Conceptual Questions

3.1 Problem 1

Imagine that you are working on a big and complicated Convolutional Neural Network(CNN) with many convolutional building blocks shown on the right. The convolutional building block has three layers: a convolution layer with kernel $K \in \mathbb{R}^{k\times k}$, stride=1, and no padding, a sigmoid activation layer(with sigmoid function $f(x) = \frac{1}{1+e^{-x}}$), and a 2×2 average pooling layer. The building block input is $X \in \mathbb{R}^{(2n+k-1)\times(2m+k-1)}$, the output from convolution layer is $C \in \mathbb{R}^{2n\times 2m}$, the output from activation layer is $S \in \mathbb{R}^{2n\times 2m}$, and the building block output from the 2×2 average pooling layer is $P \in \mathbb{R}^{n\times m}$. $(2n+k-1,\ 2m+k-1)$

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

$f(x)' = \frac{+e^{-x}}{(1+e^{-x})^2} = \frac{e^{-x}}{e^{-2x}+2e^{-x}+1}$

$= f(x)(1-f(x))$

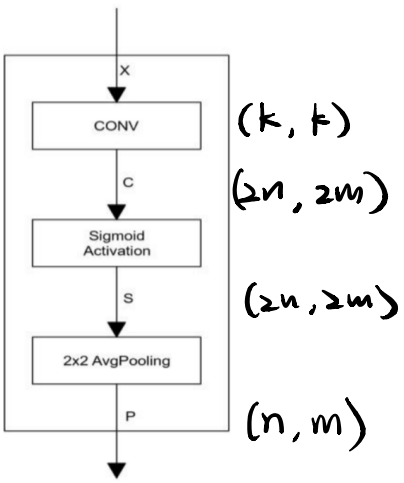


Figure 1: Convolutional Building Block For Problem 1

Given $X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,2m+k-1} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,2m+k-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{2n+k-1,1} & x_{2n+k-1,2} & \cdots & x_{2n+k-1,2m+k-1} \end{bmatrix}$, $S = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,2m} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,2m} \\ \vdots & \vdots & \ddots & \vdots \\ s_{2n,1} & s_{2n,2} & \cdots & s_{2n,2m} \end{bmatrix}$,

and $\frac{\partial loss}{\partial P} = \begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,m} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n,1} & g_{n,2} & \cdots & g_{n,m} \end{bmatrix}$, write down the expressions for $\frac{\partial loss}{\partial S}$, $\frac{\partial loss}{\partial C}$, and $\frac{\partial loss}{\partial K}$, using $g_{i,j}$ for $1 \le i \le n$, $1 \le j \le m$; $s_{d,e}$ for $1 \le d \le 2n$, $1 \le e \le 2m$; and $x_{a,b}$ for $1 \le a \le (2n+k-1)$, $1 \le b \le (2m+k-1)$. [8 pts]

$$\frac{\partial L}{\partial S} = \begin{bmatrix} \frac{\partial L}{\partial s_{1,1}} & \cdots & \frac{\partial L}{\partial s_{1,2m}} \\ \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial s_{2n,1}} & \cdots & \frac{\partial L}{\partial s_{2n,2m}} \end{bmatrix} = \begin{bmatrix} \frac{\partial L}{\partial P_{1,1}}\cdot\frac{\partial P_{1,1}}{\partial s_{1,1}} & \frac{\partial L}{\partial P_{1,1}}\cdot\frac{\partial P_{1,1}}{\partial s_{1,2}} & \cdots & \frac{\partial L}{\partial P_{1,m}}\cdot\frac{\partial P_{1,m}}{\partial s_{1,2m-1}} & \frac{\partial L}{\partial P_{1,m}}\cdot\frac{\partial P_{1,m}}{\partial s_{1,2m}} \\ \frac{\partial L}{\partial P_{1,1}}\cdot\frac{\partial P_{1,1}}{\partial s_{2,1}} & \frac{\partial L}{\partial P_{1,1}}\cdot\frac{\partial P_{1,1}}{\partial s_{2,2}} & \cdots & \frac{\partial L}{\partial P_{1,m}}\cdot\frac{\partial P_{1,m}}{\partial s_{2,2m-1}} & \frac{\partial L}{\partial P_{1,m}}\cdot\frac{\partial P_{1,m}}{\partial s_{2,2m}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial L}{\partial P_{n,1}}\cdot\frac{\partial P_{n,1}}{\partial s_{2n-1,1}} & \frac{\partial L}{\partial P_{n,1}}\cdot\frac{\partial P_{n,1}}{\partial s_{2n-1,2}} & \cdots & \frac{\partial L}{\partial P_{n,m}}\cdot\frac{\partial P_{n,m}}{\partial s_{2n-1,2m-1}} & \frac{\partial L}{\partial P_{n,m}}\cdot\frac{\partial P_{n,m}}{\partial s_{2n-1,2m}} \\ \frac{\partial L}{\partial P_{n,1}}\cdot\frac{\partial P_{n,1}}{\partial s_{2n,1}} & \frac{\partial L}{\partial P_{n,1}}\cdot\frac{\partial P_{n,1}}{\partial s_{2n,2}} & \cdots & \frac{\partial L}{\partial P_{n,m}}\cdot\frac{\partial P_{n,m}}{\partial s_{2n,2m-1}} & \frac{\partial L}{\partial P_{n,m}}\cdot\frac{\partial P_{n,m}}{\partial s_{2n,2m}} \end{bmatrix}$$

$$= \frac{1}{4}\begin{bmatrix} g_{1,1} & g_{1,1} & g_{1,2} & g_{1,2} & \cdots & g_{1,m} & g_{1,m} \\ g_{1,1} & g_{1,1} & g_{1,2} & g_{1,2} & \cdots & g_{1,m} & g_{1,m} \\ g_{2,1} & g_{2,1} & g_{2,2} & g_{2,2} & & \vdots & \vdots \\ g_{2,1} & g_{2,1} & g_{2,2} & g_{2,2} & & & \\ \vdots & \vdots & & & \ddots & \vdots & \vdots \\ g_{n,1} & g_{n,1} & \cdots & & \cdots & g_{n,m} & g_{n,m} \\ g_{n,1} & g_{n,1} & \cdots & & \cdots & g_{n,m} & g_{n,m} \end{bmatrix}$$

$$\frac{\partial L}{\partial C} = \begin{bmatrix} \frac{\partial L}{\partial c_{1,1}} & \cdots & \frac{\partial L}{\partial c_{1,2m}} \\ \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial c_{2n,1}} & \cdots & \frac{\partial L}{\partial c_{2n,2m}} \end{bmatrix} = \begin{bmatrix} \frac{\partial L}{\partial s_{1,1}} \cdot \frac{\partial s_{1,2}}{\partial c_{1,2}} & \cdots & \frac{\partial L}{\partial s_{1,2m-1}} \cdot \frac{\partial s_{1,2m}}{\partial c_{1,2m}} \\ \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial s_{2n,1}} \cdot \frac{\partial s_{2n,2}}{\partial c_{2n,2}} & \cdots & \frac{\partial L}{\partial s_{2n,2m-1}} \cdot \frac{\partial s}{\partial c_{2n,2m}} \end{bmatrix}$$

$$= \frac{1}{4}\begin{bmatrix} g_{1,1}f'(c_{1,1}) & g_{1,1}f'(c_{1,2}) & g_{1,2}f'(c_{1,3}) & g_{1,2}f'(c_{1,4}) & \cdots & g_{1,m}f'(c_{1,2m-1}) & g_{1,m}f'(c_{1,2m}) \\ g_{1,1}f'(c_{2,1}) & g_{1,1}f'(c_{2,2}) & g_{1,2}f'(c_{2,3}) & g_{1,2}f'(c_{2,4}) & \cdots & g_{1,m}f'(c_{2,2m-1}) & g_{1,m}f'(c_{2,2m}) \\ g_{2,1}f'(c_{3,1}) & g_{2,1}f'(c_{3,2}) & g_{2,2}f'(c_{3,3}) & g_{2,2}f'(c_{3,4}) & & \vdots & \vdots \\ g_{2,1}f'(c_{4,1}) & g_{2,1}f'(c_{4,2}) & g_{2,2}f'(c_{4,3}) & g_{2,2}f'(c_{4,4}) & & & \\ \vdots & \vdots & & & \ddots & \vdots & \vdots \\ g_{n,1}f'(c_{2n-1,1}) & g_{n,1}f'(c_{2n-1,2}) & \cdots & & \cdots & g_{n,m}f'(c_{2n-1,2m-1}) & g_{n,m}f'(c_{2n-1,2m}) \\ g_{n,1}f'(c_{2n,1}) & g_{n,1}f'(c_{2n,2}) & \cdots & & \cdots & g_{n,m}f'(c_{2n,2m-1}) & g_{n,m}f'(c_{2n,2m}) \end{bmatrix}$$

where $f'(C_{d,e}) = f(C_{d,e})(1-f(C_{d,e})) = S_{d,e}(1-S_{d,e})$ for $1 \le d \le 2n$, $1 \le e \le 2m$

For $\frac{\partial L}{\partial K}$, firstly we consider forward:

$$\begin{bmatrix} x_{1,1} & \cdots & x_{1,2m+k-1} \\ \vdots & \ddots & \vdots \\ x_{2n+k-1,1} & \cdots & x_{2n+k-1,2m+k-1} \end{bmatrix} * \begin{bmatrix} K_{1,1} & \cdots & K_{1,k} \\ \vdots & \ddots & \vdots \\ K_{k,1} & \cdots & K_{k,k} \end{bmatrix} = \begin{bmatrix} C_{1,1} & \cdots & C_{1,2m} \\ \vdots & \ddots & \vdots \\ C_{2n,1} & \cdots & C_{2n,2m} \end{bmatrix}$$

According to the property of convolution, we have

$$\frac{\partial L}{\partial k} = \begin{bmatrix} \frac{\partial L}{\partial k_{1,1}} & \cdots & \frac{\partial L}{\partial k_{1,k}} \\ \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial k_{k,1}} & \cdots & \frac{\partial L}{\partial k_{k,k}} \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,\,2m+k-1} \\ \vdots & \ddots & \vdots \\ x_{2n+k-1,\,1} & \cdots & x_{2n+k-1,\,2m+k-1} \end{bmatrix} * \begin{bmatrix} \frac{\partial L}{\partial C_{1,1}} & \cdots & \frac{\partial L}{\partial C_{1,2m}} \\ \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial_{2n,1}} & \cdots & \frac{\partial L}{\partial_{2n,2m}} \end{bmatrix}$$

$$= \begin{bmatrix} x_{1,1} & \cdots & x_{1,\,2m+k-1} \\ \vdots & \ddots & \vdots \\ x_{2n+k-1,\,1} & \cdots & x_{2n+k-1,\,2m+k-1} \end{bmatrix} * \frac{1}{4} \begin{bmatrix} g_{1,1}f'(C_{1,1}) & g_{1,1}f'(C_{1,2}) & g_{1,2}f'(C_{1,3}) & g_{1,2}f'(C_{1,4}) & \cdots & g_{1,m}f'(C_{1,2m-1}) & g_{1,m}f'(C_{1,2m}) \\ g_{1,1}f'(C_{2,1}) & g_{1,1}f'(C_{2,2}) & g_{1,2}f'(C_{2,3}) & g_{1,2}f'(C_{2,4}) & \cdots & g_{1,m}f'(C_{2,2m-1}) & g_{1,m}f'(C_{2,2m}) \\ g_{2,1}f'(C_{3,1}) & g_{2,1}f'(C_{3,2}) & g_{2,2}f'(C_{3,3}) & g_{2,2}f'(C_{3,4}) & & \vdots & \vdots \\ g_{2,1}f'(C_{4,1}) & g_{2,1}f'(C_{4,2}) & g_{2,2}f'(C_{4,3}) & g_{2,2}f'(C_{4,4}) & & & \\ \vdots & \vdots & & & \ddots & \vdots & \vdots \\ g_{n,1}f'(C_{2n-1,1}) & g_{n,1}f'(C_{2n-1,2}) & \cdots & & \cdots & g_{n,m}f'(C_{2n-1,2m-1}) & g_{n,m}f'(C_{2n-1,2m}) \\ g_{n,1}f'(C_{2n,1}) & g_{n,1}f'(C_{2n,2}) & \cdots & & \cdots & g_{n,m}f'(C_{2n,2m-1}) & g_{n,m}f'(C_{2n,2m}) \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^{2n}\sum_{j=1}^{2m} x_{i,j}\, g_{\lfloor\frac{i+1}{2}\rfloor,\lfloor\frac{j+1}{2}\rfloor} f'(C_{i,j}) & \sum_{i=1}^{2n}\sum_{j=1}^{2m} x_{i,j+1}\, g_{\lfloor\frac{i+1}{2}\rfloor,\lfloor\frac{j+1}{2}\rfloor} f'(C_{i,j+1}) & \cdots & \sum_{i=1}^{2n}\sum_{j=1}^{2m} x_{i,j+k-1}\, g_{\lfloor\frac{i+1}{2}\rfloor,\lfloor\frac{j+1}{2}\rfloor} f'(C_{i,j+k-1}) \\ \sum_{i=1}^{2n}\sum_{j=1}^{2m} x_{i+1,j}\, g_{\lfloor\frac{i+1}{2}\rfloor,\lfloor\frac{j+1}{2}\rfloor} f'(C_{i+1,j}) & \ddots & & \vdots \\ \vdots & & & \vdots \\ \sum_{i=1}^{2n}\sum_{j=1}^{2m} x_{i+k-1,j}\, g_{\lfloor\frac{i+1}{2}\rfloor,\lfloor\frac{j+1}{2}\rfloor} f'(C_{i+k-1,j}) & \cdots & \cdots & \sum_{i=1}^{2n}\sum_{j=1}^{2m} x_{i+k-1,j+k-1}\, g_{\lfloor\frac{i+1}{2}\rfloor,\lfloor\frac{j+1}{2}\rfloor} f'(C_{i+k-1,j+k-1}) \end{bmatrix}$$

where $f'(C_{d,e}) = f(C_{d,e})(1-f(C_{d,e})) = S_{d,e}(1-S_{d,e})$ for $1 \le d \le 2n$, $1 \le e \le 2m$

3.2 Problem 2

3.2.1 Problem 2.1

Please state at least two advantages of one-stage models compared with multi-stage models in object detection tasks. [3 pts]

3.2.2 Problem 2.2

Please state briefly why Non-maximum suppression(NMS) is generally needed for bounding-box based approaches in object detection tasks. [3 pts]

3.2.1 Advantage 1: Since there is only one stage, training and inference speed will be faster.

Advantage 2: Since the one-stage model updates parameters as a whole out once, the generalization ability can be better.

3.2.2. Because during detection, usually we get a lot of bounding box, which may have many places overlapped. Therefore we need NMS algorithm to filter the bounding boxes with low confidence of containing the objects.

3.3 Problem 3

Imagine we have a Convolutional Neural Network with its structure shown below:

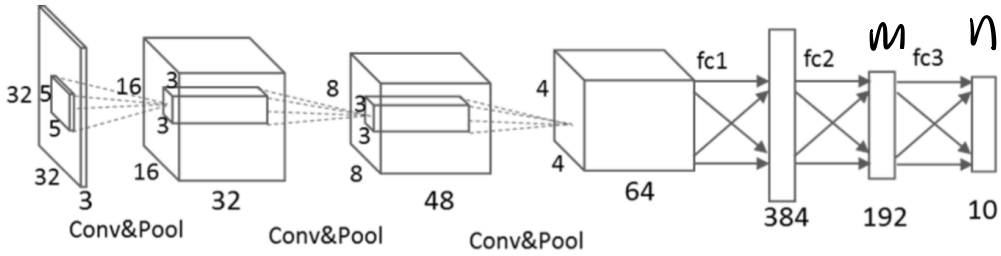


Figure 2: CNN For Problem 3

Please calculate the total number of parameters (including all weights and all biases, suppose biases are used wherever possible) of this CNN. [6 pts]

$(5\times5\times3+1)\times32+(3\times3\times32+1)\times48+(3\times3\times48+1)\times64$

$+(4\times4\times64+1)\times384+(384+1)\times192+(192+1)\times10=513466$

3.4 Problem 4

Suppose we have a neural network model with a softmax layer shown as below:

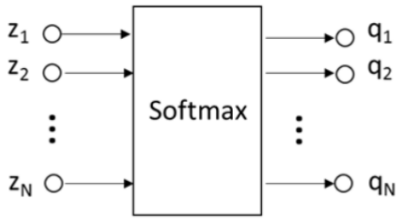


Figure 3: Softmax layer For Problem 4

Where $q_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$ for $1 \leq i \leq N$.

We want our softmax layer to learn a mapping from $Z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}$ to $P = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{bmatrix}$,

where we have

$$\sum_{i=1}^{N} p_i = 1$$

Given the objective function

$$E = -\sum_{i=1}^{N} (p_i \times ln(q_i))$$

Please prove: $\frac{\partial E}{\partial z_x} = q_x - p_x$, for $1 \leq x \leq N$. [8 pts]

$$\frac{\partial E}{\partial z_x} = \frac{\partial}{\partial z_x} - \sum_{i=1}^{N} \left(P_i \times \ln(q_i)\right) = -\sum_{i=1}^{N} \left(\frac{\partial}{\partial z_x} P_i \times \ln(q_i)\right) = -\sum_{i=1}^{N} \frac{P_i}{q_i} \cdot \frac{\partial q_i}{\partial z_x}$$

$$= -\sum_{i=1}^{N} \frac{P_i}{q_i} \cdot \frac{\partial}{\partial z_x} \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} = -\sum_{i=1, i\neq x} \frac{P_i}{q_i} \frac{\partial}{\partial z_x} \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} - \frac{P_x}{q_x} \frac{\partial}{\partial z_x} \frac{e^{z_x}}{\sum_{j=1}^{N} e^{z_j}}$$

$$= -\sum_{i=1, i\neq x}^{N} \frac{P_i}{q_i} \frac{-e^{z_i} \cdot e^{z_x}}{\left(\sum_{j=1}^{N} e^{z_j}\right)^2} - \frac{P_x}{q_x} \frac{e^{z_x} \sum_{j=1}^{N} e^{z_j} - e^{z_x} e^{z_x}}{\left(\sum_{j=1}^{N} e^{z_j}\right)^2}$$

$$= \sum_{i=1, i\neq x}^{N} \frac{P_i}{q_i} q_i \cdot q_x - \frac{P_x}{q_x}(q_x - q_x^2) = \sum_{i=1}^{N} P_i q_x - P_x = q_x - P_x$$

3.5 Problem 5

Suppose we have a simple neural network shown as below:

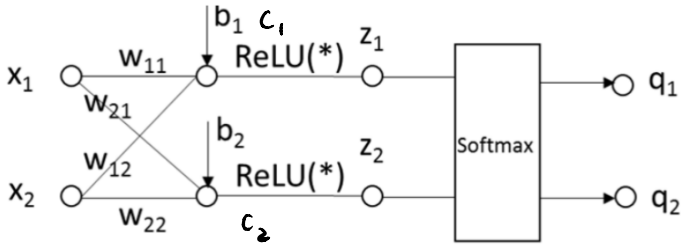


Figure 4: Neural Network For Problem 5

$$q_1 = \frac{e^{z_1}}{e^{z_1}+e^{z_2}}$$

$$\frac{\partial q_1}{\partial z_1} = \frac{e^{z_1}(e^{z_1}+e^{z_2}) - e^{z_1}e^{z_1}}{(\quad)^2} = q_1 - q_1^2$$

$$\frac{\partial q_1}{\partial z_2} = -q_1 q_2$$

The initial parameters of this simple neural network model are: $W = \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \mathrm{ReLU}\left(\begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \right)$$

We use batch size = 1, learning rate $\alpha = 0.02$, objective function $E = -p_1 ln(q_1) - p_2 ln(q_2)$, in the first training iteration our input sample (X, P) is: $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, P = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix}$.

If we use gradient decent formula

$$W' = W - \alpha \times \frac{\partial E}{\partial W}, \quad B' = B - \alpha \times \frac{\partial E}{\partial B}$$

during model training, please calculate the new parameters in our simple neural network W and B after the first iteration in training. [7 pts]

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \mathrm{ReLU}\left(\begin{bmatrix} w_{1,1} & w_{1,2} & b_1 \\ w_{2,1} & w_{2,2} & b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \right) \qquad H = \begin{bmatrix} w_{1,1} & w_{1,2} & b_1 \\ w_{2,1} & w_{2,2} & b_2 \end{bmatrix}$$

$$\frac{\partial E}{\partial H} = \begin{bmatrix} \frac{\partial E}{\partial w_{1,1}} & \frac{\partial E}{\partial w_{1,2}} & \frac{\partial E}{\partial b_1} \\ \frac{\partial E}{\partial w_{2,1}} & \frac{\partial E}{\partial w_{2,2}} & \frac{\partial E}{\partial b_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial q_1}\frac{\partial q_1}{\partial z_1}\frac{\partial z_1}{\partial c_1}\frac{\partial c_1}{\partial w_{1,1}} & \frac{\partial E}{\partial q_1}\frac{\partial q_1}{\partial z_1}\frac{\partial z_1}{\partial c_1}\frac{\partial c_1}{\partial w_{1,2}} & \frac{\partial E}{\partial q_1}\frac{\partial q_1}{\partial z_1}\frac{\partial z_1}{\partial c_1}\frac{\partial c_1}{\partial b_1} \\ \frac{\partial E}{\partial q_2}\frac{\partial q_2}{\partial z_2}\frac{\partial z_2}{\partial c_2}\frac{\partial c_2}{\partial w_{2,1}} & \frac{\partial E}{\partial q_2}\frac{\partial q_2}{\partial z_2}\frac{\partial z_2}{\partial c_2}\frac{\partial c_2}{\partial w_{2,2}} & \frac{\partial E}{\partial q_2}\frac{\partial q_2}{\partial z_2}\frac{\partial z_2}{\partial c_2}\frac{\partial c_2}{\partial b_2} \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{P_1}{q_1}(q_1 - q_1^2)\frac{\partial z_1}{\partial c_1}x_1 & -\frac{P_1}{q_1}(q_1 - q_1^2)\frac{\partial z_1}{\partial c_1}x_2 & -\frac{P_1}{q_1}(q_1 - q_1^2)\frac{\partial z_1}{\partial c_1} \\ -\frac{P_2}{q_2}(q_2 - q_2^2)\frac{\partial z_2}{\partial c_2}x_1 & -\frac{P_2}{q_2}(q_2 - q_2^2)\frac{\partial z_2}{\partial c_2}x_2 & -\frac{P_2}{q_2}(q_2 - q_2^2)\frac{\partial z_2}{\partial c_2} \end{bmatrix}$$

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \mathrm{ReLU}\left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right) = \mathrm{ReLU}\begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Rightarrow \begin{cases} \frac{\partial z_1}{\partial c_1} = 1 \\ \frac{\partial z_2}{\partial c_2} = 0 \end{cases} \Rightarrow \begin{cases} q_1 = \frac{e}{e+1} \\ q_2 = \frac{1}{e+1} \end{cases}$$

$$\frac{\partial E}{\partial H} = \begin{bmatrix} -0.3\frac{1}{e+1} & -0.3\frac{1}{e+1} & -0.3\frac{1}{e+1} \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -0.081 & -0.081 & -0.081 \\ 0 & 0 & 0 \end{bmatrix}$$

Therefore, the new parameters are:

$$W = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} - 0.02 \begin{bmatrix} -0.081 & -0.081 \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1.0016 & 0.0016 \\ 0 & -1 \end{bmatrix}$$

$$b = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - 0.02 \begin{bmatrix} -0.081 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.0016 \\ 0 \end{bmatrix}$$

3.6 Problem6

In recurrent neural network, define loss function:

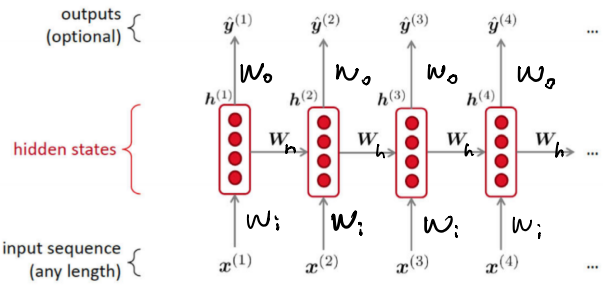
$$L = \sum_{t=0}^{T} L_t$$

Differentiate L_t with respect to W:

$$\frac{\partial L_t}{\partial W^o} = \sum_{t=0}^{T} \frac{\partial L_t}{\partial y_t} \frac{\partial y_t}{\partial W^o} \quad \text{.....(1)}$$

$$\frac{\partial L_t}{\partial W^i} = \sum_{t=0}^{T} \sum_{k=0}^{t} \frac{\partial L_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \left(\prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_k}{\partial W^i} \quad \text{.....(2)}$$

$$\frac{\partial L_t}{\partial W^h} = \sum_{t=0}^{T} \sum_{k=0}^{t} \frac{\partial L_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \left(\prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_k}{\partial W^h} \quad \text{.....(3)}$$



Define:

$$h_t = \sigma(W^i x_t + W^h h_{t-1})$$

where σ denotes sigmoid activation function.

Based on equation (2) or (3), explain why there will be gradient exploding and gradient vanishing. [5pts]

The key point is the $\left(\prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} \right)$ term

$\frac{\partial h_j}{\partial h_{j-1}} = (h_j * (1 - h_j)) W^h$ (since $\sigma'(x) = \sigma(x)(1 - \sigma(x))$)

Therefore, if $\left\| \frac{\partial h_j}{\partial h_{j-1}} \right\| = \left\| (h_j * (1 - h_j)) W^h \right\| < 1$, then

after multiplying t times the gradient will shrink exponentially to 0, which is called the gradient vanishing.

On the contrary, if $\left\| \frac{\partial h_j}{\partial h_{j-1}} \right\| = \left\| (h_j * (1 - h_j)) W^h \right\| > 1$, then

the gradient may expand exponentially to $+\infty$, which is called the gradient exploding.