Machine Learning

SEU — 2024 Spring

Assignment 3

58122204 Xie xing *Instructor: Xu-Ying Liu*

§ Model Selection and Evaluation & Neural Networks §

Problem 1: Multi-layer Perception

(1) Given the network structure with one hidden layer and two neurons, we need to configure the weights and biases to model the XOR function $x_1 \oplus x_2$.

Neural Network Configuration:

- **First Hidden Neuron:**
  - Weights: $w_{11} = 1$, $w_{12} = 1$
  - Bias: $b_1 = -1$

  This gives the ReLU output:

$$\text{ReLU}(w_{11}x_1 + w_{12}x_2 + b_1) = \text{ReLU}(x_1 + x_2 - 1)$$

- **Second Hidden Neuron:**
  - Weights: $w_{21} = 1$, $w_{22} = 1$
  - Bias: $b_2 = -2$

  This gives the ReLU output:

$$\text{ReLU}(w_{21}x_1 + w_{22}x_2 + b_2) = \text{ReLU}(x_1 + x_2 - 2)$$

- **Output Neuron:**
  - Weights: $w_{01} = 1$, $w_{02} = -2$
  - Bias: $b_0 = 0$

  The output is computed as:

$$f(w_{01}a_1 + w_{02}a_2 + b_0)$$

Output Analysis:

- When $x_1 = x_2 = 0$: Both hidden neurons output 0, leading to an output of 0.
- When $x_1 = 1, x_2 = 0$ or $x_1 = 0, x_2 = 1$: The first hidden neuron outputs 1, the second outputs 0, and the final output is 1.
- When $x_1 = x_2 = 1$: The first hidden neuron outputs 1, the second outputs 0, leading to an output of 0.

Thus, the network successfully models the XOR function.

This assignment is due May 17, 2024 and the date of submission is May 8, 2024.

(2) Given the neural network structure with boolean inputs $x \in \{0,1\}^p$ and boolean output $y \in \{0,1\}$, we need to demonstrate that such a network can implement any arbitrary boolean function $h : \{0,1\}^p \to \{0,1\}$. We are allowed to use any finite number of neurons in the hidden layer, each followed by a threshold function.

Strategy

The strategy involves using a hidden layer that has enough neurons to represent every possible input combination. There are 2^p possible combinations for p boolean inputs.

Configuration

- **Hidden Layer:** Introduce 2^p neurons in the hidden layer. Each neuron n_i is dedicated to activating for exactly one input combination that maps to 1 in the function h.
- **Weights and Biases:** For each neuron n_i, configure the weights $\mathbf{w}_i$ and bias b_i such that:

$$n_i = \text{ReLU}(\mathbf{w}_i \cdot \mathbf{x} + b_i) \tag{1.1}$$

  where $\mathbf{w}_i$ has a large positive value for inputs that should trigger n_i and b_i is set to slightly less than the negative sum of the weights so that n_i activates only when its corresponding input pattern is present.
- **Output Layer:** The output neuron uses weights of 1 for all connections from the hidden neurons and a bias of -0.5 (assuming activation if the sum is positive). It computes the final output as:

$$y = f\left(\sum_{i=1}^{2^p} n_i - 0.5\right) \tag{1.2}$$

  where $f(v) = 1$ if $v > 0$ otherwise $f(v) = 0$.

Result

This configuration allows each neuron in the hidden layer to represent a specific pattern of the input that corresponds to an output of 1 according to the function h. The output neuron effectively sums these activations to determine if the input combination should result in a 1 or 0, thus being able to represent any arbitrary boolean function h.

Problem 2: Gradient explosion and gradient vanishing

From the meaning of the question, combined with the chain derivation rule, we can know:

$$\begin{aligned}\frac{\partial o}{\partial b_1} &= \frac{\partial o}{\partial h_3} \cdot \frac{\partial h_3}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial b_1} \\ &= w_4 \cdot w_3 \cdot \sigma'(w_3 h_2 + b_3) \cdot w_2 \cdot \sigma'(w_2 h_1 + b_2) \cdot \sigma'(w_1 x + b_1) \\ &= w_4 \cdot w_3 \cdot w_2 [1 - \sigma(w_3 h_2 + b_3)] \cdot \sigma(w_3 h_2 + b_3) \\ &\quad \cdot [1 - \sigma(w_2 h_1 + b_2)] \cdot \sigma(w_2 h_1 + b_2) \\ &\quad \cdot [1 - \sigma(w_1 x + b_1)] \cdot \sigma(w_1 x + b_1)\end{aligned} \tag{2.1}$$

(1) Bring in $w_1 = 100$, $w_2 = 150$, $w_3 = 200$, $w_4 = 200$, $b_1 = -300$, $b_2 = -75$, $b_3 = -100$, $b_4 = 10$,we can get:

$$\frac{\partial o}{\partial b_1} = 187500 \tag{2.2}$$

.

(2) Bring in $w_1 = 0.2$, $w_2 = 0.5$, $w_3 = 0.3$, $w_4 = 0.6$, $b_1 = 1$, $b_2 = 2$, $b_3 = 2$, $b_4 = 1$,we can get:

$$\frac{\partial o}{\partial b_1} = 0.000975758116082039 \tag{2.3}$$

.

Problem 3: CNN

(1) The parameters of the CNN are $v_1, v_2, v_3, w_1, w_2, w_3, w_4$, so the total number of parameters in this neural network is 7.

(2) $\frac{\partial L}{\partial w}$

(3) $\frac{\partial L}{\partial v_i}$

Problem 4: CNN

We need to calculate $\frac{\partial E}{\partial K}$.

$$\begin{aligned}\frac{\partial E}{\partial k} &= \frac{\partial E}{\partial o}\cdot\frac{\partial o}{\partial Y}\cdot\frac{\partial Y}{\partial k}\\ &= (z-\sigma(W\odot Y))\cdot\sigma'(W\odot Y)\cdot\begin{bmatrix}w_{11} & w_{12} & w_{13} & w_{14}\end{bmatrix}\\ &= (z-\sigma(W\odot Y))\cdot\sigma'(W\odot Y)\cdot W*X\\ &= (z-\sigma(W\odot Y))\,(1-\sigma(W\odot Y))\cdot W*X\end{aligned} \tag{4.1}$$

Problem 5: RNN: BPTT

We need to calculate $\frac{\partial E}{\partial V}$, $\frac{\partial E}{\partial W}$, $\frac{\partial E}{\partial U}$.

Noticed that $\frac{\partial E}{\partial U}=\Sigma_t\frac{\partial E_t}{\partial U}$(the same is true for the partial derivatives of V and W), just take the partial derivative of the loss function at each moment and add it up.

(1) calculate $\frac{\partial E}{\partial V}$

$$\frac{\partial E_t}{\partial V_{ij}} = tr((\frac{\partial E_t}{\partial z_t})^T\frac{\partial z_t}{\partial V_{ij}}) = tr((\hat{o_t}-o_t)^T\begin{bmatrix}0\\ \cdots\\ \frac{\partial z_t^{(i)}}{\partial V_{ij}}\\ \cdots\\ 0\end{bmatrix}) = (\hat{o_t^{(i)}}-o_t^{(i)})h_t^{(j)}$$

1. calculate $(\frac{\partial E_t}{\partial z_t})^T=(\hat{y_t}-y_t)^T$

$$\left(\frac{\partial E_t}{\partial z_t}\right)^T=\left(\frac{\partial E_t}{\partial \hat{o}_t}\right)^T\frac{\partial \hat{o}_t}{\partial z_t}$$

,

$$\begin{aligned}\left(\frac{\partial E_t}{\partial \hat{o}_t}\right)^T &= \begin{bmatrix}\frac{\partial(-o_t^{(1)}\log\hat{o}_t^{(1)})}{\partial\hat{o}_t^{(1)}} & \frac{\partial(-o_t^{(2)}\log\hat{o}_t^{(2)})}{\partial\hat{o}_t^{(2)}} & \cdots & \frac{\partial(-o_t^{(i)}\log\hat{o}_t^{(i)})}{\partial\hat{o}_t^{(i)}} & \cdots & \frac{\partial(-o_t^{(k)}\log\hat{o}_t^{(k)})}{\partial\hat{o}_t^{(k)}}\end{bmatrix}\\ &= \begin{bmatrix}-\frac{o_t^{(1)}}{\hat{o}_t^{(1)}} & -\frac{o_t^{(2)}}{\hat{o}_t^{(2)}} & \cdots & -\frac{o_t^{(i)}}{\hat{o}_t^{(i)}} & \cdots & -\frac{o_t^{(k)}}{\hat{o}_t^{(k)}}\end{bmatrix}\end{aligned}$$

,

$$\frac{\partial \hat{o}_t}{\partial z_t}=\begin{bmatrix}\frac{\partial \hat{o_t^{(1)}}}{\partial z_t^{(1)}} & \frac{\partial \hat{o_t^{(2)}}}{\partial z_t^{(1)}} & \cdots & \frac{\partial \hat{o_t^{(k)}}}{\partial z_t^{(1)}}\\ \frac{\partial o_t^{(1)}}{\partial z_t^{(2)}} & \frac{\partial o_t^{(2)}}{\partial z_t^{(2)}} & \cdots & \frac{\partial o_t^{(K)}}{\partial z_t^{(2)}}\\ \cdots & \cdots & \cdots & \cdots\\ \frac{\partial \hat{o_t^{(1)}}}{\partial z_t^{(k)}} & \frac{\partial \hat{o_t^{(2)}}}{\partial z_t^{(k)}} & \cdots & \frac{\partial \hat{o_t^{(k)}}}{\partial z_t^{(k)}}\end{bmatrix} \tag{5.1}$$

. According to the derivation formula of softmax, we know:

$$\frac{\partial o_t^{(i)}}{\partial z_t^{(i)}}=(1-o_t^{(i)})o_t^{(i)} \tag{5.2}$$

,

$$\frac{\partial \hat{o_t^{(j)}}}{\partial z_t^{(k)}}=-\hat{o_t^{(j)}}\hat{o_t^{(k)}}, j\neq k \tag{5.3}$$

So:

$$\begin{aligned}
&\left(\frac{\partial E_t}{\partial \hat{o}_t}\right)^T \frac{\partial \hat{o}_t}{\partial z_t} \\
&= \left[\begin{array}{ccccccc} \frac{-o_t^{(1)}}{o_t^{(1)}} & \frac{-o_t^{(2)}}{o_t^{(2)}} & \cdots & \frac{-o_t^{(i)}}{o_t^{(i)}} & \cdots & \frac{-o_t^{(k)}}{o_t^{(K)}} \end{array}\right] \left[\begin{array}{cccc} (1-o_t^{(1)})o_t^{(1)} & -o_t^{(1)}o_t^{(2)} & \cdots & -o_t^{(1)}o_t^{(k)} \\ -o_t^{(2)}o_t^{(1)} & (1-o_t^{(2)})o_t^{(2)} & \cdots & -o_t^{(2)}o_t^{(k)} \\ \cdots & \cdots & \ddots & \cdots \\ -o_t^{(k)}o_t^{(1)} & -o_t^{(k)}o_t^{(2)} & \cdots & (1-o_t^{(k)})o_t^{(k)} \end{array}\right] \\
&= \left[\hat{o}_t^{(1)} - o_t^{(1)} \quad \hat{o}_t^{(2)} - o_t^{(2)} \quad \cdots \quad \hat{o}_t^{(i)} - o_t^{(i)} \quad \cdots \quad \hat{o}_t^{(k)} - o_t^{(k)}\right] = (\hat{o}_t - o_t)^T
\end{aligned} \tag{5.4}$$

namely,

$$\frac{\partial E_t}{\partial z_t})^T = (\hat{y}_t - y_t)^T \tag{5.5}$$

2. Derivation of $\frac{\partial z_t}{\partial V_{ij}}$

$$\frac{\partial z_t}{\partial V_{ij}} = \left[\begin{array}{c} \frac{\partial z_t^{(1)}}{\partial V_{ij}} \\ \vdots \\ \frac{\partial z_t^{(i)}}{\partial V_{ij}} \\ \vdots \\ \frac{\partial z_t^{(k)}}{\partial V_{ij}} \end{array}\right]$$

because $z_t^{(i)} = \Sigma_{l=1}^m V_{il} h_t^{(l)}$, we can get $z_t^{(i)} = \Sigma_{l=1}^m V_{il} h_t^{(l)}$ and $\frac{\partial z_t^{(l)}}{\partial V_{ij}} = 0, l \neq i$, so:

$$\frac{\partial z_t}{\partial V_{ij}} = \left[\begin{array}{c} 0 \\ \vdots \\ \frac{\partial z_t^{(i)}}{\partial V_{ij}} \\ \vdots \\ 0 \end{array}\right]$$

Finally, multiply the $(\hat{o_t} - o_t)^T$ and $\left[\begin{array}{c} 0 \\ \vdots \\ \frac{\partial z_t^{(i)}}{\partial V_{ij}} \\ \vdots \\ 0 \end{array}\right]$ matrices (vectors) and take the sum of the diagonal elements to get the final result. At the same time, according to the relevant knowledge of matrix outer product, we can get:

$$\frac{\partial E_t}{\partial V} = (\hat{o}_t - o_t) \otimes (h_t)^T \tag{5.6}$$

From the above derivation we can get:

$$\frac{\partial E}{\partial V} = \Sigma_t (\hat{o}_t - o_t) \otimes (h_t)^T \tag{5.7}$$

(2) calculate $\frac{\partial E}{\partial W}$

Because W is shared by each time step, the changes in W caused by the time steps before t all contribute to E_t, so the entire sequence can be calculated periodically :

$$\frac{\partial E_t}{\partial W} = \sum_{k=0}^{t} \frac{\partial s_k}{\partial W} \frac{\partial E_t}{\partial s_k} \tag{5.8}$$

Here, due to the derivation algorithm of vectors to matrices, there is:

$$\frac{\partial E_t}{\partial W_{ij}} = \sum_{k=0}^{t} \mathrm{tr}\left(\left(\frac{\partial E_t}{\partial s_k}\right)^T \frac{\partial s_k}{\partial W_{ij}}\right) = \sum_{k=0}^{t} \mathrm{tr}\left((\delta_k)^T \frac{\partial s_k}{\partial W_{ij}}\right), \quad \text{where } \delta_k = \frac{\partial E_t}{\partial s_k}$$

1. Derivation of δ

The dependencies of each variable follow:

$$s_k \to h_k \to s_{k+1} \to \cdots \to s_t \to h_t \to E_t$$

The chain rule is:

$$\delta_k = \frac{\partial h_k}{\partial s_k}\frac{\partial s_{k+1}}{\partial h_k}\frac{\partial E_t}{\partial s_{k+1}}$$

Where $\frac{\partial h_k}{\partial s_k}$ is defined as:

$$\frac{\partial h_k}{\partial s_k} = \begin{bmatrix} \frac{\partial h_k^{(1)}}{\partial s_k^{(1)}} & \frac{\partial h_k^{(2)}}{\partial s_k^{(1)}} & \cdots & \frac{\partial h_k^{(m)}}{\partial s_k^{(1)}} \\ \frac{\partial h_k^{(1)}}{\partial s_k^{(2)}} & \frac{\partial h_k^{(2)}}{\partial s_k^{(2)}} & \cdots & \frac{\partial h_k^{(m)}}{\partial s_k^{(2)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_k^{(1)}}{\partial s_k^{(m)}} & \frac{\partial h_k^{(2)}}{\partial s_k^{(m)}} & \cdots & \frac{\partial h_k^{(m)}}{\partial s_k^{(m)}} \end{bmatrix}$$

It is easy to get from the derivation of tanh function:

$$\frac{\partial h_k^{(i)}}{\partial s_k^{(i)}} = 1 - (h_k^{(i)})^2, \quad \frac{\partial h_k^{(a)}}{\partial s_k^{(b)}} = 0, a \neq b \tag{5.9}$$

, So:

$$\frac{\partial h_k}{\partial s_k} = \begin{bmatrix} 1-(h_k^{(1)})^2 & 0 & \cdots & 0 \\ 0 & 1-(h_k^{(2)})^2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1-(h_k^{(m)})^2 \end{bmatrix} = diag(1 - h_k \odot h_k) \tag{5.10}$$

It is easy to see that :

$$\frac{\partial s_{k+1}}{\partial h_k} = W^T, \frac{\partial E_t}{\partial s_{k+1}} = \delta_{k+1} \tag{5.11}$$

Thus,

$$\delta_k = \frac{\partial h_k}{\partial s_k}\frac{\partial s_{k+1}}{\partial h_k}\frac{\partial E_t}{\partial s_{k+1}} = diag(1 - h_k \odot h_k)U^T\delta_{k+1} = (U^T\delta_{k+1}) \odot (1 - h_k \odot h_k) \tag{5.12}$$

Next, calculate δ_t, which is easily obtained by the chain rule:

$$\delta_t = \frac{\partial h_t}{\partial s_t}\frac{\partial z_t}{\partial h_t}\frac{\partial E_t}{\partial z_t} \tag{5.13}$$

It is easy to see that

$$\frac{\partial z_t}{\partial h_t} = V^T \tag{5.14}$$

,

Already asked for $(\frac{\partial E_t}{\partial z_t})^T = (\hat{o_t} - o_t)^T$, we know $\frac{\partial E_t}{\partial z_t} = \hat{o_t} - o_t$
Which leads to:

$$\begin{aligned} \delta_t = \frac{\partial h_t}{\partial s_t}\frac{\partial z_t}{\partial h_t}\frac{\partial E_t}{\partial z_t} = diag(1 - h_t \odot h_t)V^T(\hat{o_t} - o_t) = (V^T(\hat{o_t} - o_t)) \\ \odot (1 - h_t \odot h_t) \end{aligned} \tag{5.15}$$

To sum up, we get the recurrence relation about δ:

$$\begin{cases} \delta_t = (V^T(\hat{o_t} - o_t)) \odot (1 - h_t \odot h_t) \\ \delta_k = (U^T\delta_{k+1}) \odot (1 - h_k \odot h_k) \end{cases} \tag{5.16}$$

Starting from δ_t, we can push forward every δ.

2. calculate $\frac{\partial s_k}{\partial W_{ij}}$

For $\frac{\partial s_k}{\partial W_{ij}}$, because $s_k^{(i)} = \Sigma_{l=1}^{m} W_{il} h_{k-1}^{(l)}$,
To sum up, we get:

$$\frac{\partial E_t}{\partial W_{ij}} = \Sigma_{k=0}^{t} \delta_k^{(i)} h_{k-1}^{(j)} \tag{5.17}$$

$$\frac{\partial E_t}{\partial W} = \Sigma_{k=0}^{t} \delta_k \otimes (h_{k-1})^T \tag{5.18}$$

Final Results:

$$\frac{\partial E}{\partial W} = \Sigma_{t=0}^{T} \Sigma_{k=0}^{t} \delta_k \otimes (h_{k-1})^T \tag{5.19}$$

In order not to lose rigor, define $h_{-1}$is a vector of all 0.

(3) calculate $\frac{\partial E}{\partial U}$

Same as calculating $\frac{\partial E}{\partial W}$, we get:

$$\frac{\partial E_t}{\partial U} = \Sigma_{k=0}^{t} \delta_k \otimes (x_k)^T \tag{5.20}$$

$$\frac{\partial E}{\partial U} = \Sigma_{t=0}^{T} \Sigma_{k=0}^{t} \delta_k \otimes (x_k)^T \tag{5.21}$$