DL-Assignments

Shuang Wu
East China Normal University
10164102141@stu.ecnu.edu.cn

Abstract

Deep Learning assignments for DATA0031132019.01 in DaSE 2020 Fall. These assignments should be handed in September but I forgot.

1 Chapter 4

Feedforward neural network:

1.1 Problem 4.1

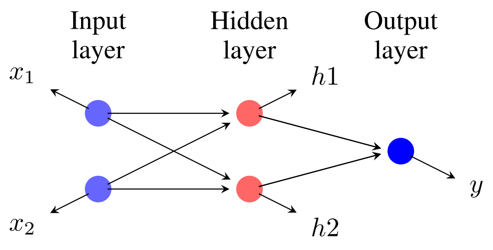
Let the loss function be, the update formula for the parameters of the first layer is

$$\frac{\partial J}{\partial w^{(1)}} = \frac{\partial J}{\partial \sigma} \cdot \frac{\partial \sigma}{\partial w^{(1)}} = \frac{\partial J}{\partial \sigma} \cdot \sigma(1-\sigma) \cdot x$$

where $\sigma, 1-\sigma \in (0,1)$, $\frac{\partial J}{\partial \sigma}$ is the same for all the ω for each neuron in the first layer, so all the gradient symbols are the same. That is, they are either positive or negative. Therefore, when there are positive and negative in the optimal solution, all the ωs of the neuron will update in the same direction, resulting in the phenomenon of "Z" discounted update, and the convergence speed will be very slow.

1.2 Problem 4.2

The topology goes like



$$h = f\left(W^T \vec{x} + \vec{b}\right)$$
$$f = \max\{0, z\}$$
$$y = \vec{w}^T \vec{h} + \vec{b}$$

1.3 Problem 4.3

For example, the initialization of parameters prevents some neurons from being activated at the beginning, and the parameters cannot be updated during further iterations. Or once this neuron parameter causes $wx + b < 0$ in an iterative update, it will cause "death". Fix the *ReLU* function, use *ReLU* with leak (*LeakyReLU*) as activation function

$$LeakyReLU = \begin{cases} x & x > 0 \\ \gamma x & x \le 0 \end{cases}$$

1.4 Problem 4.4

The diverge goes as

$$\text{swish}(x) = x\sigma(\beta x)$$
$$(\text{swish}(x))' = \sigma(\beta x) + \beta x \sigma(\beta x)(1-\sigma(\beta x))$$

1.5 Problem 4.5

If fully connected, the number of arguments is

$$(L-1)\left(\frac{N-1}{L-1}\right)\left(\frac{N-1}{L-1}+1\right) + \frac{N-1}{L-1} + 1 = N\left(\frac{N-1}{L-1}+1\right)$$

1.6 Problem 4.6

Cannot figure out. And thanks to my classmate Zhangyue Yin, we find a solution in arxiv.org.

1.7 Problem 4.7

The data required to fit the bias accurately is usually much less than the fit weight. Each weight specifies how the two variables interact. We need to observe these two variables under various conditions to fit the weights well. And each bias controls only one univariate. This means that we don't regularize it and don't cause much variance. In addition, regularization bias parameters may cause significant underfitting.

1.8 Problem 4.8

There may be two problems

1. It may cause the neuron output to be all zero during forward propagation (such as the ReLU activation function) and the back-propagation gradient to be all zero.
2. Even if the neuron output is not 0 (such as the Sigmoid function), all parameters are initialized to 0, which will cause all neurons in the same layer to learn the same features during training.

1.9 Problem 4.9

No. Sine the gradient decreases exponentially, and it does not make sense to increase the learning rate. It may even be too high a learning rate at first, leaving it on the wrong path forever

2 Chapter 5

Convolutional neural network:

2.1 Problem 5.1

The solution goes as follow
Let $Y = W\tilde{\otimes}X = W \otimes \tilde{X}, Z = X\tilde{\otimes}W = X \otimes \tilde{W}$, then we have

$$y_{ij} = \sum_{u=1}^{m}\sum_{v=1}^{n} w_{u,v}x_{i-u+1,j-v+1}$$
$$z_{ij} = \sum_{p=1}^{M}\sum_{q=1}^{N} x_{p,q}w_{i-p+1,j-q+1}$$

then we have $\forall i, j$, if $u = i - p + 1$, we have $p = i - u + 1$.
For y_{ij}, x is not in padding, to satisfy the requirement $(i + 1 > i - u + 1 > m - 1) \wedge (j + 1 > j - v + 1 > n - 1)$, we need $(m - 1 < p < i + 1) \wedge (n - 1 < q < j + 1)\} \subset \{(0 < p < M) \wedge (0 < q < N)$,
So for y_{ij}, x is not the terms in padding, z_{ij} also exists, and the remaining terms are all 0.
It can be proved by the same reason that for z_{ij}, the term ω is not in padding, the term exists in y_{ij}.
In summary, $y_{ij} = z_{ij}$.
So, wide convolution is commutative.

2.2 Problem 5.3

The costs are

- Case 1:
  Time complexity - $2 \times 3 \times 3 \times 256 \times 100 \times 100 \times 256$;
  Space complexity - $3 \times 3 \times 256 \times 256$;
- Case 2:
  Time complexity - $2 \times 1 \times 1 \times 64 \times 100 \times 100 \times 256 + 2 \times 3 \times 3 \times 64 \times 100 \times 100 \times 256$;
  Space complexity - $1 \times 1 \times 64 \times 256 + 3 \times 3 \times 64 \times 256$.

2.3 Problem 5.4

Let the input matrics is $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{11} & a_{32} & a_{43} \end{bmatrix}$, the core is $W = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}$.
Then the transform goes like

$$\begin{bmatrix} w_{11} & w_{21} & 0 \\ 0 & w_{11} & w_{21} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{11} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$
$$+ \begin{bmatrix} w_{12} & 0 \\ 2 & 0 \\ w_{12} & w_{22} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

2.4 Problem 5.5

The gradient goes as

- $\nabla y = \max(x_1, \cdots, x_d) = (0, \cdots, 0, 1, 0, \cdots, 0)^T$ where $i = \sup\{x_i\}$;
- $y = \arg\max(x_1, \cdots, x_d)$ does not have a gradient.

2.5 Problem 5.6

Since we know that

$$Z^{(l+1)} = \sum_{d=1}^{D} W_d^{(l+1)} \otimes X_d^{(l)}$$
$$\delta^{(l)} = f_l'\left(Z^{(l)}\right) \odot \sum_{d=1}^{D}\left(\operatorname{rot}180\left(W_d^{(l+1)}\right)\right)\bar{\otimes}\delta^{(l+1)}\Big)$$

we have $\delta^{(i)} = \sum_{d=1}^{D}\left(\operatorname{rot}180\left(W_d^{(l+1)}\right)\bar{\otimes}\delta^{(l+1)}\right) = \sum_{d=1}^{D}\left(\left(W_d^{(l+1)}\right)^T\bar{\otimes}\delta^{(l+1)}\right)$.

2.6 Problem 5.7

the p goes as

$$n = n - (m + (m-1)(d-1)) + 2p + 1$$
$$p = \frac{d(m-1)}{2}$$

3 Chapter 6

Convolutional neural network:

3.1 Problem 6.1

Delayed neural network is to add a delayer to the non-output layer in the feed-forward network to record the recent neuron output.

$$h_t^{(l+1)} = f\left(h_t^{(l)}, h_{t-1}^{(l)}, \cdots, h_{t-p}^{(l)}\right)$$

A convolutional neural network performs a convolution calculation on the input matrix.
Recurrent neural network processes time-series data through self-feedback neurons

$$h_t = f(h_{t-1}, x_t)$$

Delay neural networks share weights in the time dimension to reduce the number of parameters. Therefore, for sequence input, a delayed neural network is equivalent to a convolutional neural network.
The common feature of delayed neural networks and recurrent neural networks is that they both want to use the previous or previous outputs to retain historical information, but recurrent neural networks use self-feedback and use less space.
Both convolutional neural networks and recurrent neural networks can process time series data, but convolutional neural networks have a wider range of applications and are used in more fields.

3.2 Problem 6.2

- The $\frac{\partial L}{\partial W}$ is

$$\begin{gathered}\frac{\partial L_t}{\partial w_{ij}}=\textstyle\sum_{k=1}^{t}\frac{\partial^{+} z_k}{\partial w_{ij}}\frac{\partial L_t}{\partial z_k}\\ \frac{\partial^{+} z_k}{\partial w_{ij}}=\left[0,\cdots,\left[x_{k-1}\right]_j,\cdots,0\right]\\ \delta_{t,k}=\frac{\partial L_t}{\partial z_k}\\ \frac{\partial L_t}{\partial W}=\textstyle\sum_{k=1}^{t}\delta_{t,k}x_k^T\\ \frac{\partial L}{\partial W}=\textstyle\sum_{t=1}^{T}\sum_{k=1}^{t}\delta_{t,k}x_k^T\end{gathered}$$

- The $\frac{\partial L}{\partial b}$ goes as

$$\begin{aligned}\frac{\partial L_t}{\partial b_i}&=\sum_{k=1}^{t}\frac{\partial L_t}{\partial z_k}\\ \delta_{t,k}&=\frac{\partial L_t}{\partial z_k}\\ \frac{\partial L}{\partial b}&=\sum_{t=1}^{T}\sum_{k=1}^{t}\delta_{t,k}\end{aligned}$$

3.3 Problem 6.3

Since

$$\begin{aligned}\frac{\partial \mathcal{L}_t}{\partial h_{k-1}}=\frac{\partial h_k}{\partial h_{k-1}}\frac{\partial \mathcal{L}_t}{\partial h_k}&=\frac{\partial\left(h_{k-1}+g\left(x_k,h_{k-1};\theta\right)\right)}{\partial h_{k-1}}\frac{\partial \mathcal{L}_t}{\partial h_k}\\ \delta_{t,k}&=\prod_{i=k}^{t-1}\frac{\partial\left(h_{i-1}+g\left(x_i,h_{i-1};\theta\right)\right)}{\partial h_{i-1}}\delta_{t,t}\end{aligned}$$

In which $\frac{\partial\left(h_{k-1}+g\left(x_k,h_{k-1};\theta\right)\right)}{\partial h_{k-1}}>1$, so we may have *gradient explosion*.
The problem of gradient explosion can be solved by using a gating-based recurrent neural network.

3.4 Problem 6.4

LSTM goes as

$$\begin{aligned}c_t&=f_t\odot c_{t-1}+i_t\odot\tilde{c}_t\\ h_t&=o_t\odot\tanh\left(c_t\right)\\ z_{\tilde{c},t}&=W_c x_t+U_c h_{t-1}+b_c\\ \tilde{c}_t&=\tanh\left(z_{\delta,t}\right)\\ \\ z_{t,t}&=W_t x_t+U_f h_{t-1}+b_f\\ z_{t,t}&=W_o x_t+U_o h_{t-1}+b_f\\ z_{o,t}&=W_o x_t+U_o h_{t-1}+b_o\\ \\ t_{o,t}&=\sigma\left(z_{f,t}\right)\\ o_t&=\sigma\left(z_{o,t}\right)\end{aligned}$$

Let the output value of LSTM be h_t at time and define the error term at time t as:

$$\begin{aligned}\delta_t&=\frac{\partial L_t}{\partial h_t}\\ \delta_{f,t}&=\frac{\partial L}{\partial z_{f,t}}\\ \delta_{i,t}&=\frac{\partial L}{\partial z_{i,t}}\\ \delta_{\tilde{c},t}&=\frac{\partial L}{\partial z_{o,t}}\\ \delta_{o,t}&=\frac{\partial L}{\partial z_{o,t}}\end{aligned}$$

Then the bias at time $t-1$ goes like

$$\delta_{t-1}^T=\frac{\partial L}{\partial h_{t-1}}=\frac{\partial L}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}}=\delta_t^T\frac{\partial h_t}{\partial h_{t-1}}$$

Plus

$$\begin{aligned}h_t&=o_t\odot\tanh\left(c_t\right)\\ c_t&=f_t\odot c_{t-1}+i_t\odot\overline{c_t}\end{aligned}$$

we have

$$\begin{aligned}\delta_{t-1}^T=&\delta_t^T\left(\frac{\partial h_t}{\partial o_t}\frac{\partial h_t}{\partial z_{o,t}}\frac{\partial z_{o,t}}{\partial h_{t-1}}+\frac{\partial h_t}{\partial c_t}\frac{\partial c_t}{\partial f_t}\;\frac{\partial f_t}{\partial z_{f,t}}\frac{\partial z_{f,t}}{\partial h_{t-1}}+\right.\\ &\left.\frac{\partial h_t}{\partial c_t}\frac{\partial c_t}{\partial i_t}\frac{\partial i_t}{\partial z_{i,t}}\frac{\partial z_{i,t}}{\partial h_{t-1}}\right)\\ =&\delta_{o,t}^T\frac{\partial z_{o,t}}{\partial h_{t-1}}+\delta_{f,t}^T\frac{\partial z_{f,t}}{\partial h_{t-1}}+\delta_{i,t}^T\frac{\partial z_{i,t}}{\partial h_{t-1}}+\delta_{\varepsilon,t}^T\frac{\partial z_{\bar{c}_t,t}}{\partial h_{t-1}}\end{aligned}$$

and $\frac{\partial h_{\mathrm{t}}}{\partial o_t}=\operatorname{diag}\left[\tanh\left(c_t\right)\right]$, $\frac{\partial h_t}{\partial c_t}=\operatorname{diag}\left[o_t\odot\left(1-\tanh\left(c_t\right)^2\right)\right]$, $\frac{\partial c_t}{\partial f_t}=\operatorname{diag}\left(c_{t-1}\right)$, $\frac{\partial c_t}{\partial t_t}=\operatorname{diag}\left[\varepsilon_t\right]$, $\frac{\partial c_l}{\partial \varepsilon_l}=\operatorname{diag}\left[\mathrm{i}_t\right]$, $\frac{\partial o_1}{\partial z_{ot}}=\operatorname{diag}\left[o_t\odot\left(1-o_t\right)\right]$, $\frac{\partial z_{0,t}}{\partial h_{t-1}}=W_0$, $\frac{\partial f_t}{\partial z_{f,t}}=\operatorname{diag}\left[f_t\odot\left(1-f_t\right)\right]$, $\frac{\partial z_{f,t}}{\partial h_{t-1}}=W_f$, $\frac{\partial i_t}{\partial z_{ij}}=\operatorname{diag}\left[i_t\odot\left(1-i_t\right)\right]$, $\frac{\partial z_{it}}{\partial h_{t-1}}=W_i$, $\frac{\partial \bar{c}_1}{\partial z_{\bar{k},k}}=\operatorname{diag}[1-\overline{c_1^2}]$, $\frac{\partial z_{zt}}{\partial h_{t-1}}=W_\epsilon$.
Then we will have

$$\begin{aligned}\delta_{t-1}&=\delta_{e,t}^T\frac{\partial z_{0t}}{\partial h_{t-1}}+\delta_{t,t}^T\frac{\partial z_{ts}}{\partial h_{t-1}}+\delta_{t,t}^T\frac{\partial z_{t,t}}{\partial h_{t-1}}+\delta_{t,t}^T\frac{\partial z_{dt}}{\partial h_{t-1}}\\ &=\delta_{e,t}^T W_e+\delta_{f,t}^T W_f+\delta_{t,t}^T W_t+\delta_{e,t}^T W_e\\ \delta_{o,t}^T&=\delta_t^T\odot\tanh\left(c_t\right)\odot o_t\odot\left(1-o_t\right)\\ \delta_{t,t}^T&=\delta_t^T\odot o_t\odot\left(1-\tanh\left(c_t\right)^2\right)\odot c_{t-1}\odot f_t\Theta\left(1-f_t\right)\\ \delta_{t,t}^T&=\delta_t^T\odot o_t\odot\left(1-\tanh\left(c_t\right)^2\right)\odot\varepsilon_t\odot t_t\odot\left(1-i_t\right)\\ \delta_{t,t}^T&=\delta_i^T\odot o_t\odot\left(1-\tanh\left(c_t\right)^2\right)\odot i_t\odot\left(1-\dot{e}^2\right)\end{aligned}$$

Accordingly, the equation at time t becomes

$$\delta_k^T=\prod_{f=k}^{t}-1\delta_{\alpha j}^T W_o+\delta_{f,j}^T W_f+\delta_{i,j}^T W_i+\delta_{\ell,j}^T W_e$$

Thus the bias in the previous layer is

$$\begin{aligned}\delta_{\mathrm{t}}^{[-1} &= \tfrac{\partial L}{\partial z_t^{j-1}} \\ x_t^l &= f^{l-1}\left(z_t^{l-1}\right)\end{aligned}$$

$$\begin{aligned}&\frac{\partial L}{\partial z_t^{l-1}} \\ &= \frac{\partial L}{\partial z_{f,t}^t}\frac{\partial z_{f,t}^t}{\partial x_t^I}\frac{\partial x_t^t}{\partial z_t^{L-1}} + \frac{\partial L}{\partial z_{t,t}^L}\frac{\partial z_{t,t}^L}{\partial x_t^L}\frac{\partial x_t^t}{\partial z_{t,t}^{t-1}} \\ &+ \frac{\partial L}{\partial z_{t,t}^t}\frac{\partial z_{e,t}^t}{\partial x_t^t}\frac{\partial x_t^t}{\partial z_t^{L-1}} + \frac{\partial L}{\partial z_{ct}^t}\frac{\partial z_{e,f}^t}{\partial x_t^t}\frac{\partial x_t^t}{\partial z_{ox}^{t-1}} \\ &= \delta_{f,x}^T W_f \odot f'\left(z_t^{L-1}\right) + \delta_{i,t}^T W_i \odot f'\left(z_t^{t-1}\right) \\ &+ \delta_{\varepsilon,t}^T W_\varepsilon \odot f'\left(z_t^{-1}\right) + \delta_{\alpha t}^T W_\rho \odot f'\left(z_t^{L-1}\right) \\ &= \left(\delta_{f,t}^T W_f + \delta_{i,t}^T W_i + \delta_{\delta,t}^T W_c + \delta_{a,t}^T W_s\right) \odot f'\left(z_t^{t-1}\right)\end{aligned}$$

$$\begin{aligned}\tfrac{\partial L}{\partial U_{e,t}} &= \tfrac{\partial L}{\partial z_{\partial,t}}\tfrac{\partial z_{a,t}}{\partial U_{o,t}} = \delta_{at} h_{t-1}^T \\ \tfrac{\partial L}{\partial U_{ft}} &= \tfrac{\partial L}{\partial z_{f,t}}\tfrac{\partial z_{ft}}{\partial U_{f,t}} = \delta_{ft} h_{t-1}^T \\ \tfrac{\partial L}{\partial U_{t,t}} &= \tfrac{\partial L}{\partial z_{t,t}}\tfrac{\partial z_{t,t}}{\partial U_{t,t}} = \delta_{i,t} \hbar_{t-1}^T \\ \tfrac{\partial L}{\partial U_{es}} &= \tfrac{\partial L}{\partial z_{t,t}}\tfrac{\partial z_{t,t}}{\partial U_{c,t}} = \delta_{c,t} h_{t-1}^T\end{aligned}$$

$$\frac{\partial L}{\partial U_0} = \sum_{j=1}^{t} \delta_{n,j} h_{j-1}^T$$

$$\frac{\partial L}{\partial U_f} = \sum_{j=1}^{f} \delta_{f,j} h_{j-1}^T$$

$$\frac{\partial L}{\partial U_1} = \sum_{j=1}^{l} \delta_{i,j} h_{j-1}^T$$

$$\frac{\partial L}{\partial U_e} = \sum_{j=1}^{t} \delta_{k,j} h_{j-1}^T$$

$$\begin{aligned}\tfrac{\partial L}{\partial b_{a,x}} &= \tfrac{\partial L}{\partial z_{ax}}\tfrac{\partial z_{aA}}{\partial b_{a,t}} = \delta_{at} \\ \tfrac{\partial L}{\partial b_{fz}} &= \tfrac{\partial L}{\partial z_{f,t}}\tfrac{\partial z_{f,t}}{\partial b_{f,t}} = \delta_{f,t} \\ \tfrac{\partial L}{\partial b_{i,d}} &= \tfrac{\partial L}{\partial z_{t,t}}\tfrac{\partial z_{b,t}}{\partial b_{b,t}} = \delta_t \\ \tfrac{\partial L}{\partial b_{c,t}} &= \tfrac{\partial L}{\partial z_{\ell,t}}\tfrac{\partial z_{b,t}}{\partial b_{c,t}} = \delta_t\end{aligned}$$

$$\frac{\partial L}{\partial b_0} = \sum_{j=1}^{t} \delta_{o_j j}$$

$$\frac{\partial L}{\partial b_i} = \sum_{j=1}^{f} \delta_{i,j}$$

$$\frac{\partial L}{\partial b_f} = \sum_{j=1}^{f} \delta_{fj}$$

$$\frac{\partial L}{\partial b_e} = \sum_{j=1}^{f} \delta_{k,j}$$

$$\begin{aligned}\frac{\partial L}{\partial W_o} &= \frac{\partial L}{\partial z_{o,t}}\frac{\partial z_{ot}}{\partial W_o} = \delta_{e,t} x_i^T \\ \frac{\partial L}{\partial W_f} &= \frac{\partial L}{\partial z_{f,i}}\frac{\partial z_{f,t}}{\partial W_f} = \delta_{f,x_i}^T \\ \frac{\partial L}{\partial W_i} &= \frac{\partial L}{\partial z_{i,t}}\frac{\partial z_{i,t}}{\partial W_i} = \delta_{i,t} x_i^T \\ \frac{\partial L}{\partial W_\rho} &= \frac{\partial L}{\partial z_{d,d}}\frac{\partial z_{l,t}}{\partial W_e} = \delta_{\dot{e},t} x_t^T\end{aligned}$$

The avoidance of LSTM on the disappearance of gradients is mainly based on $\frac{\partial c_1}{\partial c_1 - 1}$, $\frac{\partial h_1}{\partial h_2 - 1}$.
Due to the nature of the sigmoid function, most of the above two items are not 0 or 1, which greatly reduces the probability of gradient disappearance. (A value of 0 indicates that the information at the previous time has no effect on the current time).

3.5 Problem 6.5

- Calculate the feedforward function:

$$\begin{aligned}r_t &= \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right) \\ z_t &= \sigma\left(W_t x_t + U_s h_{t-1} + b_t\right) \\ h_t &= \tanh\left(W_h x_t + U_h\left(r_t \odot h_{t-1}\right) + b_B\right) \\ h_t &= \left(1 - z_t\right) \odot \hbar_t + z_t \odot h_{t-1} \\ y_t &= \operatorname{softmax}\left(V h_t + b_v\right)\end{aligned}$$

- Update the gradients:

$$\frac{\partial L_t}{\partial W_z} = \frac{\partial L_t}{\partial h_t}\frac{\partial h_t}{\partial W_z}$$
$$\frac{\partial L_t}{\partial h_t} = V\frac{\partial L_t}{\partial y_t}$$
$$\begin{aligned}\tfrac{\partial h_t}{\partial W_z} &= \textstyle\sum_{i=1}^t \left(\tfrac{\partial h_t}{\partial h_i}\left(\left(s_{t-1} - h_t\right) \odot z_t \odot \left(1 - z_t\right)\right) x_t^T\right) \\ &= \textstyle\sum_{i=1}^t \left(\prod_{j=i}^{t-1} \tfrac{\partial h_{j+1}}{\partial h_j}\right)\left(\left(s_{t-1} - h_t\right) \odot z_t \odot \left(1 - z_t\right)\right) x_t^T\right)\end{aligned}$$

$$\begin{aligned}\frac{\partial h_t}{\partial h_{t-1}} &= \frac{\partial h_t}{\partial \hat{h}_t}\frac{\partial \hat{h}_t}{\partial h_{t-1}} + \frac{\partial h_t}{\partial z_t}\frac{\partial z_t}{\partial h_{t-1}} + \frac{\partial^+ h_t}{\partial h_{t-1}} \\ &= \frac{\partial h_t}{\partial \hat{h}_t}\left(\frac{\partial \hat{h}_t}{\partial r_t}\frac{\partial r_t}{\partial h_{t-1}} + \frac{\partial^+ h_t}{\partial h_{t-1}}\right) + \frac{\partial h_t}{\partial z_t}\frac{\partial z_t}{\partial h_{t-1}} + \frac{\partial^+ h_t}{\partial h_{t-1}}\end{aligned}$$

We can easily find out that during the calculation, there is still a multiplication part, but due to the existence of the update gate, the nature of the sigmoid function, most of which are non-zero or 1, greatly reduces the probability of the disappearance of the gradient.