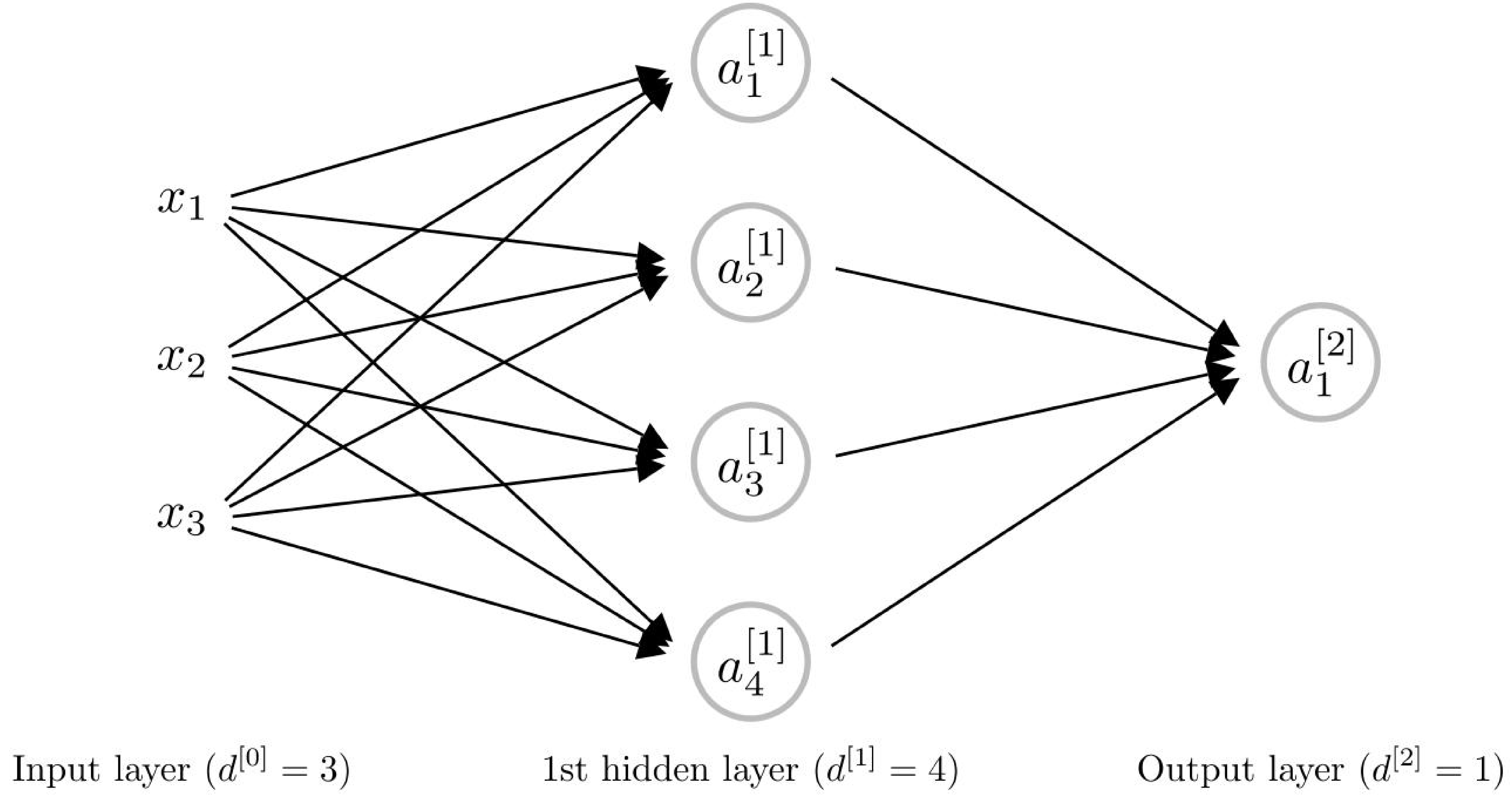
2-2 Neural Network with One Hidden Layer II

Zhonglei Wang

WISE and SOE, XMU, 2025

Contents

Recall


x_1
x_2
x_3
$a_1^{[1]}$
$a_2^{[1]}$
$a_3^{[1]}$
$a_4^{[1]}$
$a_1^{[2]}$
Input layer ($d^{[0]} = 3$)
1st hidden layer ($d^{[1]} = 4$)
Output layer ($d^{[2]} = 1$)

Recall

1. Recall that
   - L : number of layers in the neural network
   - $d^{[l]}$: number of neurons in the lth layer $\quad (l = 0, \ldots, L)$
   - $\boldsymbol{a}^{[l]} = (a_1^{[l]}, \ldots, a_{d^{[l]}}^{[l]})^{\mathrm{T}} \in \mathbb{R}^{d^{[l]} \times 1}$
   - $\boldsymbol{W}^{[l]} = (\boldsymbol{w}_1^{[l]}, \ldots, \boldsymbol{w}_{d^{[l]}}^{[l]})^{\mathrm{T}} \in \mathbb{R}^{d^{[l]} \times d^{[l-1]}}$
   - $\boldsymbol{b}^{[l]} = (b_1^{[l]}, \ldots, b_{d^{[l]}}^{[l]})^{\mathrm{T}} \in \mathbb{R}^{d^{[l]} \times 1}$

Recall

1. If we have only one training example $(\boldsymbol{x}, y)$, forward propagation is

$$
\begin{aligned}
\boldsymbol{z}^{[1]} &= \boldsymbol{b}^{[1]} + \boldsymbol{W}^{[1]}\boldsymbol{x} \\
\boldsymbol{a}^{[1]} &= \sigma(\boldsymbol{z}^{[1]}) \\
z^{[2]} &= b^{[2]} + \boldsymbol{W}^{[2]}\boldsymbol{a}^{[1]} \\
a^{[2]} &= \sigma(z^{[2]})
\end{aligned}
$$

2. Cost function

$$\mathcal{J} = \mathcal{L} = -\left\{y \log a^{[2]} + (1-y)\log\left(1 - a^{[2]}\right)\right\}$$

Forward propagation

1. Suppose we have n training examples $\{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, n\}$
2. Denote $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^{\mathrm{T}} \in \mathbb{R}^{n \times d}$
3. Forward propagation (using vectorization)

$$\boldsymbol{Z}^{[1]} = (\boldsymbol{b}^{[1]})^{\mathrm{T}} + \boldsymbol{X}(\boldsymbol{W}^{[1]})^{\mathrm{T}} \in \mathbb{R}^{n \times d^{[1]}}$$

$$\boldsymbol{A}^{[1]} = \sigma(\boldsymbol{Z}^{[1]}) \in \mathbb{R}^{n \times d^{[1]}}$$

$$\boldsymbol{Z}^{[2]} = b^{[2]} + \boldsymbol{A}^{[1]}(\boldsymbol{W}^{[2]})^{\mathrm{T}} \in \mathbb{R}^{n \times d^{[2]}}$$

$$\boldsymbol{A}^{[2]} = \sigma(\boldsymbol{Z}^{[2]}) \in \mathbb{R}^{n \times d^{[2]}}$$

- Broadcasting in python is implicitly used for $\boldsymbol{A}^{[1]}$ and $\boldsymbol{A}^{[2]}$

Forward propagation

1. For the example, we have $d^{[0]} = 3, d^{[1]} = 4, d^{[2]} = 1$
2. Thus,

$$\boldsymbol{Z}^{[1]} = (\boldsymbol{b}^{[1]})^{\mathrm{T}} + \boldsymbol{X}(\boldsymbol{W}^{[1]})^{\mathrm{T}} \in \mathbb{R}^{n \times 4}$$
$$\boldsymbol{A}^{[1]} = \sigma(\boldsymbol{Z}^{[1]}) \in \mathbb{R}^{n \times 4}$$
$$\boldsymbol{Z}^{[2]} = b^{[2]} + \boldsymbol{A}^{[1]}(\boldsymbol{W}^{[2]})^{\mathrm{T}} \in \mathbb{R}^{n \times 1}$$
$$\boldsymbol{A}^{[2]} = \sigma(\boldsymbol{Z}^{[2]}) \in \mathbb{R}^{n \times 1}$$

Forward propagation

1. Cost function

$$\mathcal{J} = -n^{-1} \sum_{i=1}^{n} \left\{ y_i \log a_i^{[2]} + (1 - y_i) \log \left(1 - a_i^{[2]} \right) \right\}$$

- $a_i^{[2]}$: "activated" value based on feature $\boldsymbol{x}_i$

Backpropagation

1. Denote $\mathrm{d}\cdot = \dfrac{\partial \mathcal{J}}{\partial \cdot}$

2. Recall: For only one training example$(\boldsymbol{x}, y)$, we have

$$\mathrm{d}b^{[2]} = a^{[2]} - y, \quad \mathrm{d}\boldsymbol{W}^{[2]} = (a^{[2]} - y)(\boldsymbol{a}^{[1]})^{\mathrm{T}}$$

$$\mathrm{d}\boldsymbol{b}^{[1]} = (a^{[2]} - y)\boldsymbol{D}(\boldsymbol{W}^{[2]})^{\mathrm{T}}, \quad \mathrm{d}\boldsymbol{W}^{[1]} = (a^{[2]} - y)\boldsymbol{D}(\boldsymbol{W}^{[2]})^{\mathrm{T}}\boldsymbol{x}^{\mathrm{T}}$$

   - $\boldsymbol{D} = \mathrm{diag}(\{a_j^{[1]}(1 - a_j^{[1]}) : j = 1, \ldots, d^{[1]}\})$

3. Note that

$$\mathrm{d}z^{[2]} = a^{[2]} - y, \quad \mathrm{d}\boldsymbol{z}^{[1]} = (a^{[2]} - y)\boldsymbol{D}(\boldsymbol{W}^{[2]})^{\mathrm{T}} = \mathrm{d}z^{[2]}\boldsymbol{D}(\boldsymbol{W}^{[2]})^{\mathrm{T}}$$

Backpropagation

1. Furthermore, notice that

$$\boldsymbol{D}(\boldsymbol{W}^{[2]})^{\mathrm{T}} = (\boldsymbol{W}^{[2]})^{\mathrm{T}} \circ \sigma'(\boldsymbol{z}^{[1]})$$

   - $\sigma'(\boldsymbol{z}^{[1]}) = (a_1^{[1]}(1 - a_1^{[1]}), \ldots, a_{d^{[1]}}^{[1]}(1 - a_{d^{[1]}}^{[1]}))^{\mathrm{T}}$
   - $\circ$: Hadamard production

2. We have

$$\mathrm{d}z^{[2]} = a^{[2]} - y, \quad \mathrm{d}b^{[2]} = \mathrm{d}z^{[2]}, \quad \mathrm{d}\boldsymbol{W}^{[2]} = \mathrm{d}z^{[2]}(\boldsymbol{a}^{[1]})^{\mathrm{T}}$$
$$\mathrm{d}\boldsymbol{z}^{[1]} = \mathrm{d}z^{[2]}\boldsymbol{D}(\boldsymbol{W}^{[2]})^{\mathrm{T}}, \quad \mathrm{d}\boldsymbol{b}^{[1]} = \mathrm{d}\boldsymbol{z}^{[1]}, \quad \mathrm{d}\boldsymbol{W}^{[1]} = \mathrm{d}\boldsymbol{z}^{[1]}\boldsymbol{x}^{\mathrm{T}}$$

Backpropagation

1. For general case, if we have n examples, the cose function is

$$\mathrm{d}\boldsymbol{\theta} = n^{-1} \sum_{i=1}^{n} \frac{\partial \mathcal{J}(y_i, a_i)}{\partial \boldsymbol{\theta}}$$

2. Thus, to obtain $\mathrm{d}\boldsymbol{\theta}$, we only need to take average of derivatives for each example

3. We use broadcasting

$$\sum_{k=1}^{K} a_k b_k = \boldsymbol{a}\boldsymbol{b}$$

- $\boldsymbol{a} = (a_1, \ldots, a_K)$ and $\boldsymbol{b} = (b_1, \ldots, b_K)^{\mathrm{T}}$
- a_k can be a scalar or a column vector
- b_k can be a scalar or a row vector

Backpropagation

1. Then, we have

$$\mathrm{d}\boldsymbol{Z}^{[2]} = n^{-1}(\boldsymbol{A}^{[2]} - \boldsymbol{Y}) \quad \mathrm{d}\boldsymbol{b}^{[2]} = (\mathrm{d}\boldsymbol{Z}^{[2]})^{\mathrm{T}}\mathbf{1} \quad \mathrm{d}\boldsymbol{W}^{[2]} = (\mathrm{d}\boldsymbol{Z}^{[2]})^{\mathrm{T}}\boldsymbol{A}^{[1]}$$

$$\mathrm{d}\boldsymbol{Z}^{[1]} = \mathrm{d}\boldsymbol{Z}^{[2]}\boldsymbol{W}^{[2]} \circ \sigma'(\boldsymbol{Z}^{[1]}) \quad \mathrm{d}\boldsymbol{b}^{[1]} = (\mathrm{d}\boldsymbol{Z}^{[1]})^{\mathrm{T}}\mathbf{1} \quad \mathrm{d}\boldsymbol{W}^{[1]} = (\mathrm{d}\boldsymbol{Z}^{[1]})^{\mathrm{T}}\boldsymbol{X}$$

- $\sigma'(z)$: derivative of $\sigma(z)$
- $\sigma'(\boldsymbol{Z}^{[1]})$: we calculate derivatives for each element in $\boldsymbol{Z}^{[1]}$

(Batch) gradient descent algorithm

Step 1. Randomly initialize $\boldsymbol{\theta}^{(0)}$

Step 2. Based on $\boldsymbol{\theta}^{(t)}$ obtain

$$\nabla \mathcal{J}\left(\boldsymbol{\theta}^{(t)}\right) = \frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^{(t)})$$

Step 3. Update parameter

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \alpha \nabla \mathcal{J}\left(\boldsymbol{\theta}^{(t)}\right)$$

Step 4. Go back to Step 2 until convergence