\boldsymbol{v_1}$, which means $\boldsymbol{v_2}$ can not be the form

$$\begin{bmatrix} d \\ d \\ \vdots \\ d \end{bmatrix} \tag{5}$$

On the other hand, because G is a connected graph, we can find a path from node i to node j. We define the nodes on path be $n_1 = i, n_2, ..., n_{k-1}, n_k = j$. $\sum_{1 \le i,j \le N} w_{ij}(v_{2i} - v_{2j})^2 \ge \sum_{1 \le r \le k-1} (v_{2n_r} - v_{2n_r+1})^2 > 0$ due to the fact that v_2 is not the multiply of all one vector. Also by 6., we get $\boldsymbol{v_2}^T \boldsymbol{L} \boldsymbol{v_2} = \boldsymbol{v_2}^T \boldsymbol{\lambda_2} \boldsymbol{v_2} = \boldsymbol{\lambda_2} ||\boldsymbol{v_2}||_2 = \sum_{1 \le i,j \le N} w_{ij}(v_{2i} - v_{2j})^2 > 0$, where $\boldsymbol{\lambda_2}$ is the second smallest eigenvalue. Hence by the above inequality, we show that the second smallest eigenvalue of $\boldsymbol{L}$ will be nonzero.

Problem 3 (Multiclass AdaBoost)(1.5%)

Let $\mathcal{X}$ be the input space, $\mathscr{F}$ 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 $f_t \in \mathscr{F}$ and $\alpha_t \in \mathbb{R}$ for all $t \in [1, T]$. Here for $f \in \mathscr{F}$, 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 $h : \mathcal{X} \to [1, K]$ is defined as

$$x \mapsto \underset{1 \le k \le K}{\operatorname{argmax}} \, g_{T+1}^k(x)$$

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

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

Solution: Given $\mathbf{g}_{t-1} = \left\{g_{t-1}^k\right\}_{k=1}^K$, we update $\mathbf{g}_t = \left\{g_{t-1}^k + \alpha_t f_t^k\right\}_{k=1}^K = \mathbf{g}_{t-1} + \alpha_t \mathbf{f}_t$ as follows:

$$\begin{aligned}
&\mathbf{f}_t \in \underset{f\in\mathcal{F}}{\operatorname{argmin}} \frac{\partial}{\partial\alpha} L\left(\mathbf{g}_{t-1} + \alpha\mathbf{f}\right)\Big|_{\alpha=0} \\
&= \underset{f\in\mathcal{F}}{\operatorname{argmin}} \frac{\partial}{\partial\alpha} \sum_{i=1}^n \exp\left(\left(\frac{1}{K-1}\sum_{k\neq\hat{y}_i} g_{t-1}^k\left(x_i\right) - g_{t-1}^{\hat{y}_i}\left(x_i\right)\right) + \alpha\left(\frac{1}{K-1}\sum_{k\neq\hat{y}_i} f^k\left(x_i\right) - f^{\hat{y}_i}\left(x_i\right)\right)\right)\Big|_{\alpha=0} \\
&= \underset{f\in\mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^n \exp\left(\frac{1}{K-1}\sum_{k\neq\hat{y}_i} g_{t-1}^k\left(x_i\right) - g_{t-1}^{\hat{y}_i}\left(x_i\right)\right)\left(\frac{1}{K-1}\sum_{k\neq\hat{y}_i} f^k\left(x_i\right) - f^{\hat{y}_i}\left(x_i\right)\right) \\
&= \underset{f\in\mathcal{F}}{\operatorname{argmin}} Z_t \mathbb{E}_{i\sim D_t}\left[\frac{1}{K-1}\sum_{k\neq\hat{y}_i} f^k\left(x_i\right) - f^{\hat{y}_i}\left(x_i\right)\right] \\
&= \underset{f\in\mathcal{F}}{\operatorname{argmin}} Z_t \mathbb{E}_{i\sim D_t}\left[\frac{1}{K-1}\cdot 1\left\{f\left(x_i\right)\neq\hat{y}_i\right\} - 1\left\{f\left(x_i\right)=\hat{y}_i\right\}\right] \\
&= \underset{f\in\mathcal{F}}{\operatorname{argmin}} Z_t\left(\frac{K}{K-1}\mathbb{P}_{i\sim D_t}\left[f\left(x_i\right)\neq\hat{y}_i\right] - 1\right) = \underset{f\in\mathcal{F}}{\operatorname{argmin}} \mathbb{P}_{i\sim D_t}\left[f\left(x_i\right)\neq\hat{y}_i\right] \\
&\alpha_t \in \underset{\alpha\in\mathbb{R}}{\operatorname{argmin}} L\left(\mathbf{g}_{t-1} + \alpha\mathbf{f}_t\right) \\
&= \underset{\alpha\in\mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^n \exp\left(\left(\frac{1}{K-1}\sum_{k\neq\hat{y}_i} g_{t-1}^k\left(x_i\right) - g_{t-1}^{\hat{y}_i}\left(x_i\right)\right) + \alpha\left(\frac{1}{K-1}\sum_{k\neq\hat{y}_i} f_t^k\left(x_i\right) - f_t^{\hat{y}_i}\left(x_i\right)\right)\right) \\
&= \underset{\alpha\in\mathbb{R}}{\operatorname{argmin}} Z_t \mathbb{E}_{i\sim D_t}\left[e^{\alpha\left(\frac{1}{K-1}\sum_{k\neq\hat{y}_i} f_t^k\left(x_i\right) - f_t^{\hat{y}_i}\left(x_i\right)\right)}\right] \\
&= \underset{\alpha\in\mathbb{R}}{\operatorname{argmin}} Z_t \mathbb{E}_{i\sim D_t}\left[e^{\frac{\alpha}{K-1}}\cdot 1\left\{f_t\left(x_i\right)\neq\hat{y}_i\right\} + e^{-\alpha}\cdot 1\left\{f_t\left(x_i\right)=\hat{y}_i\right\}\right] \\
&= \underset{\alpha\in\mathbb{R}}{\operatorname{argmin}} Z_t\left(\epsilon_t e^{\frac{\alpha}{K-1}} + e^{-\alpha}\left(1-\epsilon_t\right)\right) = \left\{\frac{K-1}{K}\log\frac{\left(K-1\right)\left(1-\epsilon_t\right)}{\epsilon_t}\right\}
\end{aligned}$$

where

$$Z_t = \sum_{i=1}^n \exp\left(\frac{1}{K-1}\sum_{k\neq\hat{y}_i} g_{t-1}^k\left(x_i\right) - g_{t-1}^{\hat{y}_i}\left(x_i\right)\right)$$

and that D_t is a probability distribution for $t = 1, \cdots, n$ given by

$$D_t(i) = \frac{1}{Z_t}\exp\left(\frac{1}{K-1}\sum_{k\neq\hat{y}_i} g_{t-1}^k\left(x_i\right) - g_{t-1}^{\hat{y}_i}\left(x_i\right)\right)$$

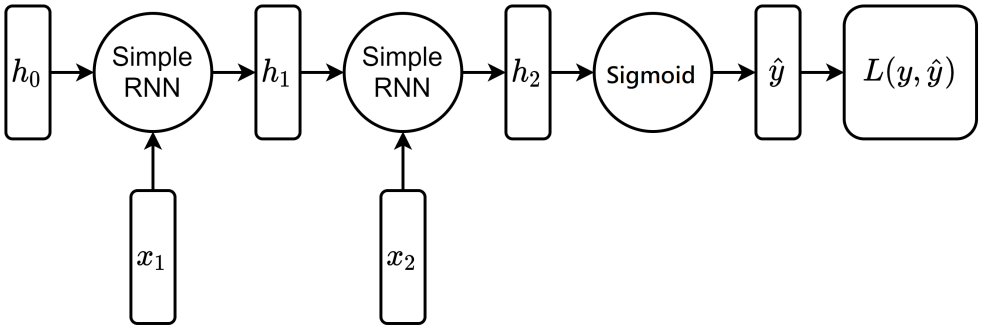
and that $\epsilon_t = \mathbb{P}_{i\sim D_t}\left[f_t\left(x_i\right)\neq\hat{y}_i\right]$ is the error of f_t on training sample weighted by the distribution D_t.

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_x and w_h, such that $h_t = \tanh(w_x x_t + w_h h_{t-1})$, 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 $\frac{\partial L(y,\hat{y})}{\partial w_o}$, $\frac{\partial L(y,\hat{y})}{\partial w_h}$, $\frac{\partial L(y,\hat{y})}{\partial w_x}$ in terms of x_1, x_2, h_0, h_1, h_2, w_x, w_o, and w_h.

Solution:

$$\frac{\partial L(y,\hat{y})}{\partial w_o} = \frac{\partial \hat{y}^2}{\partial w_o} - 2y\frac{\partial \hat{y}}{\partial w_o} = 2(\hat{y}-y)\frac{\exp(-w_0 h_2)h_2}{(1+\exp(-w_o h_2))^2}$$



For the rest of the part, we first calculate two recursive relation.

$$\frac{\partial h_t}{\partial w_x} = \frac{\partial \tanh(w_x x_t + w_h h_{t-1})}{\partial (w_x x_t + w_h h_{t-1})} \frac{\partial (w_x x_t + w_h h_{t-1})}{\partial w_x} = \operatorname{sech}^2(w_x x_t + w_h h_{t-1}) x_t + \operatorname{sech}^2(w_x x_t + w_h h_{t-1}) w_h \frac{\partial h_{t-1}}{\partial w_x}$$

$$\frac{\partial h_t}{\partial w_h} = \frac{\partial \tanh(w_x x_t + w_h h_{t-1})}{\partial (w_x x_t + w_h h_{t-1})} \frac{\partial (w_x x_t + w_h h_{t-1})}{\partial w_h} = \operatorname{sech}^2(w_x x_t + w_h h_{t-1}) h_{t-1} + \operatorname{sech}^2(w_x x_t + w_h h_{t-1}) \frac{\partial h_{t-1}}{\partial w_x}$$

Actually if you can identify this two relation I will give you full credit. The followings are the rest of the part.

$$\frac{\partial L(y, \hat{y})}{\partial w_h} = (\hat{y} - y) \frac{\exp(-w_0 h_2) w_0}{(1 + \exp(-w_o h_2))^2} \frac{\partial h_2}{\partial w_h}$$

$$\frac{\partial L(y, \hat{y})}{\partial w_h} = (\hat{y} - y) \frac{\exp(-w_0 h_2) w_0}{(1 + \exp(-w_o h_2))^2} \frac{\partial h_2}{\partial w_x}$$

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:

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

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

$p_i^j :=$ 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.

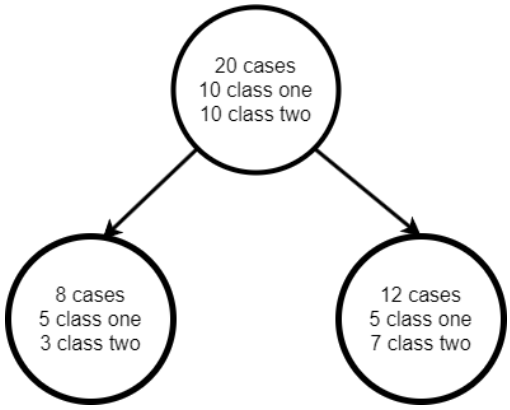
(a) (i) A 50/50 split with the first part containing 80% of positive examples and the second part containing 75% of positive examples.

(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.

Solution:

(i) Gini index = 0.35, Shannon information gain = 0.77



(ii) Gini index = 0.036, Shannon information gain = 0.094

(iii) Gini index = 0.018, Shannon information gain = 0.073

(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.

Solution: The most easy way is adding a weight in front of different classes. You, however, can design your own loss function to satisfy this criteria. It is better if you can make sure that it keeps the same order with the origin when we fix the split or fix the proportion in two nodes.

Version Description

1. First Edition: Finish Problem Solution 1 to 5