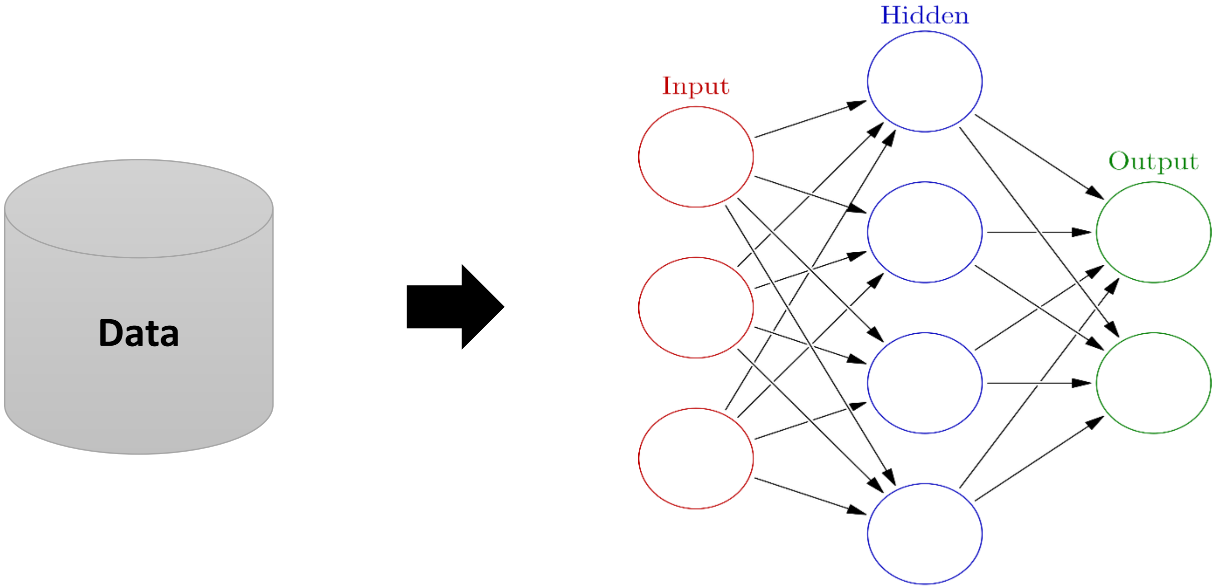
Error Backpropagation Algorithm

Data Intelligence and Learning (DIAL) Lab
Prof. Jongwuk Lee

Error Backpropagation Algorithm

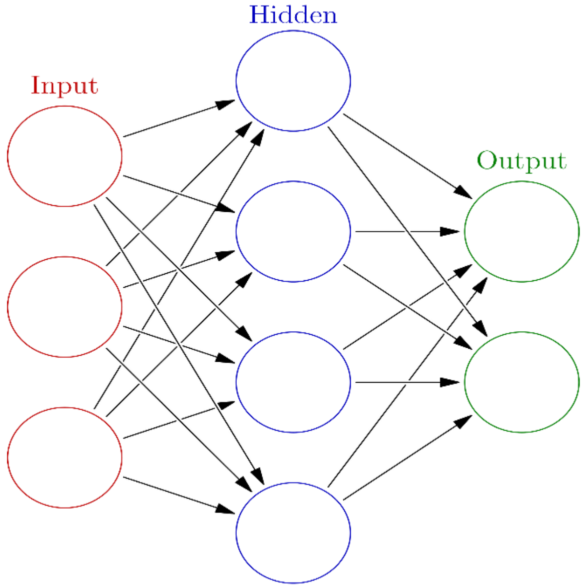
Forward Computation

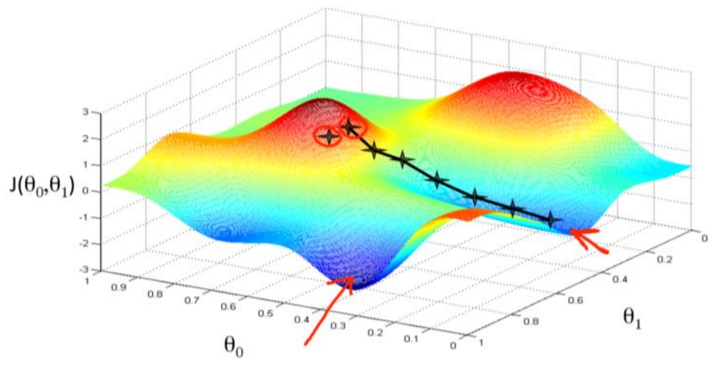
- Collect annotated data.
- Define a model and initialize weights randomly.
- Predict a value using the current model.
  - i.e., **forward propagation**
- Evaluate the prediction.



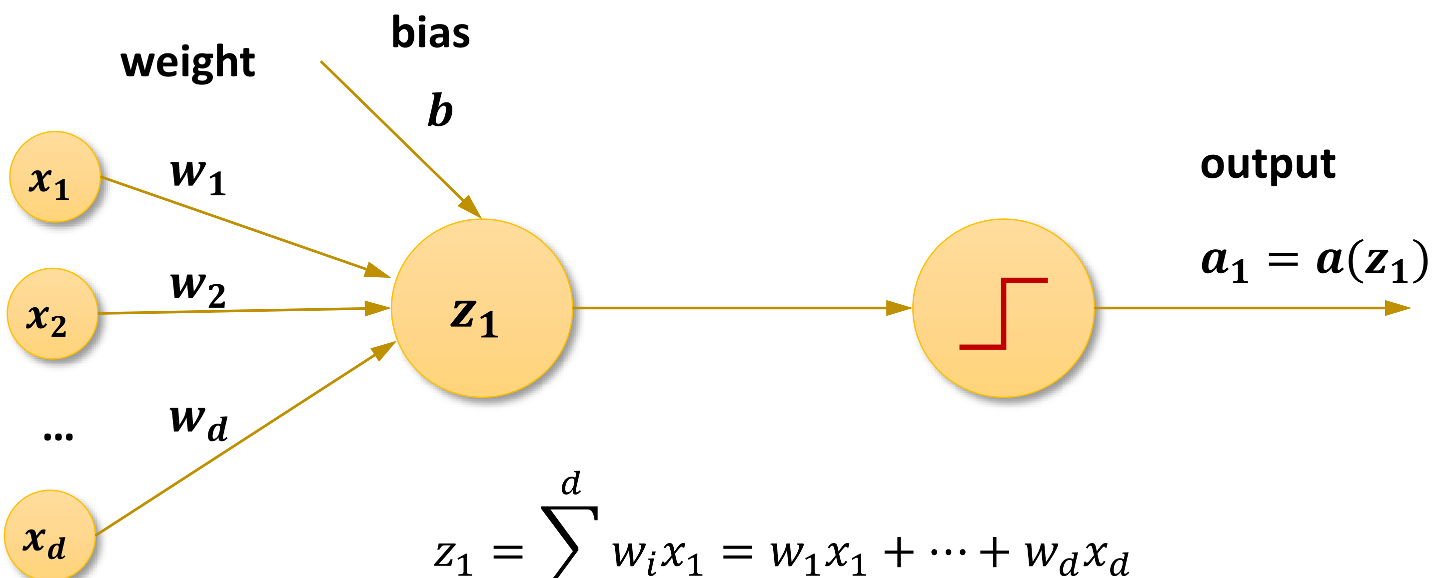
Backward Computation

- ➢ **Collect gradient data.**
- ➢ **Update the weights using gradients.**
  - i.e., **backward propagation**





Terminology

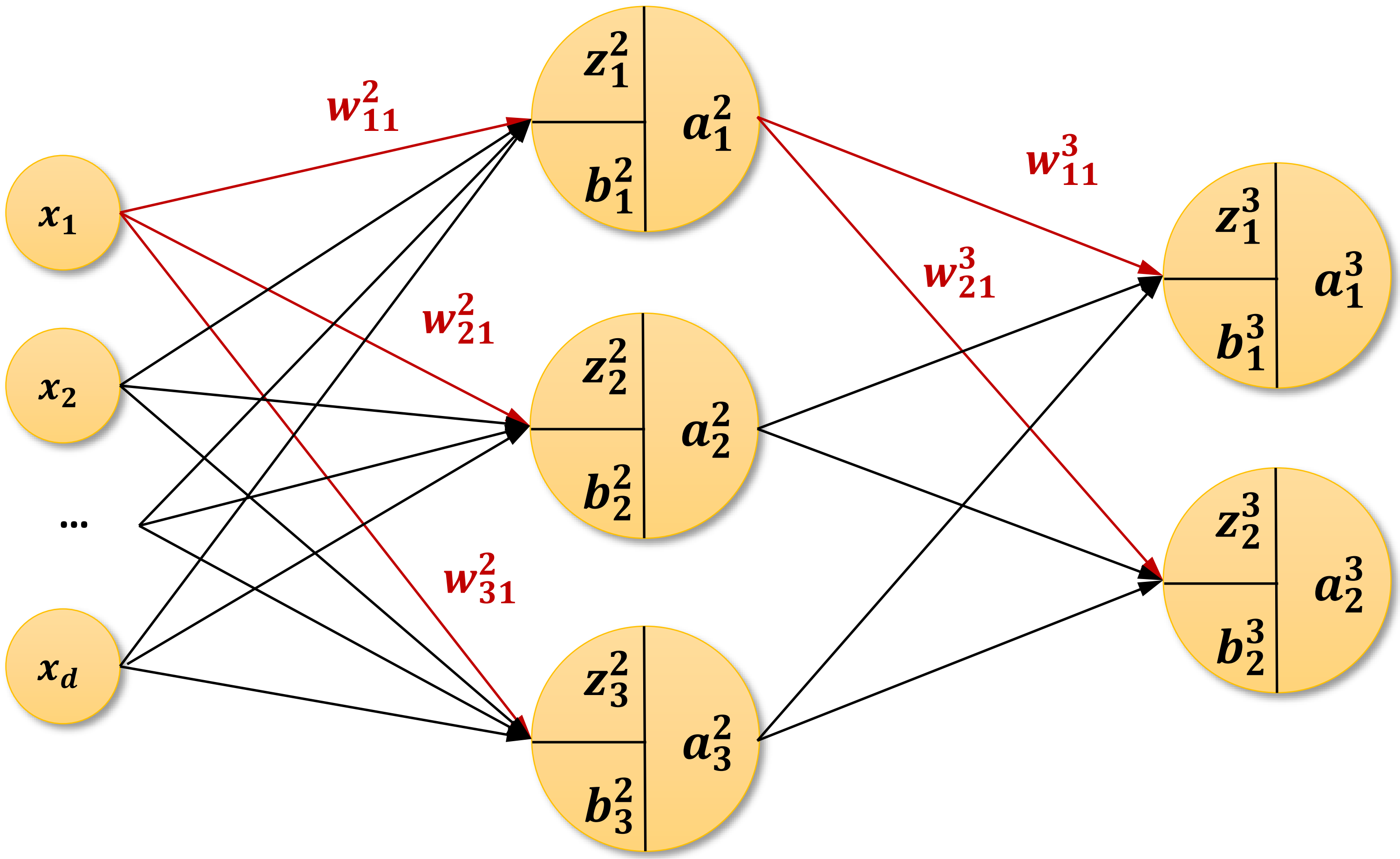


$$z_1 = \sum_{i=1}^{d} w_i x_1 = w_1 x_1 + \cdots + w_d x_d$$

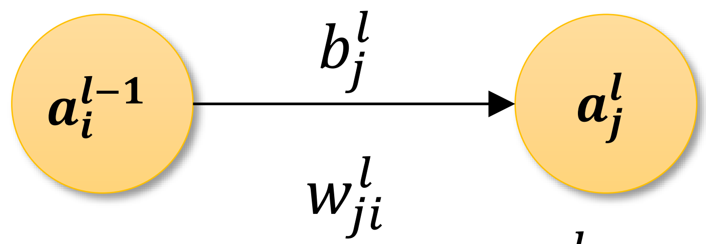
Input

$$a_1 = a(z_1)$$

Terminology



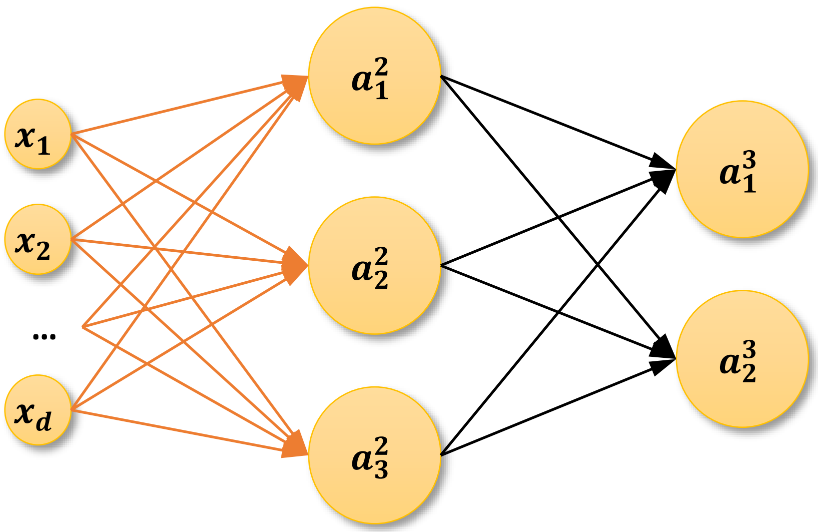
Terminology



$$a_j^l = a(z_j^l)$$

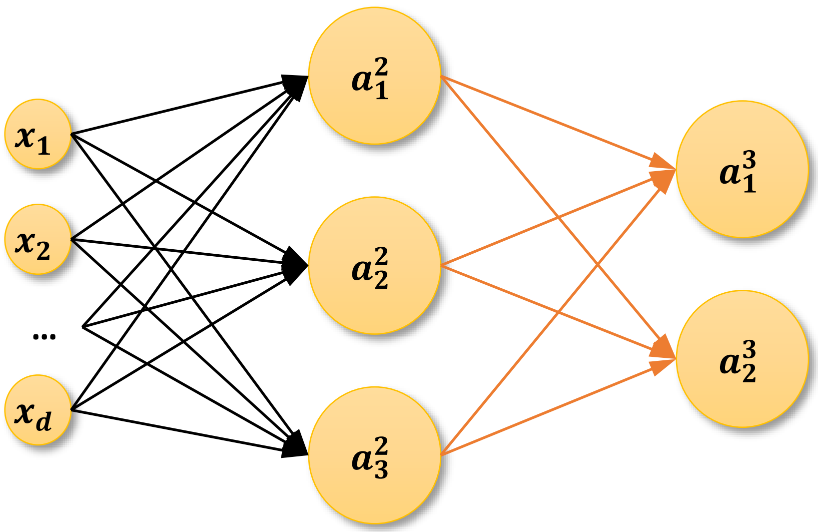
Notation	Meaning
x_i	The i-th input in the first layer
w_{ji}^l	The weight from i-th node in $(l-1)$-th layer to j-th node in the l-th layer
b_j^l	The bias from j-th node in the l-th layer
z_j^l	The intermediate output for j-th node in the l-th layer
a_j^l	The output for j-th node in the l-th layer

Example: Matrix Representation



$$\begin{bmatrix} z_1^2 \\ z_2^2 \\ z_3^2 \end{bmatrix} = \begin{bmatrix} w_{11}^2 & w_{12}^2 & \dots & w_{1d}^2 \\ w_{21}^2 & w_{22}^2 & \dots & w_{2d}^2 \\ w_{31}^2 & w_{31}^2 & \dots & w_{3d}^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} + \begin{bmatrix} b_1^2 \\ b_2^2 \\ b_3^2 \end{bmatrix}$$

Example: Matrix representation



$$\begin{bmatrix} z_1^2 \\ z_2^2 \\ z_3^2 \end{bmatrix} = \begin{bmatrix} w_{11}^2 & w_{12}^2 & \dots & w_{1d}^2 \\ w_{21}^2 & w_{22}^2 & \dots & w_{2d}^2 \\ w_{31}^2 & w_{31}^2 & \dots & w_{3d}^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} + \begin{bmatrix} b_1^2 \\ b_2^2 \\ b_3^2 \end{bmatrix}$$

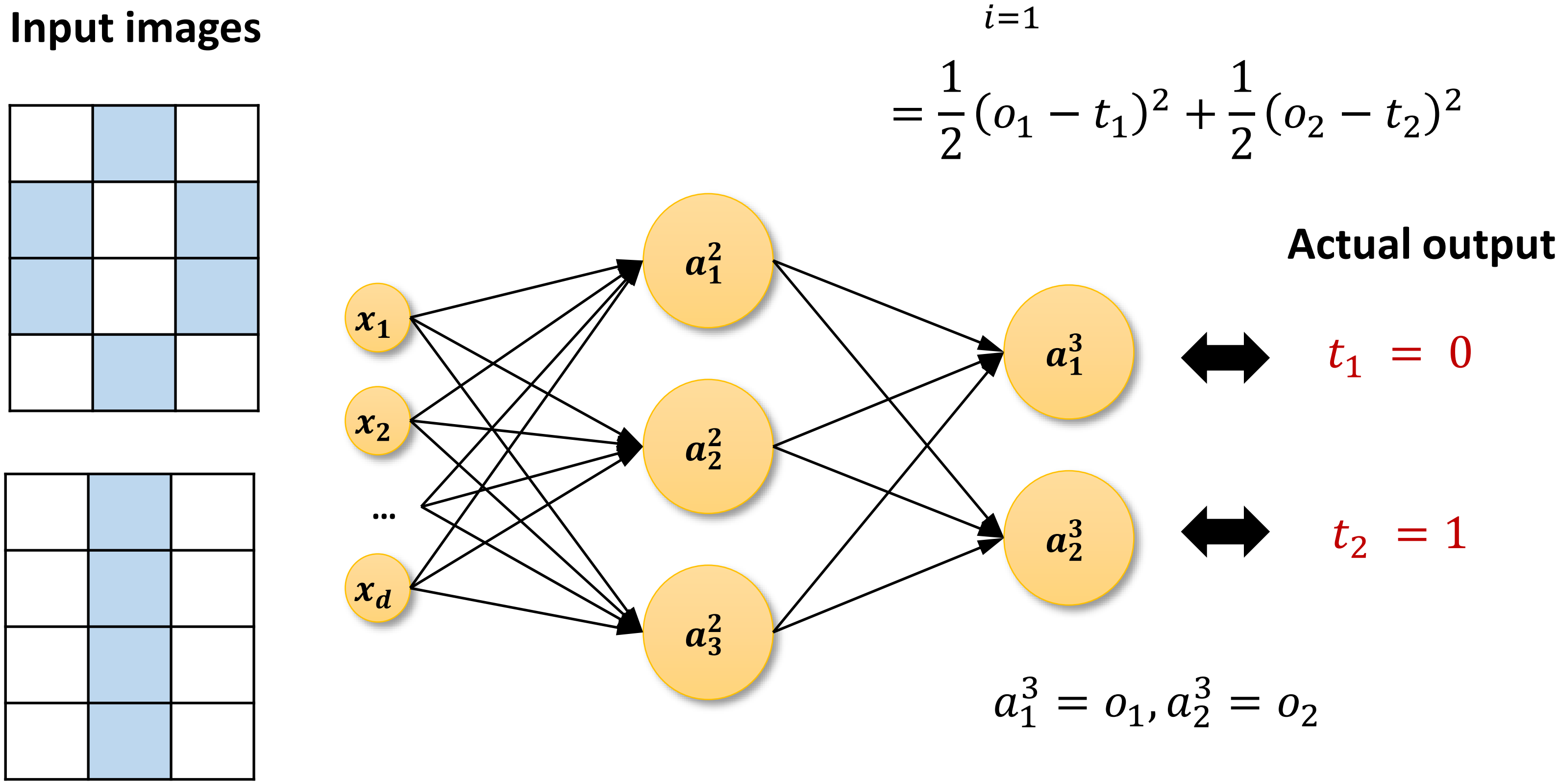
$$\begin{bmatrix} z_1^3 \\ z_2^3 \end{bmatrix} = \begin{bmatrix} w_{11}^3 & w_{12}^3 & w_{13}^3 \\ w_{21}^3 & w_{22}^3 & w_{23}^3 \end{bmatrix} \begin{bmatrix} a_1^2 \\ a_2^2 \\ a_3^2 \end{bmatrix} + \begin{bmatrix} b_1^3 \\ b_2^3 \end{bmatrix}$$

Example: Loss Function

- Also, possible to use the cross-entropy function.

$$\mathcal{L}(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(f\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2$$

$$= \frac{1}{2}(o_1 - t_1)^2 + \frac{1}{2}(o_2 - t_2)^2$$



$$a_1^3 = o_1, a_2^3 = o_2$$

Example: Loss Function

- **Given input-target pairs and output of NN**
  - $o_i = f(\mathbf{x}^{(i)})$: The output of a neural network
  - $t_i = y^{(i)}$: Target value of $\mathbf{x}^{(i)}$

$$\begin{array}{c} D_1 = (x_{11}, x_{12}, \dots, x_{1d}, t_{11}, t_{12}, \dots, t_{1m}) \\ D_2 = (x_{21}, x_{22}, \dots, x_{2d}, t_{21}, t_{22}, \dots, t_{2m}) \\ \vdots \\ D_n = (\underbrace{x_{n1}, x_{n2}, \dots, x_{nd}}_{\textbf{Inputs}}, \underbrace{t_{n1}, t_{n2}, \dots, t_{nm}}_{\textbf{Targets}}) \end{array}$$

$$\begin{array}{c} (o_{11}, o_{12}, \dots, o_{1m}) \\ (o_{21}, o_{22}, \dots, o_{2m}) \\ \vdots \\ \underbrace{(o_{n1}, o_{n2}, \dots, o_{nm})}_{\textbf{Outputs of NN}} \end{array}$$

Example: Loss Function

- **Minimizing the loss function**

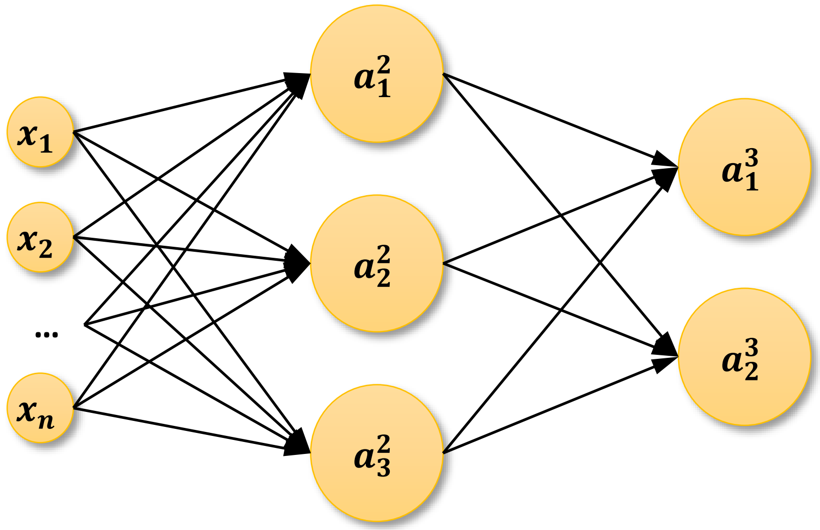
$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^{n} \mathcal{L}_i(\mathbf{w})$$

where

$$\mathcal{L}_i(\mathbf{w}) = \frac{1}{2}\sum_{k=1}^{m} (o_{ik} - t_{ik})^2$$

- **Now, we need to evaluate**

$$(\Delta w_{11}^2, \cdots, \Delta w_{11}^3, \cdots, \Delta b_1^2, \cdots, \Delta b_1^3, \cdots)$$
$$= \left(\frac{\partial \mathcal{L}}{\partial w_{11}^2}, \cdots, \frac{\partial \mathcal{L}}{\partial w_{11}^3}, \cdots, \frac{\partial \mathcal{L}}{\partial b_1^2}, \cdots, \frac{\partial \mathcal{L}}{\partial b_1^3}, \cdots\right)$$



Error Backpropagation

- **Computing the derivative is TOO complex.**

$$
\begin{aligned}
&(\Delta w_{11}^2, \cdots, \Delta w_{11}^3, \cdots, \Delta b_1^2, \cdots, \Delta b_1^3, \cdots)\\
&= \left(\frac{\partial \mathcal{L}}{\partial w_{11}^2}, \cdots, \frac{\partial \mathcal{L}}{\partial w_{11}^3}, \cdots, \frac{\partial \mathcal{L}}{\partial b_1^2}, \cdots, \frac{\partial \mathcal{L}}{\partial b_1^3}, \cdots\right)
\end{aligned}
$$

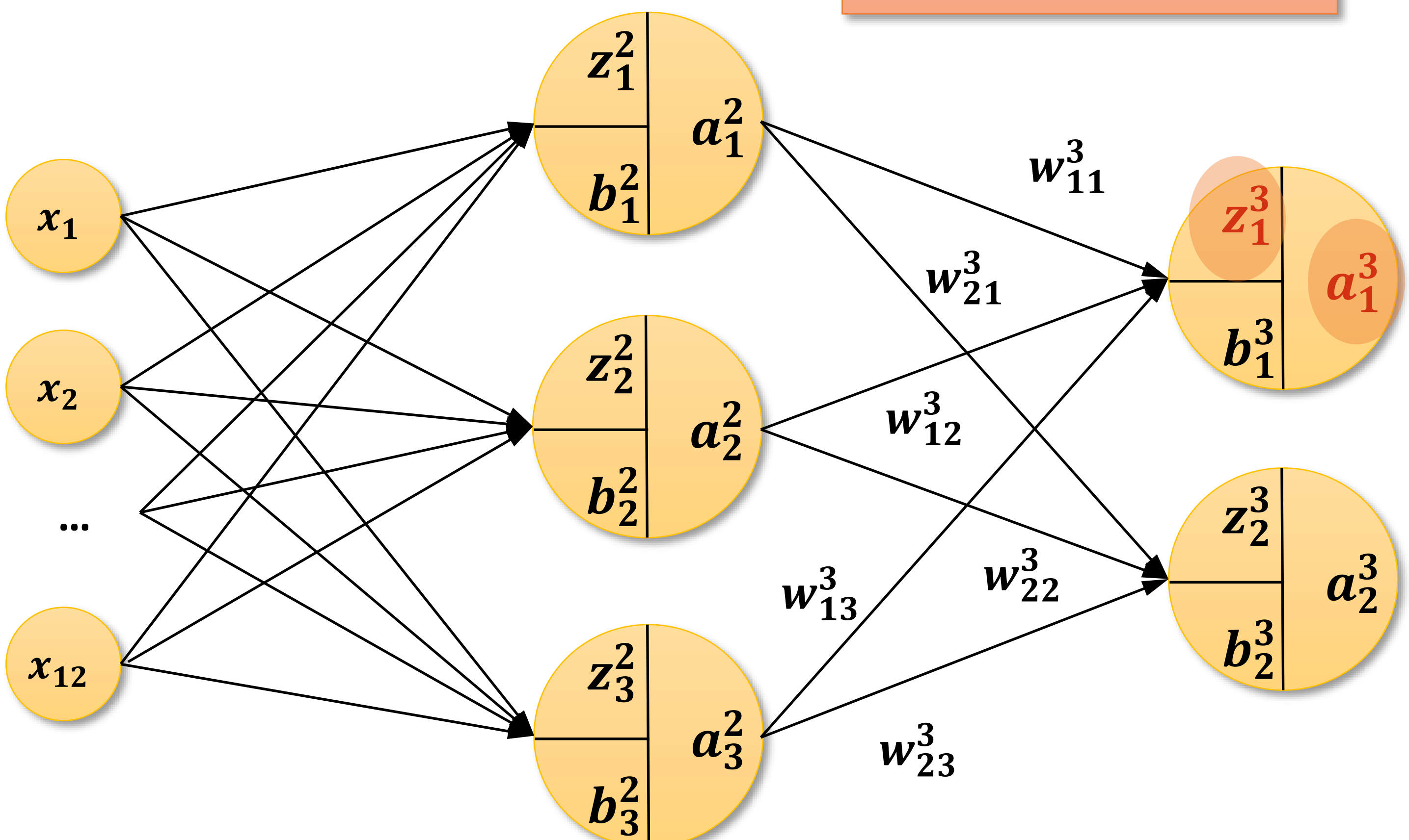
- **Backpropagation algorithm solves this problem!**
  - This is based on the **computational graph**.

Computing the Error δ_j^l

$$\delta_j^l = \frac{\partial \mathcal{L}}{\partial z_j^l} \quad (l = 2, 3, \dots)$$

$$a_1^3 = \sigma(z_1^3)$$

$$\delta_1^3 = \frac{\partial \mathcal{L}}{\partial z_1^3} = \frac{\partial a_1^3}{\partial z_1^3}\frac{\partial \mathcal{L}}{\partial a_1^3}$$

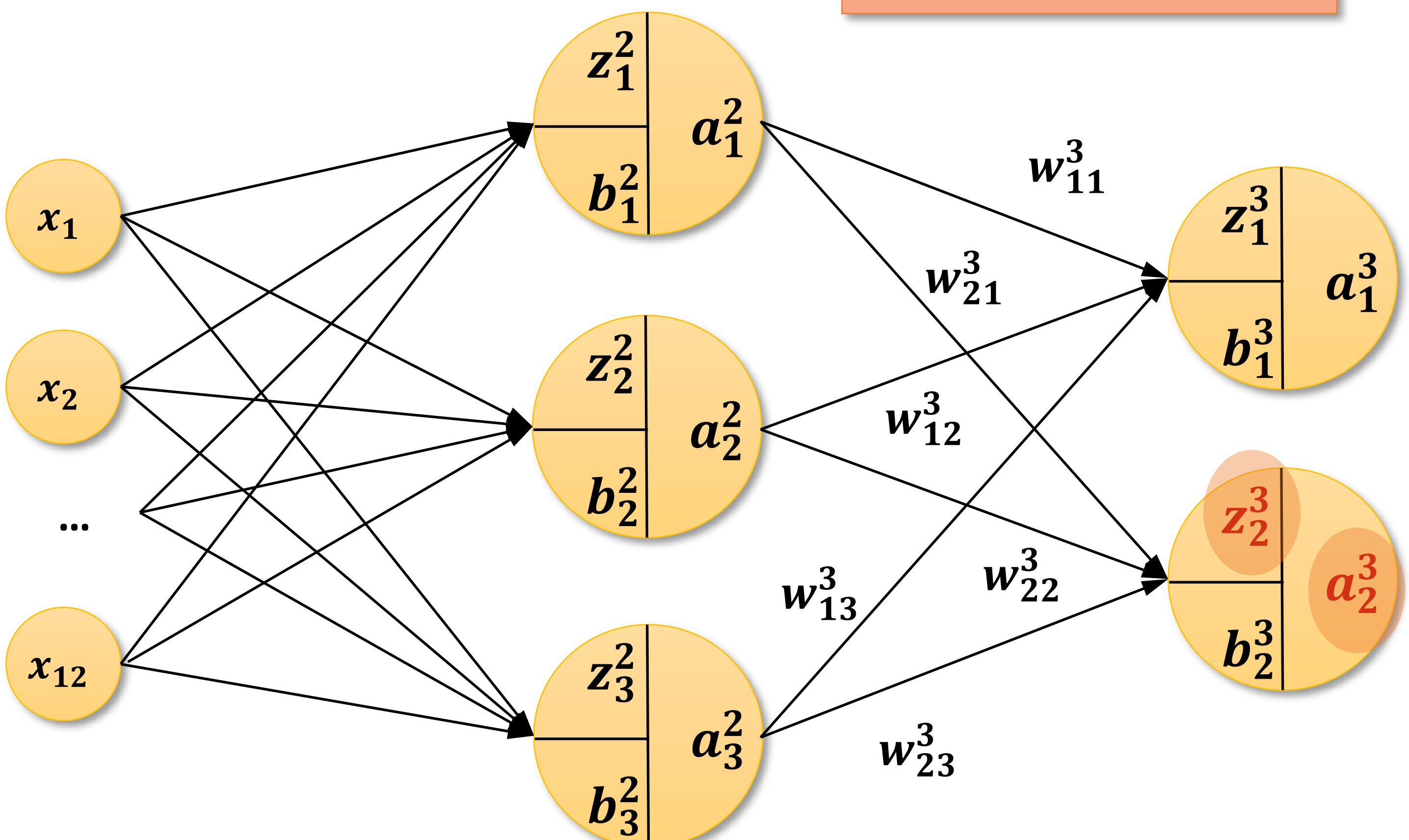


Computing the Error δ_j^l

$$\delta_j^l = \frac{\partial \mathcal{L}}{\partial z_j^l} \quad (l = 2, 3, \dots)$$

$$a_2^3 = \sigma(z_2^3)$$

$$\delta_2^3 = \frac{\partial \mathcal{L}}{\partial z_2^3} = \frac{\partial a_2^3}{\partial z_2^3}\frac{\partial \mathcal{L}}{\partial a_2^3}$$

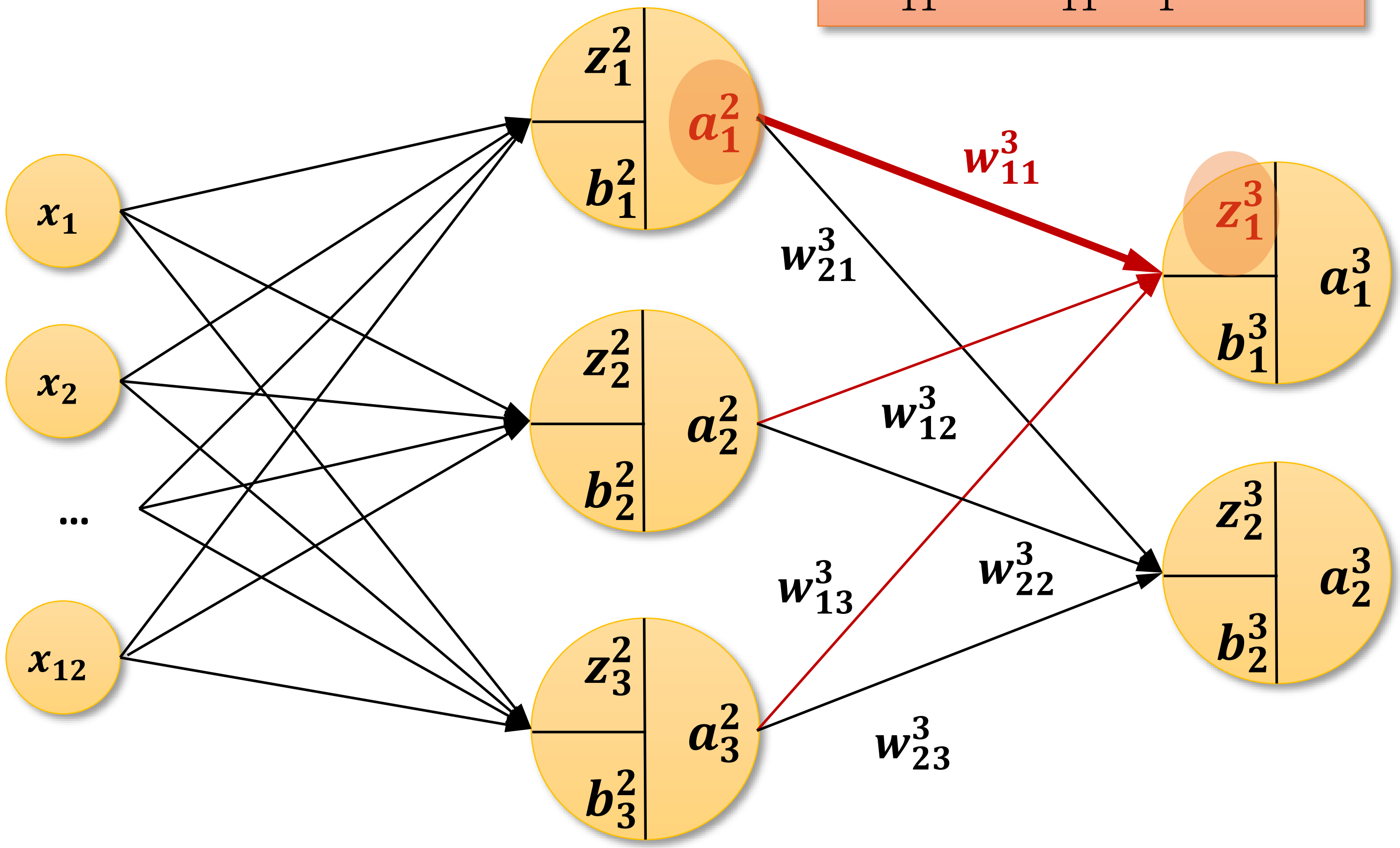


Computing the Error for w_{ji}^{l}

$$\delta_j^l = \frac{\partial \mathcal{L}}{\partial z_j^l} \quad (l = 2, 3, \dots)$$

$$z_1^3 = \sum_{i=1}^{3} w_{1i}^3 a_i^2 \qquad \delta_1^3 = \frac{\partial \mathcal{L}}{\partial z_1^3}$$

$$\frac{\partial \mathcal{L}}{\partial w_{11}^3} = \frac{\partial z_1^3}{\partial w_{11}^3} \frac{\partial \mathcal{L}}{\partial z_1^3} = a_1^2 \delta_1^3$$

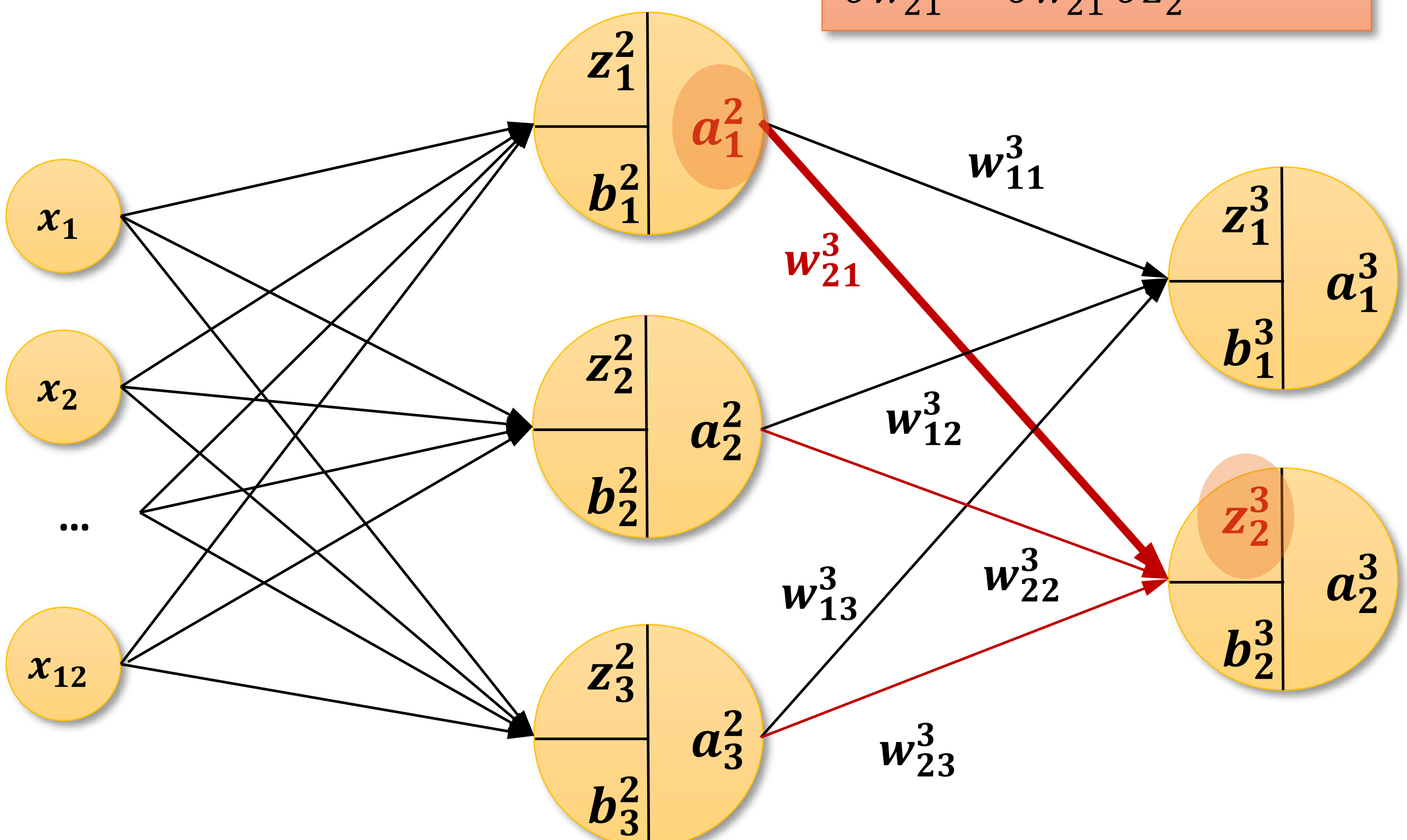


Computing the Error for w_{ji}^l

$$\delta_j^l = \frac{\partial \mathcal{L}}{\partial z_j^l} \quad (l = 2, 3, \dots)$$

$$z_1^3 = \sum_{i=1}^{3} w_{2i}^3 a_i^2 \qquad \delta_2^3 = \frac{\partial \mathcal{L}}{\partial z_2^3}$$

$$\frac{\partial \mathcal{L}}{\partial w_{21}^3} = \frac{\partial z_2^3}{\partial w_{21}^3} \frac{\partial \mathcal{L}}{\partial z_2^3} = a_1^2 \delta_2^3$$

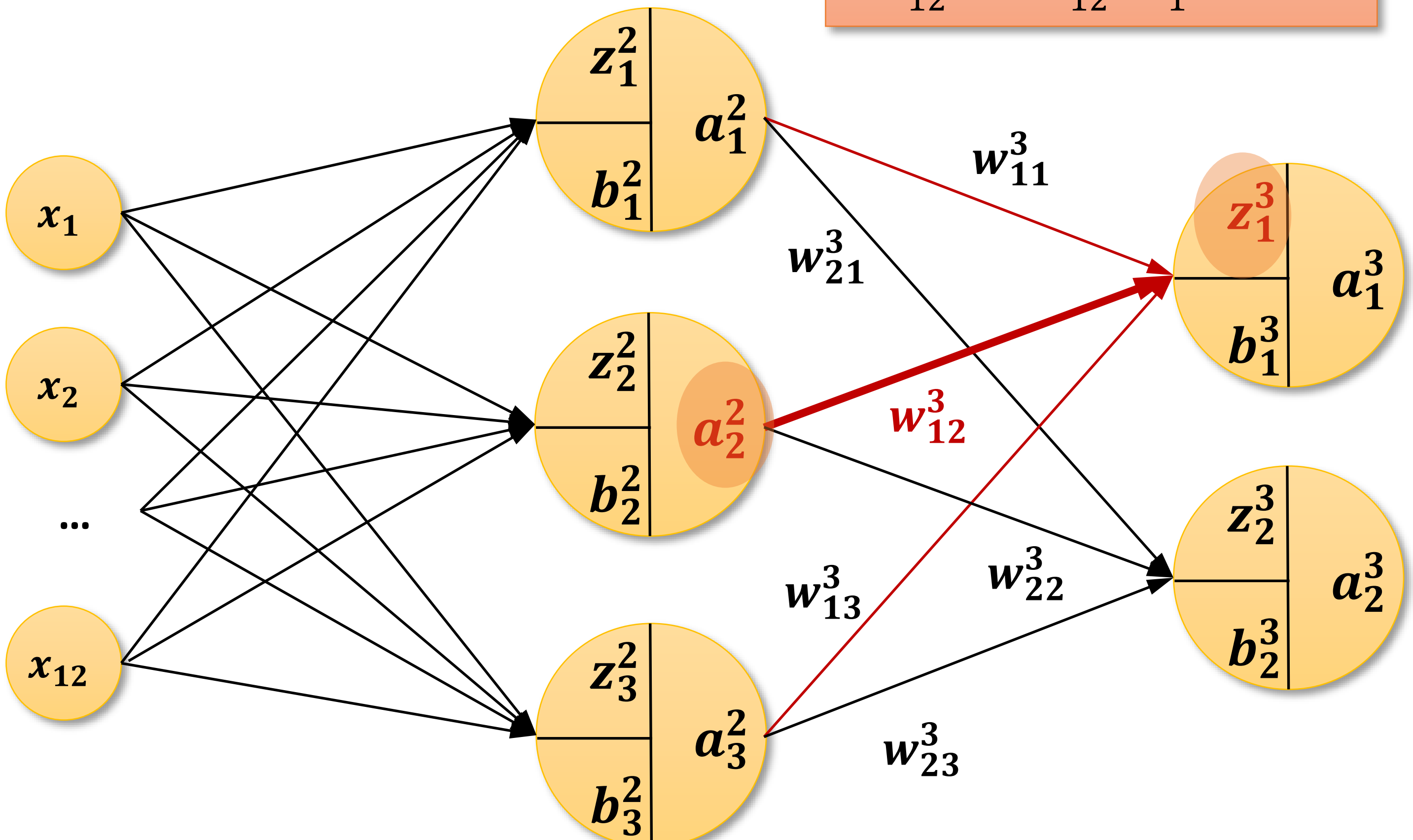


Computing the Error for w^l_{ji}

$$\delta^l_j = \frac{\partial \mathcal{L}}{\partial z^l_j} \quad (l = 2, 3, \ldots)$$

$$z^3_1 = \sum_{i=1}^{3} w^3_{1i} a^2_i \qquad \delta^3_1 = \frac{\partial \mathcal{L}}{\partial z^3_1}$$

$$\frac{\partial \mathcal{L}}{\partial w^3_{12}} = \frac{\partial z^3_1}{\partial w^3_{12}} \frac{\partial \mathcal{L}}{\partial z^3_1} = a^2_2 \delta^3_1$$

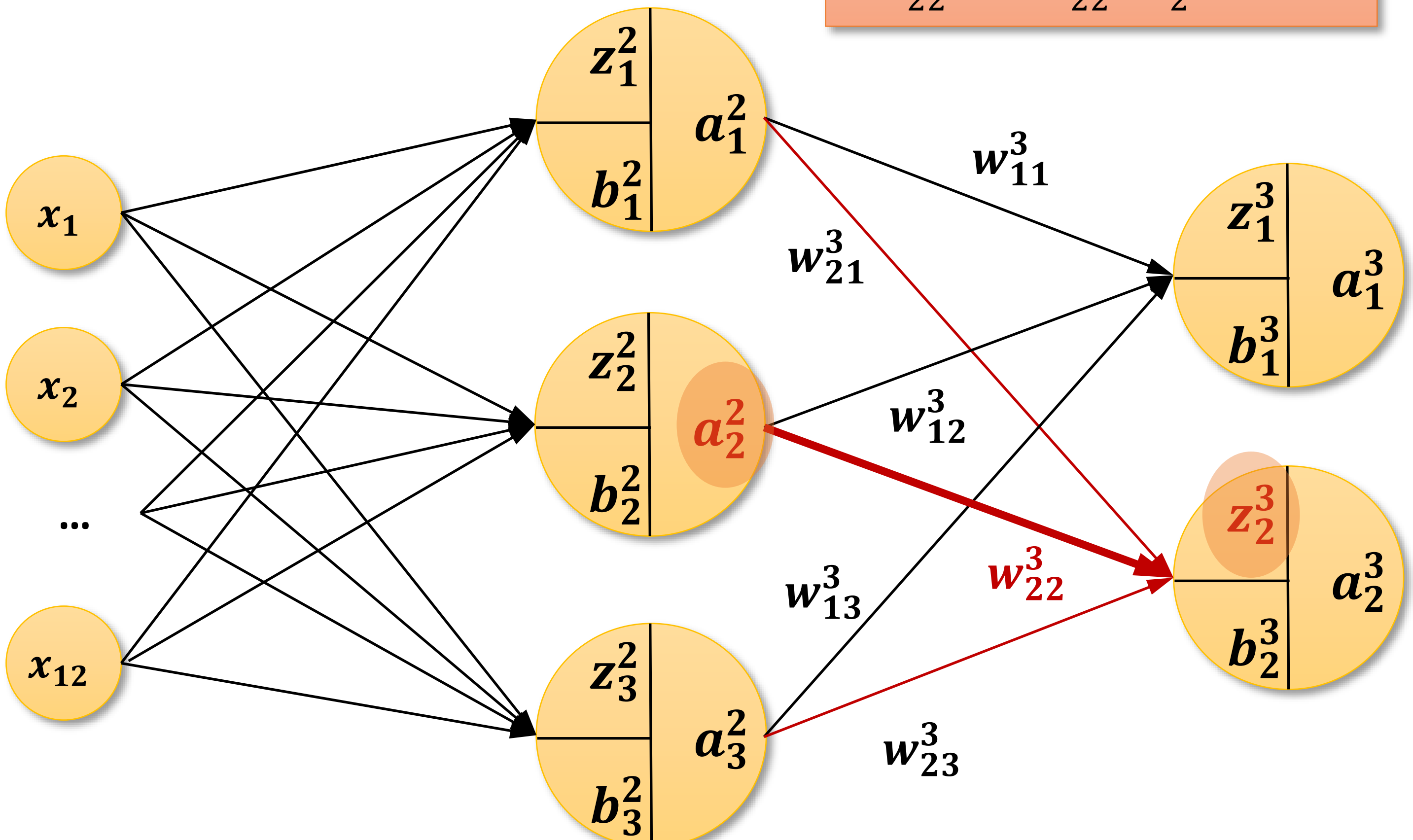


Computing the Error for w_{ji}^{l}

$$\delta_j^l = \frac{\partial \mathcal{L}}{\partial z_j^l} \quad (l = 2, 3, \dots)$$

$$z_1^3 = \sum_{i=1}^{3} w_{1i}^3 a_i^2 \qquad \delta_2^3 = \frac{\partial \mathcal{L}}{\partial z_2^3}$$

$$\frac{\partial \mathcal{L}}{\partial w_{22}^3} = \frac{\partial z_2^3}{\partial w_{22}^3} \frac{\partial \mathcal{L}}{\partial z_2^3} = a_2^2 \delta_2^3$$



Computing the Error for w_{ji}^{l} and b_{j}^{l}

- How to compute the error for w_{ji}^{l}

$$a_i^{l-1} \xrightarrow{w_{ji}^{l}} z_j^{l}$$

$$w_{ji}^{l} = \frac{\partial z_j^{l}}{\partial w_{ji}^{l}} \frac{\partial \mathcal{L}}{\partial z_j^{l}} = a_i^{l-1} \delta_j^{l}$$

- How to compute the error for b_{j}^{l}

$$1 \xrightarrow{b_{j}^{l}} z_j^{l}$$

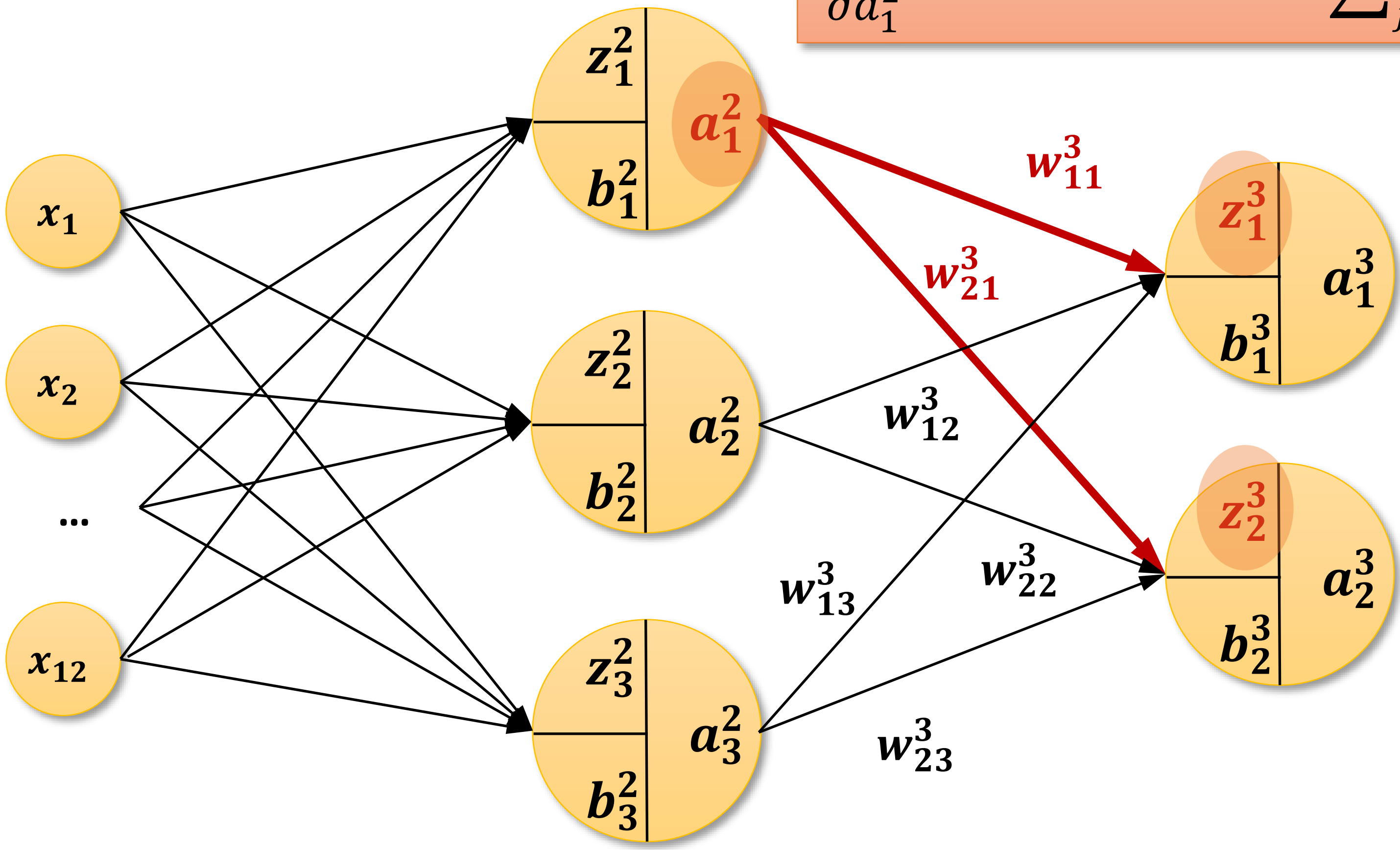
$$b_{j}^{l} = \frac{\partial z_j^{l}}{\partial b_{j}^{l}} \frac{\partial \mathcal{L}}{\partial z_j^{l}} = \delta_j^{l}$$

Computing the Error for a_j^{l-1}

$$\delta_j^l = \frac{\partial \mathcal{L}}{\partial z_j^l} \quad (l = 2, 3, \dots)$$

$$\frac{\partial \mathcal{L}}{\partial a_1^2} = \frac{\partial z_1^3}{\partial a_1^2}\frac{\partial \mathcal{L}}{\partial z_1^3} + \frac{\partial z_2^3}{\partial a_1^2}\frac{\partial \mathcal{L}}{\partial z_2^3}$$

$$\frac{\partial \mathcal{L}}{\partial a_1^2} = w_{11}^3\delta_1^3 + w_{21}^3\delta_2^3 = \sum_{j=1}^{2} w_{j1}^3\delta_j^3$$

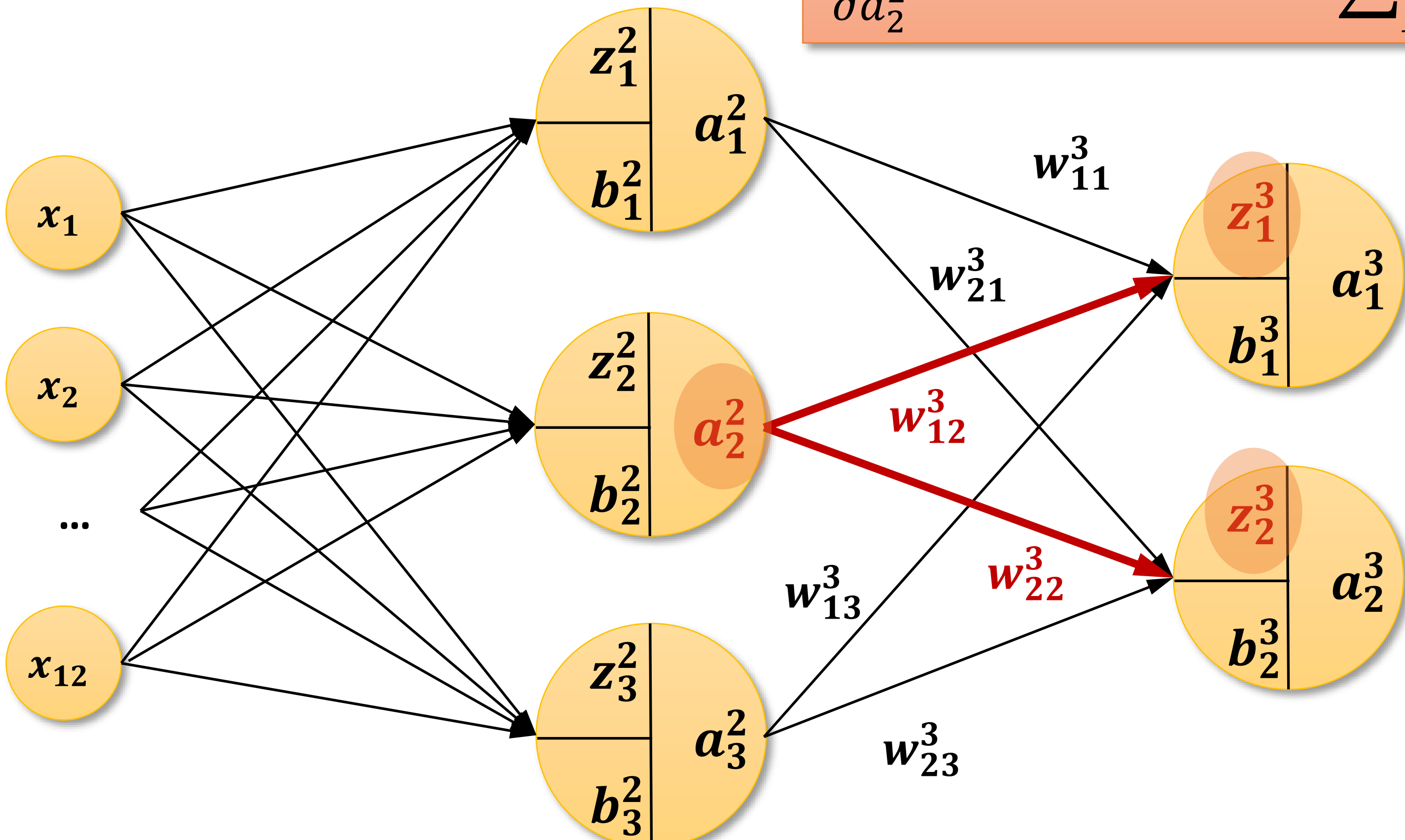


Computing the Error for a_j^{l-1}

$$\delta_j^l = \frac{\partial \mathcal{L}}{\partial z_j^l} \quad (l = 2, 3, \dots)$$

$$\frac{\partial \mathcal{L}}{\partial a_2^2} = \frac{\partial z_1^3}{\partial a_2^2}\frac{\partial \mathcal{L}}{\partial z_1^3} + \frac{\partial z_2^3}{\partial a_2^2}\frac{\partial \mathcal{L}}{\partial z_2^3}$$

$$\frac{\partial \mathcal{L}}{\partial a_2^2} = w_{12}^3\delta_1^3 + w_{22}^3\delta_2^3 = \sum_{j=1}^{2} w_{j2}^3\delta_j^3$$

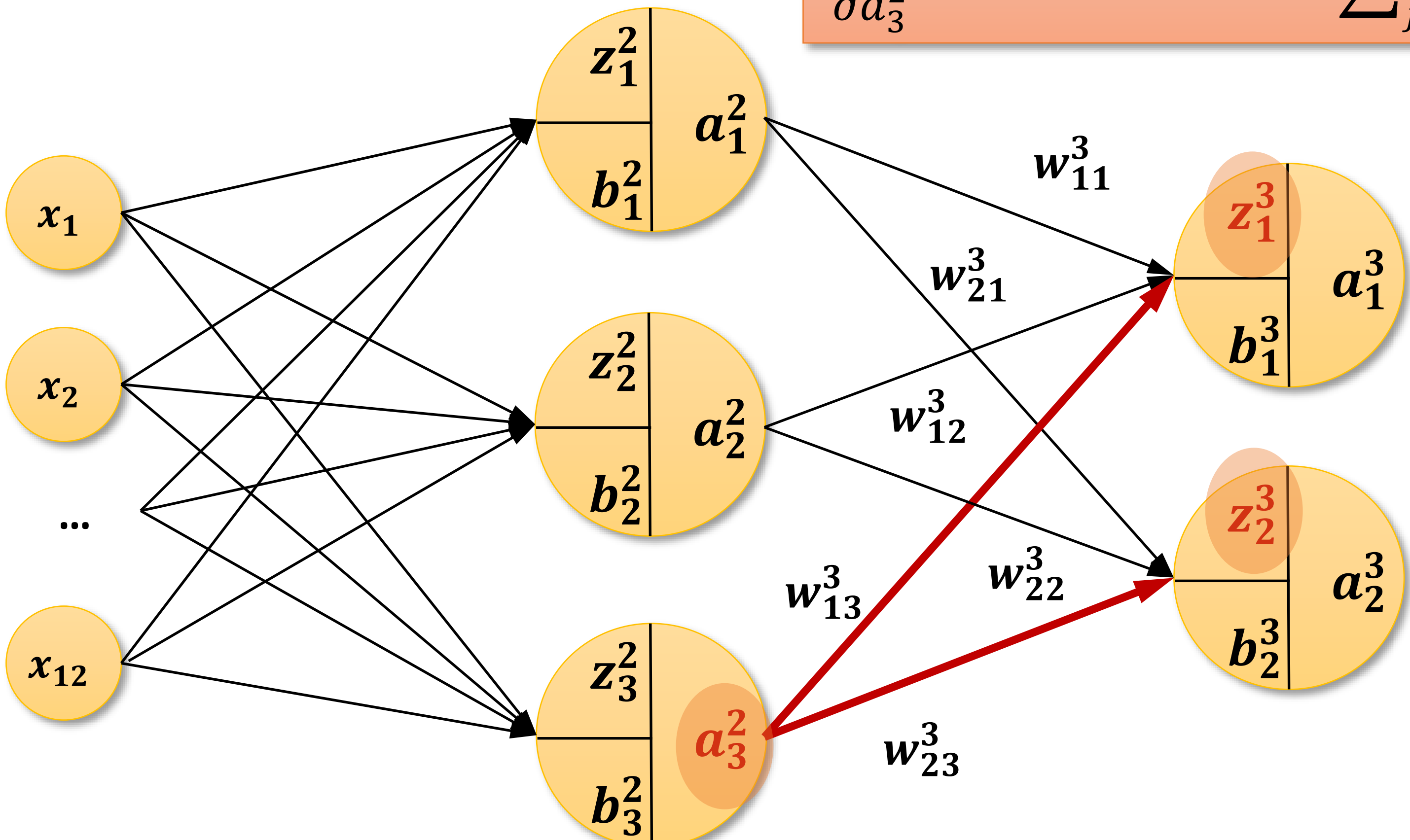


Computing the Error for a_j^{l-1}

$$\delta_j^l = \frac{\partial \mathcal{L}}{\partial z_j^l} \quad (l = 2, 3, \dots)$$

$$\frac{\partial \mathcal{L}}{\partial a_3^2} = \frac{\partial z_1^3}{\partial a_3^2}\frac{\partial \mathcal{L}}{\partial z_1^3} + \frac{\partial z_2^3}{\partial a_3^2}\frac{\partial \mathcal{L}}{\partial z_2^3}$$

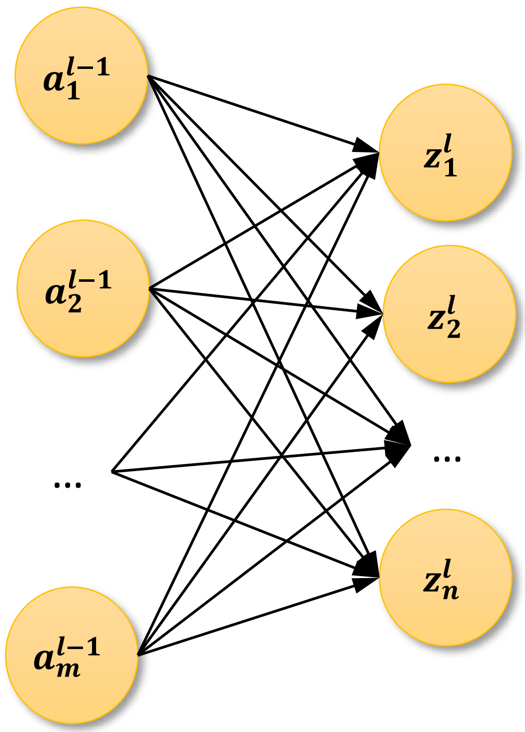
$$\frac{\partial \mathcal{L}}{\partial a_3^2} = w_{13}^3\delta_1^3 + w_{23}^3\delta_2^3 = \sum_{j=1}^{2} w_{j3}^3\delta_j^3$$



Computing the Error for a_j^{l-1}

- Computing the error for a_j^{l-1}

$$\frac{\partial \mathcal{L}}{\partial a_j^{l-1}} = \sum_{j=1}^{n} \frac{\partial z_j^l}{\partial a_j^{l-1}} \delta_j^l = \sum_{i=1}^{n} w_{ji}^l \delta_j^l$$



$$\begin{bmatrix} \frac{\partial \mathcal{L}}{\partial a_1^{l-1}} \\ \frac{\partial \mathcal{L}}{\partial a_2^{l-1}} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial a_m^{l-1}} \end{bmatrix} = \begin{bmatrix} \frac{\partial z_1^l}{\partial a_1^{l-1}} & \frac{\partial z_2^l}{\partial a_1^{l-1}} & \cdots & \frac{\partial z_n^l}{\partial a_1^{l-1}} \\ \frac{\partial z_1^l}{\partial a_2^{l-1}} & \frac{\partial z_2^l}{\partial a_2^{l-1}} & \cdots & \frac{\partial z_n^l}{\partial a_2^{l-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial z_1^l}{\partial a_m^{l-1}} & \frac{\partial z_2^l}{\partial a_m^{l-1}} & \cdots & \frac{\partial z_n^l}{\partial a_m^{l-1}} \end{bmatrix} \begin{bmatrix} \delta_1^l \\ \delta_2^l \\ \vdots \\ \delta_n^l \end{bmatrix}$$

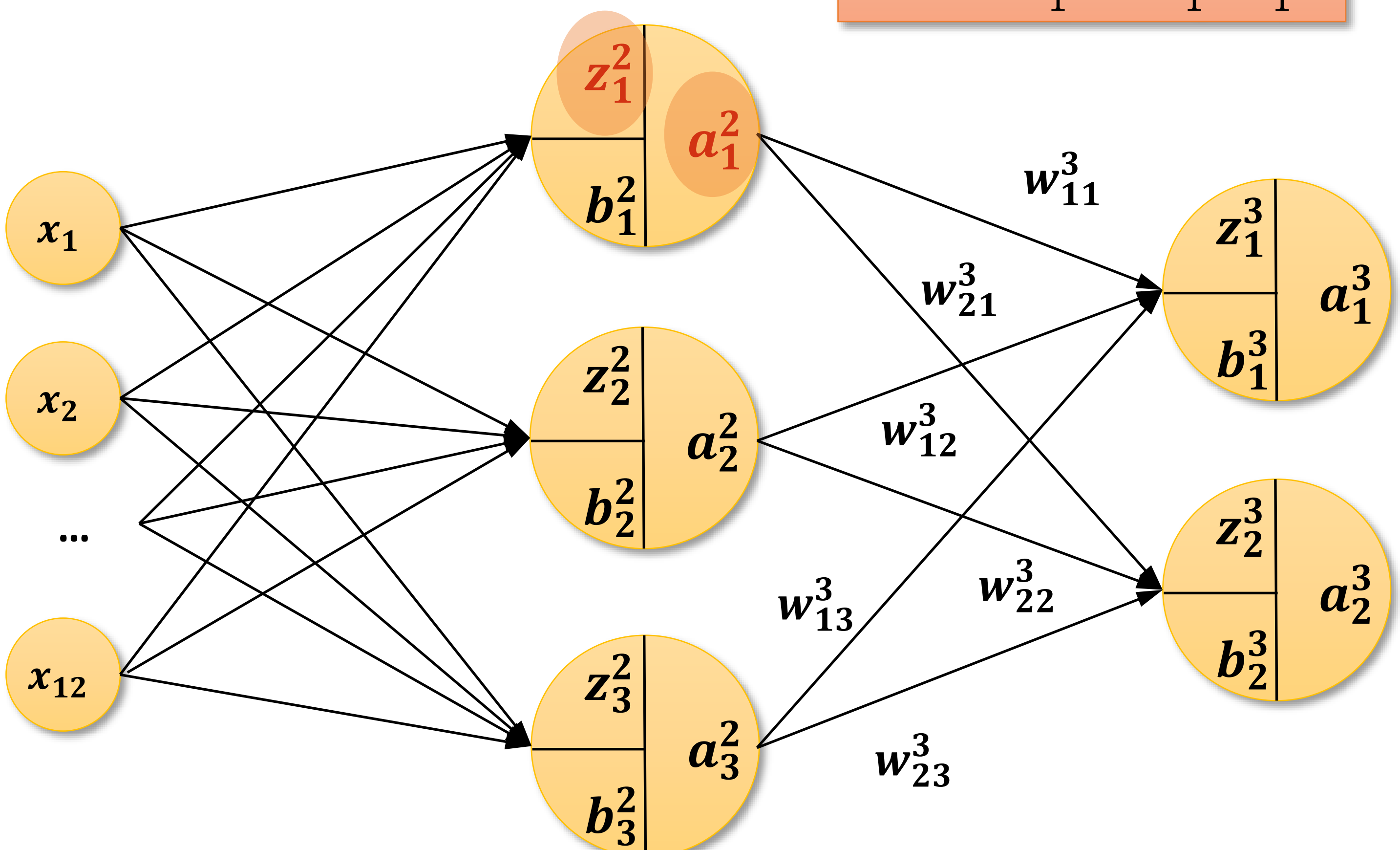
$$\begin{bmatrix} \Delta a_1^{l-1} \\ \Delta a_2^{l-1} \\ \vdots \\ \Delta a_m^{l-1} \end{bmatrix} = \begin{bmatrix} w_{11}^l & w_{21}^l & \cdots & w_{n1}^l \\ w_{12}^l & w_{22}^l & \cdots & w_{n2}^l \\ \vdots & \vdots & \ddots & \vdots \\ w_{1m}^l & w_{2m}^l & \cdots & w_{nm}^l \end{bmatrix} \begin{bmatrix} \delta_1^l \\ \delta_2^l \\ \vdots \\ \delta_n^l \end{bmatrix}$$

Computing the Error δ_j^l

$$\delta_j^l = \frac{\partial \mathcal{L}}{\partial z_j^l} \quad (l = 2, 3, \dots)$$

$$a_1^2 = \sigma(z_1^2)$$

$$\delta_1^2 = \frac{\partial \mathcal{L}}{\partial z_1^2} = \frac{\partial a_1^2}{\partial z_1^2}\frac{\partial \mathcal{L}}{\partial a_1^2}$$

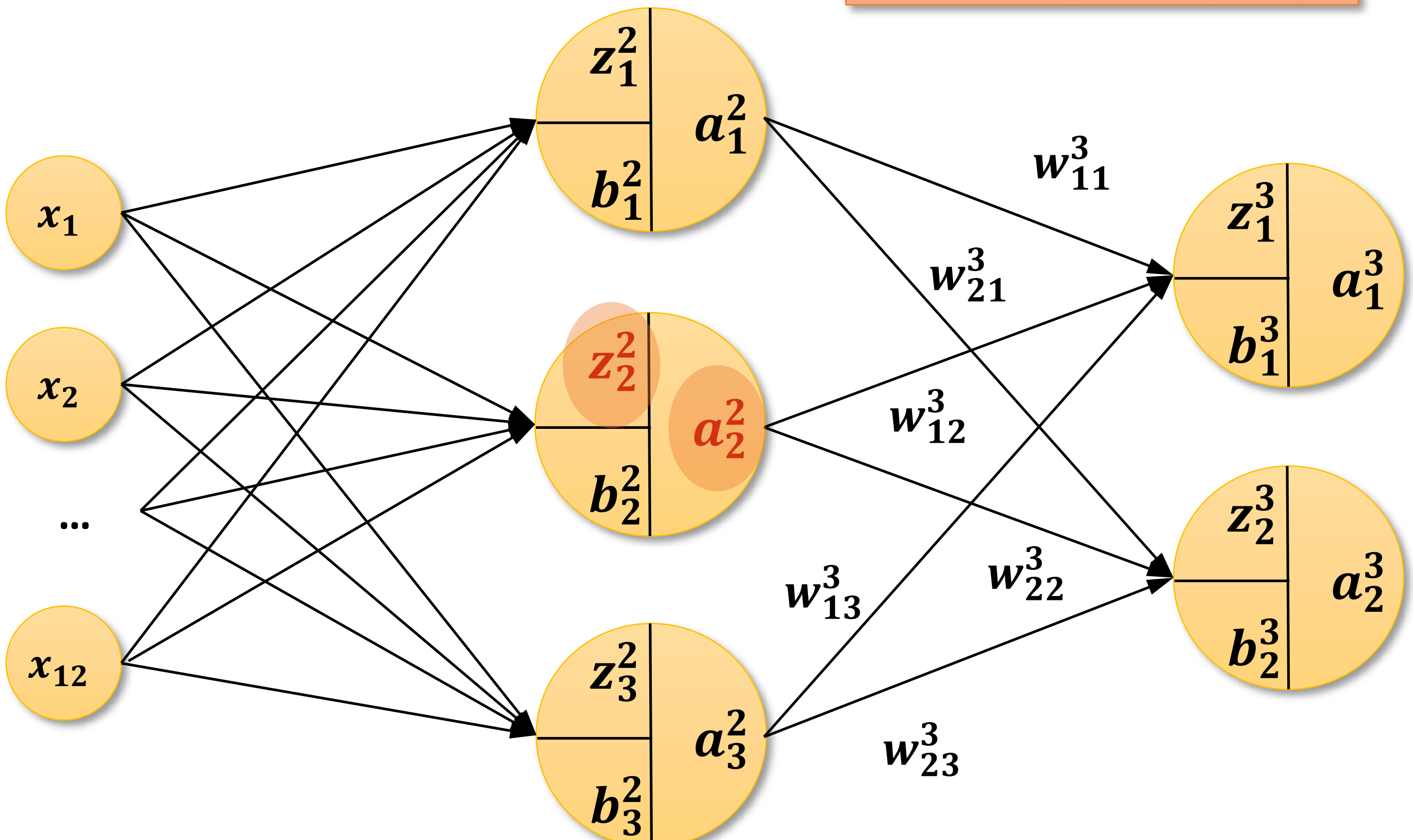


Computing the Error δ_j^l

$$\delta_j^l = \frac{\partial \mathcal{L}}{\partial z_j^l} \quad (l = 2, 3, \dots)$$

$$a_2^2 = \sigma(z_2^2)$$

$$\delta_2^2 = \frac{\partial \mathcal{L}}{\partial z_2^2} = \frac{\partial a_2^2}{\partial z_2^2} \frac{\partial \mathcal{L}}{\partial a_2^2}$$

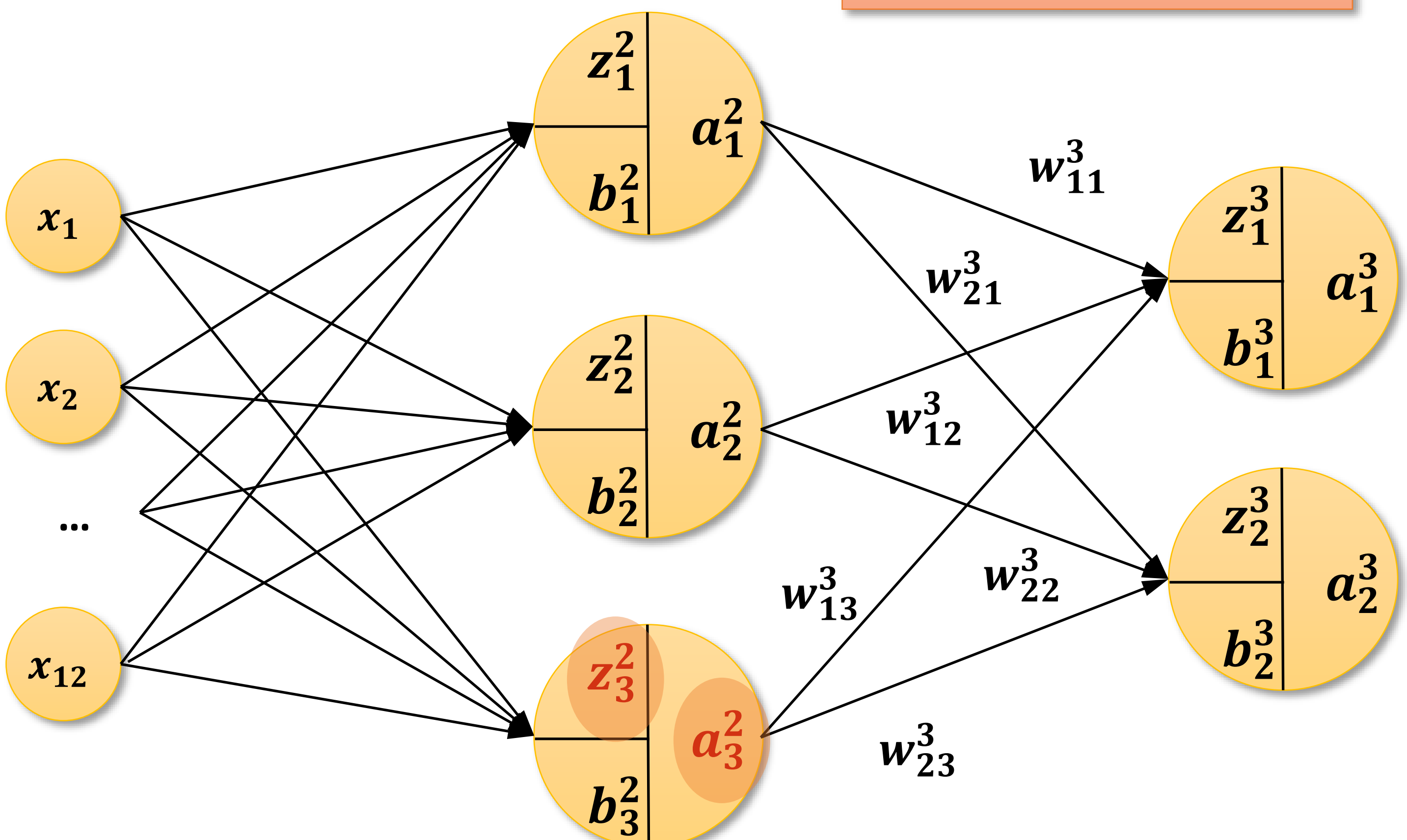


Computing the Error δ_j^l

$$\delta_j^l = \frac{\partial \mathcal{L}}{\partial z_j^l} \quad (l = 2, 3, \ldots)$$

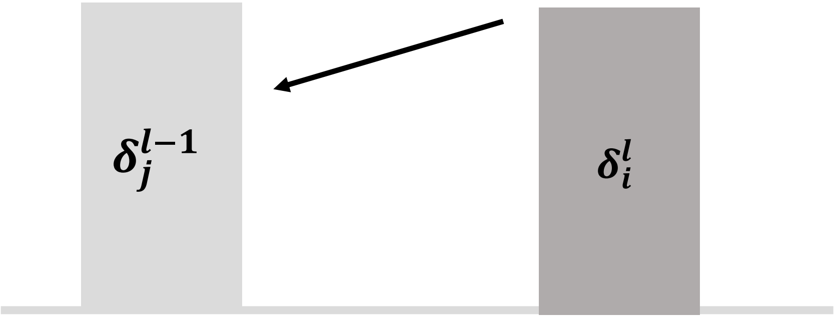
$$a_3^2 = \sigma(z_3^2)$$

$$\delta_3^2 = \frac{\partial \mathcal{L}}{\partial z_3^2} = \frac{\partial a_3^2}{\partial z_3^2}\frac{\partial \mathcal{L}}{\partial a_3^2}$$



Error Backpropagation

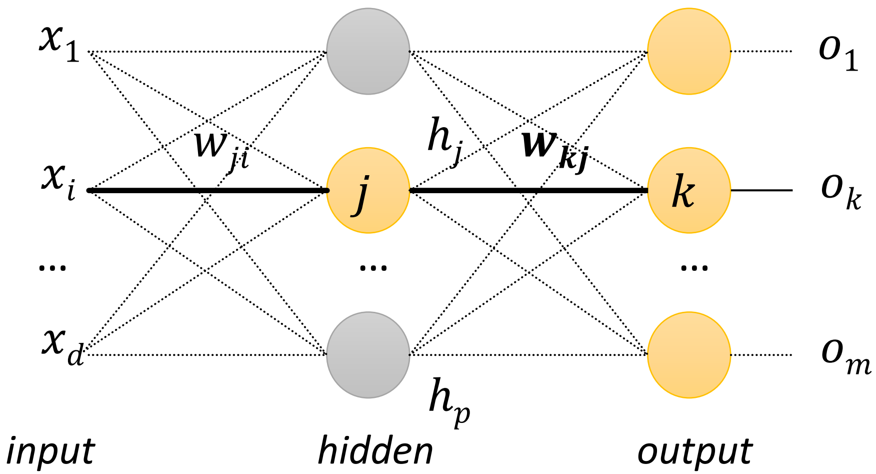
Wow! we can reuse gradient $\boldsymbol{\delta_i^l}$ to compute $\boldsymbol{\delta_j^{l-1}}$.



Example: Error Backpropagation

$(x_1, x_2, \dots, x_d, t_1, t_2, \dots, t_m)$

$(o_1, o_2, \dots, o_m)$



$$o_k = \frac{1}{1 + \exp\left(-\left(w_{k0} + \sum_{j=1}^{p} w_{kj} h_j\right)\right)}$$

$$h_j = \frac{1}{1 + \exp\left(-\left(w_{j0} + \sum_{i=1}^{d} w_{ji} x_i\right)\right)}$$

$$\mathcal{L}_n(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^{m} \left(t_k - \frac{1}{1 + \exp\left(-\left(w_{k0} + \sum_{j=1}^{p} w_{kj} \left(\frac{1}{1 + \exp\left(-\left(w_{j0} + \sum_{i=1}^{d} w_{ji} x_i\right)\right)}\right)\right)\right)} \right)^2$$

Example: Error Backpropagation

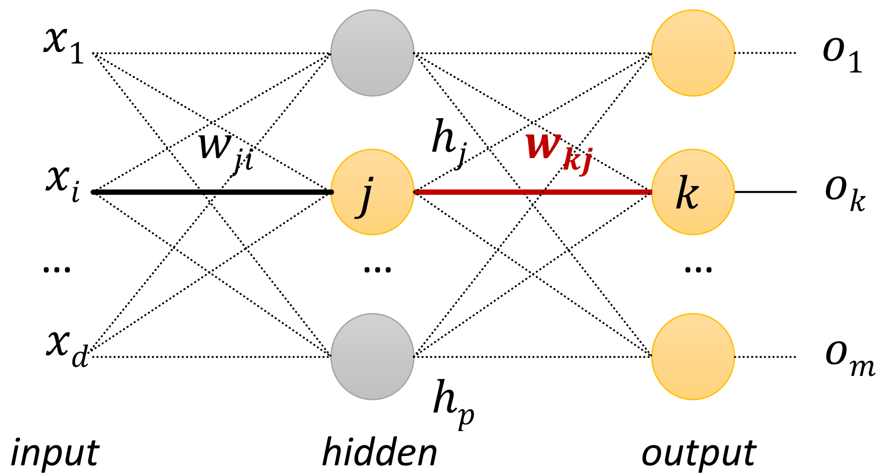
- **Too complex to differentiate**

$$\mathcal{L}_n(\mathbf{w}) = \frac{1}{2}\sum_{k=1}^{m}\left(t_k - \frac{1}{1+\exp\left(-\left(w_{k0}+\sum_{j=1}^{p} w_{kj}\left(\frac{1}{1+\exp\left(-\left(w_{j0}+\sum_{i=1}^{d} w_{ji}x_i\right)\right)}\right)\right)\right)}\right)^2$$

- **Let's use chain rule!**
  - Case 1: when *w* is between output and hidden layer
  - Case 2: when *w* is between hidden and input layer

Case 1: Computing Δw_{kj}

➢ **Gradient of w_{kj} between output and hidden layer**



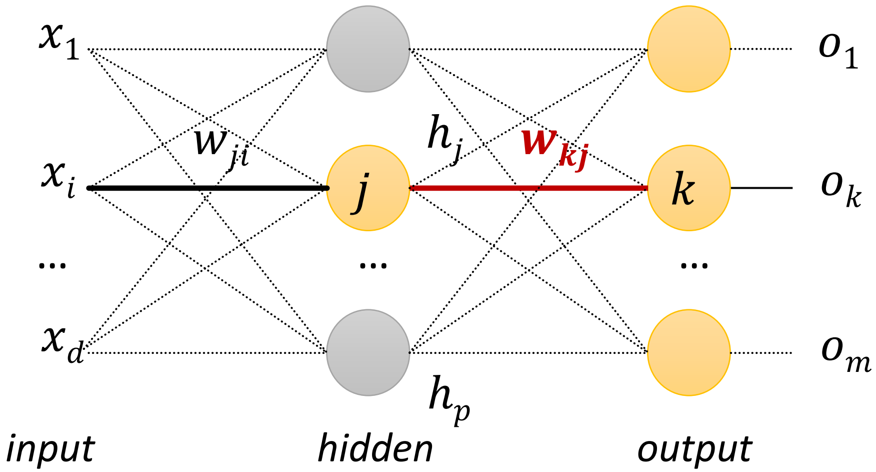
h_j: the output of j-th node in the hidden layer given $\mathbf{x}$

$$net_k = h_1 w_{k1} + h_2 w_{k2} + \cdots + h_p w_{kp}$$

$$o_k = sigmoid(net_k) \qquad \mathcal{L}_n(\mathbf{w}) = \frac{1}{2}\sum_{k=1}^{m}(t_k - o_k)^2$$

Case 1: Computing Δw_{kj}

- **Gradient of w_{kj} between output and hidden layer**



$$\frac{\partial \mathcal{L}_n}{\partial w_{kj}} = \frac{\partial \mathcal{L}_n}{\partial net_k}\frac{\partial net_k}{\partial w_{kj}}$$
$$= \frac{\partial \mathcal{L}_n}{\partial o_k}\frac{\partial o_k}{\partial net_k}\frac{\partial net_k}{\partial w_{kj}}$$
$$= \frac{\partial \mathcal{L}_n}{\partial o_k}\frac{\partial o_k}{\partial net_k} h_j$$

h_j: the output of j-th node in the hidden layer given $\mathbf{x}$

$$net_k = h_1 w_{k1} + h_2 w_{k2} + \cdots + h_p w_{kp}$$

$$o_k = sigmoid(net_k) \qquad \mathcal{L}_n(\mathbf{w}) = \frac{1}{2}\sum_{k=1}^{m}(t_k - o_k)^2$$

Case 1: Computing Δw_{kj}

- **Gradient of w_{kj} between output and hidden layer**

$$\frac{\partial \mathcal{L}_n}{\partial w_{kj}} = \frac{\partial \mathcal{L}_n}{\partial o_k} \frac{\partial o_k}{\partial net_k} h_j$$

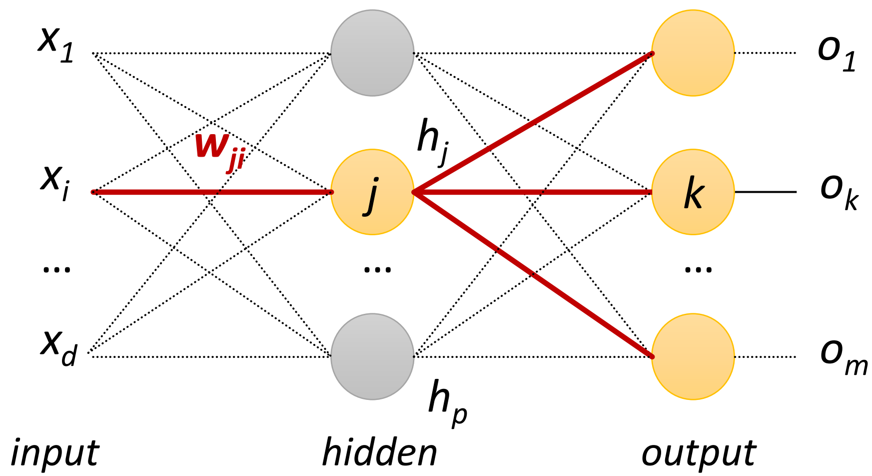
$$\frac{\partial o_k}{\partial net_k} = \frac{\partial\, sigmoid(net_k)}{\partial net_k} = o_k(1 - o_k)$$

$$\frac{\partial \mathcal{L}_n}{\partial o_k} = \frac{\partial}{\partial o_k} \frac{1}{2}(t_k - o_k)^2 = \frac{1}{2} 2(t_k - o_k) \frac{\partial (t_k - o_k)}{\partial o_k} = -(t_k - o_k)$$

$$\frac{\partial \mathcal{L}_n}{\partial w_{kj}} = -(t_k - o_k) o_k (1 - o_k) h_j$$

Case 2: Computing Δw_{ji}

➢ **Gradient of w_{ji} between output and hidden layer**



$net_j = x_1 w_{j1} + x_2 w_{j2} + \cdots + x_d w_{jd}$ $\qquad$ $net_k = h_1 w_{k1} + h_2 w_{k2} + \cdots + h_p w_{kp}$

$h_j = sigmoid(net_j)$ $\qquad$ $o_k = sigmoid(net_k)$ $\qquad$ $\mathcal{L}_n(\mathbf{w}) = \dfrac{1}{2}\sum_{k=1}^{m}(t_k - o_k)^2$

Case 2: Computing Δw_{ji}

- **Gradient of w_{ji} between output and hidden layer**

$$\frac{\partial \mathcal{L}_n}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} \frac{1}{2} \sum_{k=1}^{m} (t_k - o_k)^2$$

$$= \frac{1}{2} \sum_{k=1}^{m} \frac{\partial}{\partial w_{ji}} (t_k - o_k)^2$$

$$= \frac{1}{2} \sum_{k=1}^{m} \frac{\partial (t_k - o_k)^2}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial h_j} \frac{\partial h_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}}$$

$$= \frac{1}{2} \sum_{k=1}^{m} -2(t_{nk} - o_{nk}) \cdot o_k(1 - o_k) \cdot w_{kj} \cdot h_j(1 - h_j) \cdot x_i$$

$$= h_j(1 - h_j) x_i \sum_{k=1}^{m} -w_{kj}(t_k - o_k) o_k (1 - o_k)$$

$$\frac{\partial (t_k - o_k)^2}{\partial o_k} = -2(t_k - o_k)$$

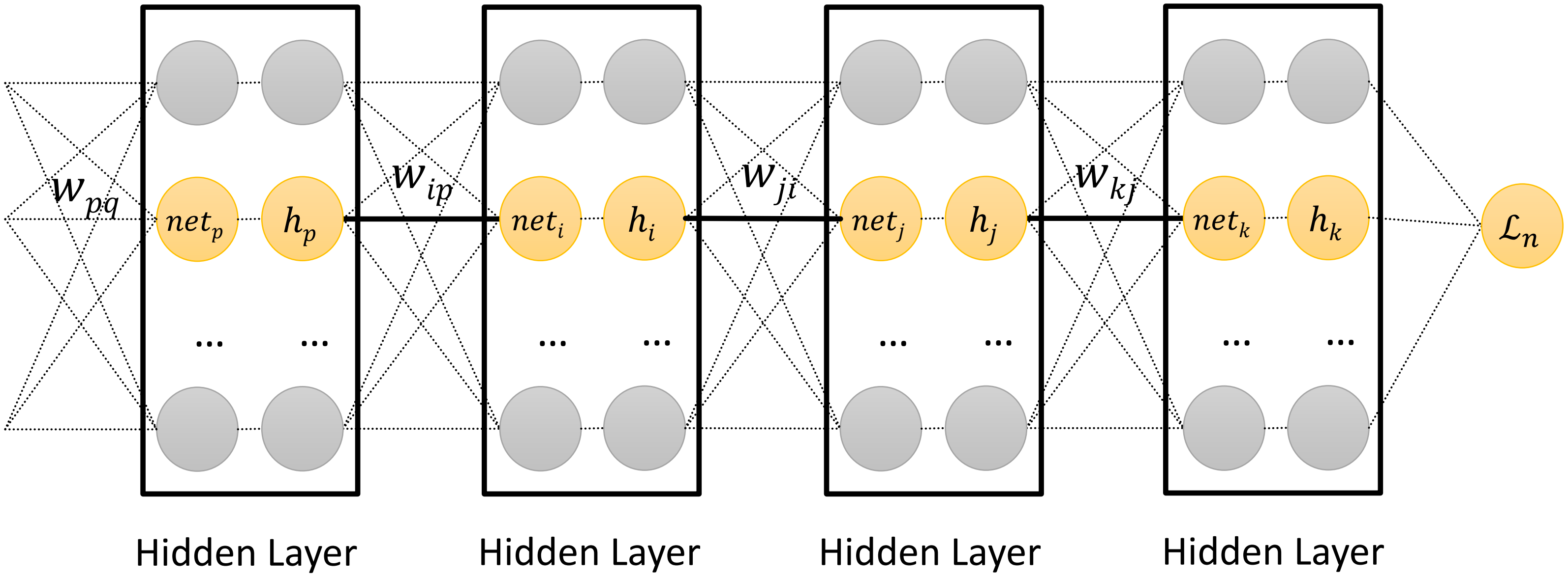
$$\frac{\partial o_k}{\partial net_k} = o_k(1 - o_k)$$

$$\frac{\partial net_k}{\partial h_j} = w_{kj}$$

$$\frac{\partial h_j}{\partial net_j} = h_j(1 - h_j)$$

$$\frac{\partial net_j}{\partial w_{ji}} = x_i$$

How to Update Deep Layers

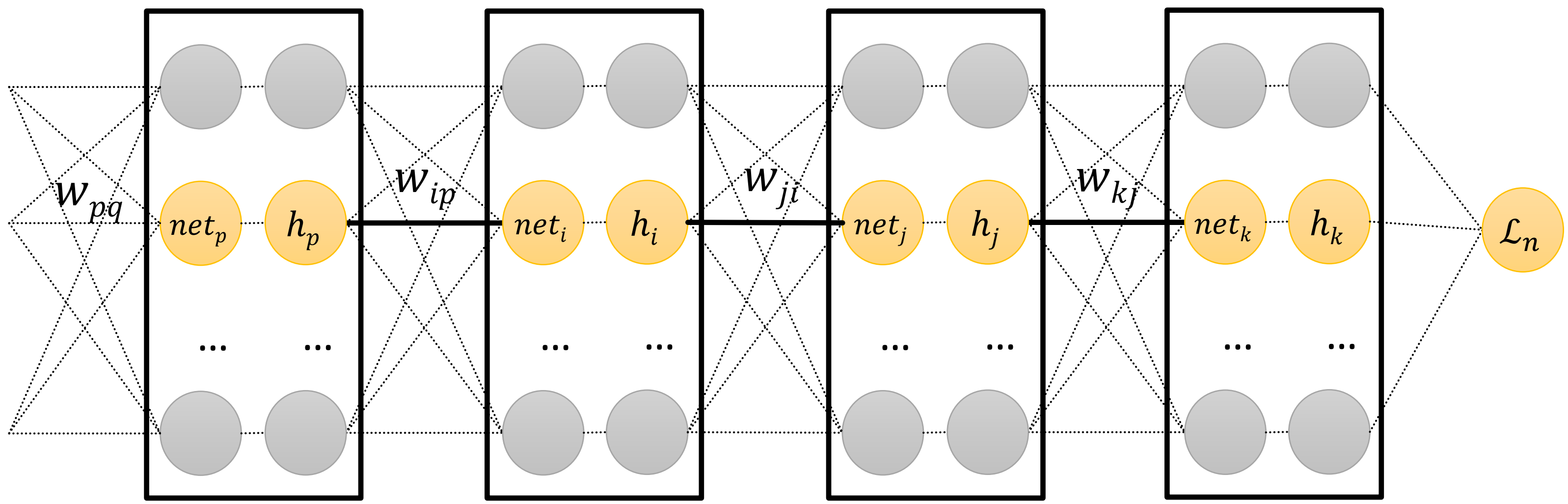


$$\frac{\partial \mathcal{L}_n}{\partial w_{kj}} = \frac{\partial \mathcal{L}_n}{\partial net_k}\frac{\partial net_k}{\partial w_{kj}} = \delta_k h_j \qquad \delta_k = \frac{\partial \mathcal{L}_n}{\partial net_k}$$

$$\frac{\partial \mathcal{L}_n}{\partial w_{ji}} = \frac{\partial \mathcal{L}_n}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}} = \delta_j h_i \qquad \delta_j = \frac{\partial \mathcal{L}_n}{\partial net_j}$$

$$\frac{\partial \mathcal{L}_n}{\partial w_{ip}} = \frac{\partial \mathcal{L}_n}{\partial net_i}\frac{\partial net_i}{\partial w_{ip}} = \delta_i h_p \qquad \delta_i = \frac{\partial \mathcal{L}_n}{\partial net_i}$$

How to Update Deep Layers

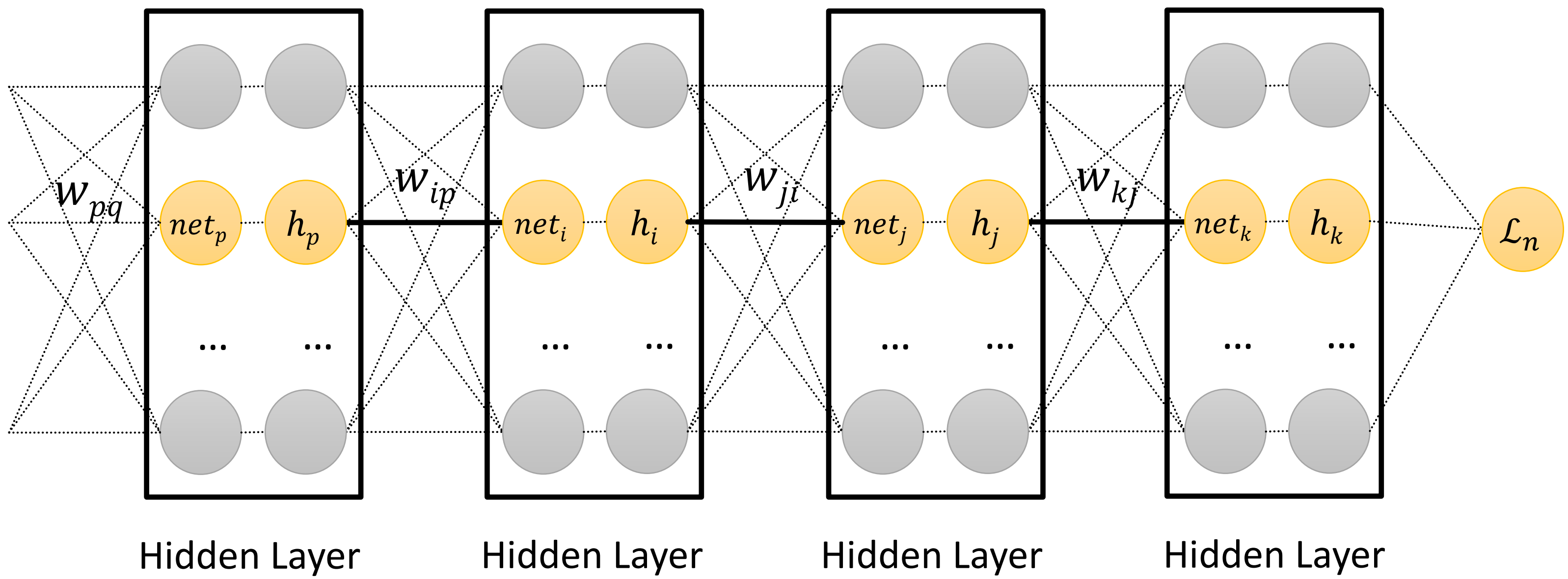


$$\delta_k = \frac{\partial \mathcal{L}_n}{\partial net_k} = \frac{\partial \mathcal{L}_n}{\partial h_k}\frac{\partial h_k}{\partial net_k}$$

$$\delta_j = \frac{\partial \mathcal{L}_n}{\partial net_j} = \frac{\partial \mathcal{L}_n}{\partial h_j}\frac{\partial h_j}{\partial net_j} = \left(\sum_{k=1}^{K} \frac{\partial \mathcal{L}_n}{\partial net_k}\frac{\partial net_k}{\partial h_j}\right)\frac{\partial h_j}{\partial net_j} = \left(\sum_{k=1}^{K} \delta_k w_{kj}\right)\frac{\partial h_j}{\partial net_j}$$

$$\delta_i = \frac{\partial \mathcal{L}_n}{\partial net_i} = \frac{\partial \mathcal{L}_n}{\partial h_i}\frac{\partial h_i}{\partial net_i} = \left(\sum_{j=1}^{J} \frac{\partial \mathcal{L}_n}{\partial net_j}\frac{\partial net_j}{\partial h_i}\right)\frac{\partial h_i}{\partial net_i} = \left(\sum_{j=1}^{J} \delta_j w_{ji}\right)\frac{\partial h_i}{\partial net_i}$$

How to Update Deep Layers



$$\frac{\partial \mathcal{L}_n}{\partial w_{kj}} = \frac{\partial \mathcal{L}_n}{\partial net_k} \frac{\partial net_k}{\partial w_{kj}} = \delta_k h_j$$

$$\delta_k = \frac{\partial \mathcal{L}_n}{\partial h_k} \frac{\partial h_k}{\partial net_k}$$

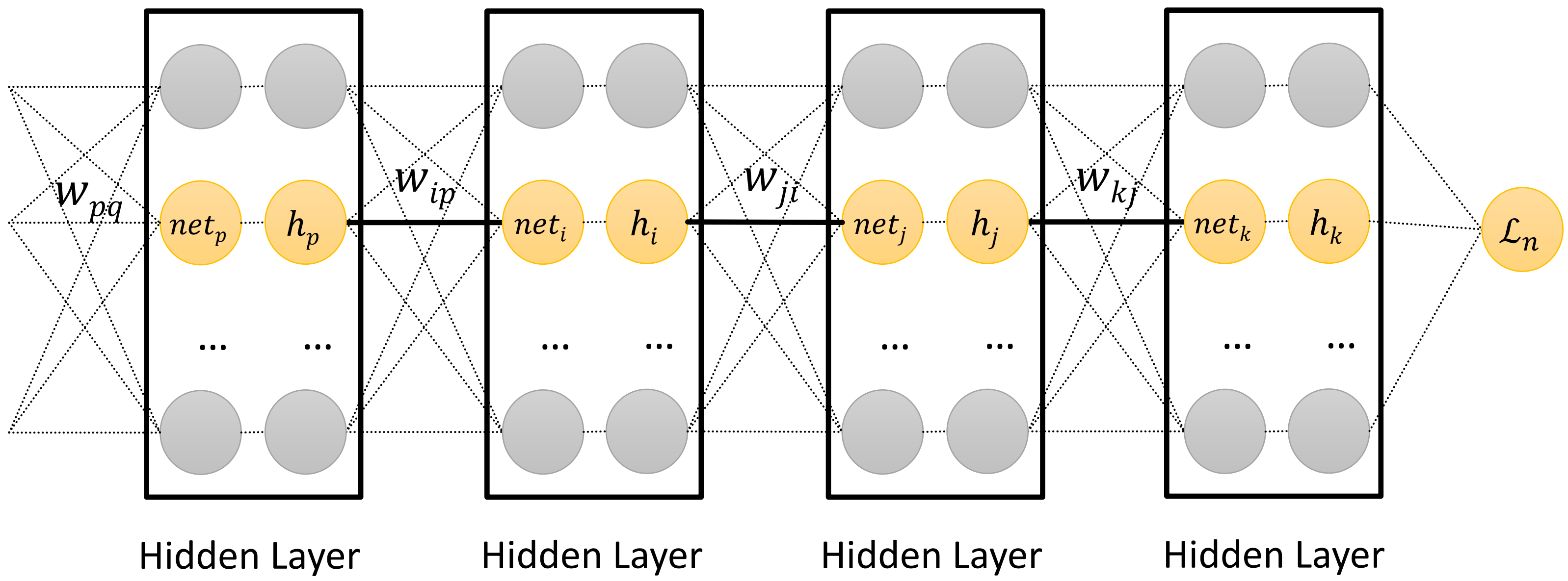
$$\frac{\partial \mathcal{L}_n}{\partial w_{ji}} = \frac{\partial \mathcal{L}_n}{\partial net_{nj}} \frac{\partial net_{nj}}{\partial w_{ji}} = \delta_j h_i$$

$$\delta_j = \left(\sum_{k=1}^{K} \delta_k w_{kj} \right) \frac{\partial h_j}{\partial net_j}$$

$$\frac{\partial \mathcal{L}_n}{\partial w_{ip}} = \frac{\partial \mathcal{L}_n}{\partial net_i} \frac{\partial net_i}{\partial w_{ip}} = \delta_i h_p$$

$$\delta_i = \left(\sum_{j=1}^{J} \delta_j w_{ji} \right) \frac{\partial h_i}{\partial net_i}$$

How to Update Deep Layers



$$\frac{\partial \mathcal{L}_n}{\partial w_{kj}} = \frac{\partial \mathcal{L}_n}{\partial net_k}\frac{\partial net_k}{\partial w_{kj}} = \delta_k h_j$$

$$\delta_k = \frac{\partial \mathcal{L}_n}{\partial h_k}\frac{\partial h_k}{\partial net_k}$$

$$\frac{\partial \mathcal{L}_n}{\partial w_{ji}} = \frac{\partial \mathcal{L}_n}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}} = \delta_j h_i$$

$$\delta_j = \left(\sum_{k=1}^{K} \delta_k w_{kj}\right)\frac{\partial h_j}{\partial net_j} = \left(\sum_{k=1}^{K} \delta_k w_{kj}\right) h_j(1 - h_j)$$

$$\frac{\partial \mathcal{L}_n}{\partial w_{ip}} = \frac{\partial \mathcal{L}_n}{\partial net_i}\frac{\partial net_i}{\partial w_{ip}} = \delta_i h_p$$

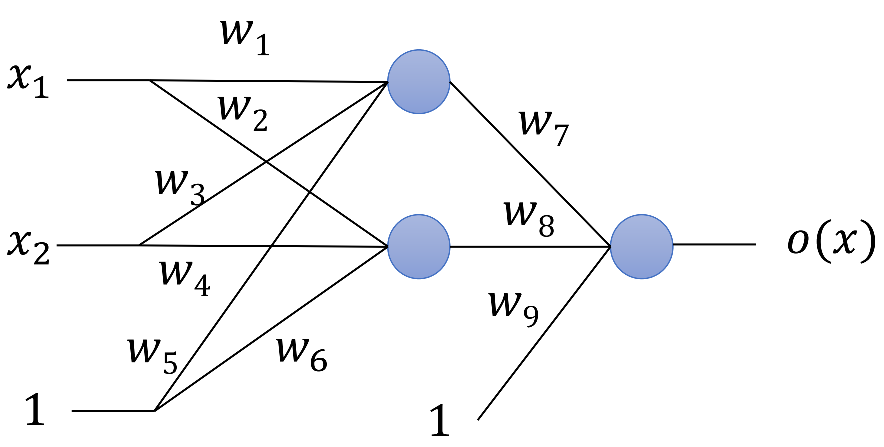
$$\delta_i = \left(\sum_{j=1}^{J} \delta_j w_{ji}\right)\frac{\partial h_i}{\partial net_i} = \left(\sum_{j=1}^{J} \delta_j w_{ji}\right) h_i(1 - h_i)$$

Example: Backpropagation

Example: XOR Operator

- **Hidden nodes : 2**
- **Learning rate : 0.5**

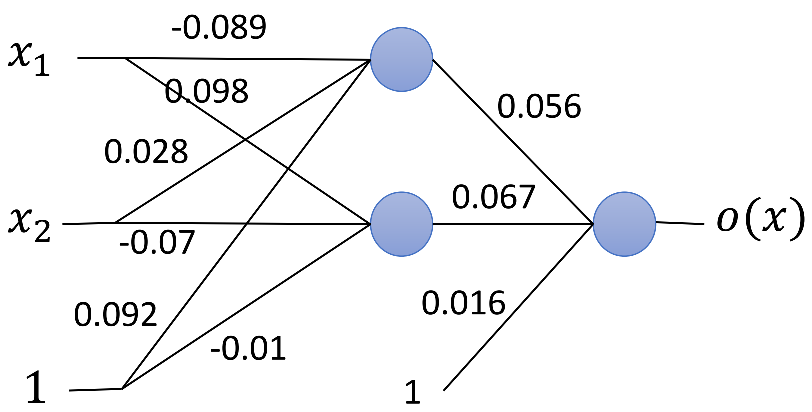
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0



Example: XOR Operator

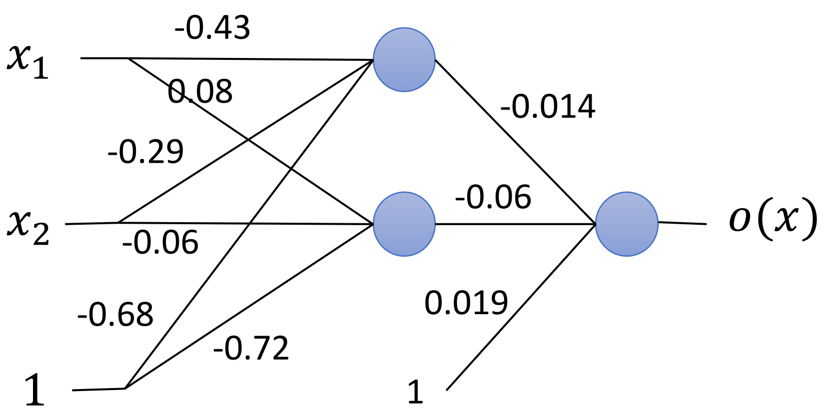
Iteration : 0

x_1	x_2	y	o
0	0	0	0.52
0	1	1	0.50
1	0	1	0.52
1	1	0	0.55



Iteration : 1,000

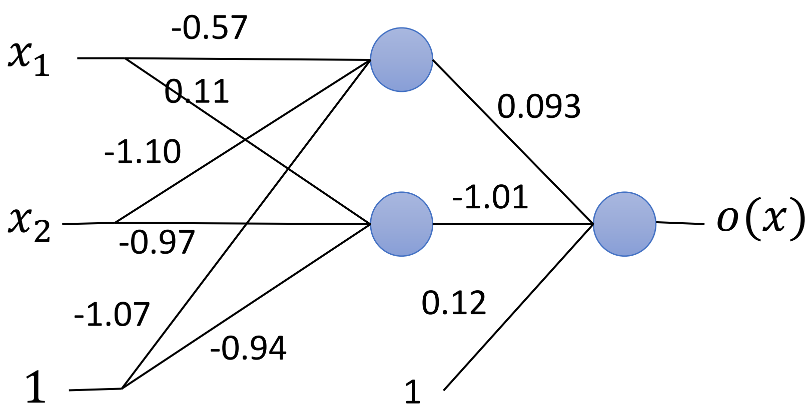
x_1	x_2	y	o
0	0	0	0.50
0	1	1	0.48
1	0	1	0.50
1	1	0	0.52



Example: XOR Operator

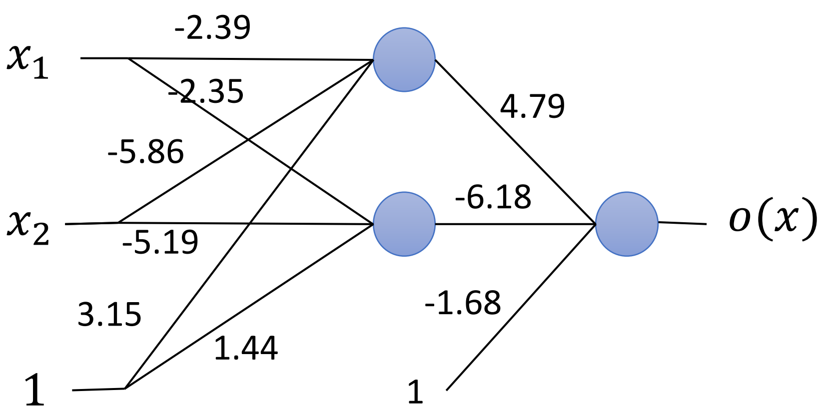
Iteration : 2,000

x_1	x_2	y	o
0	0	0	0.53
0	1	1	0.48
1	0	1	0.50
1	1	0	0.48



Iteration : 3,000

x_1	x_2	y	o
0	0	0	0.30
0	1	1	0.81
1	0	1	0.81
1	1	0	0.11



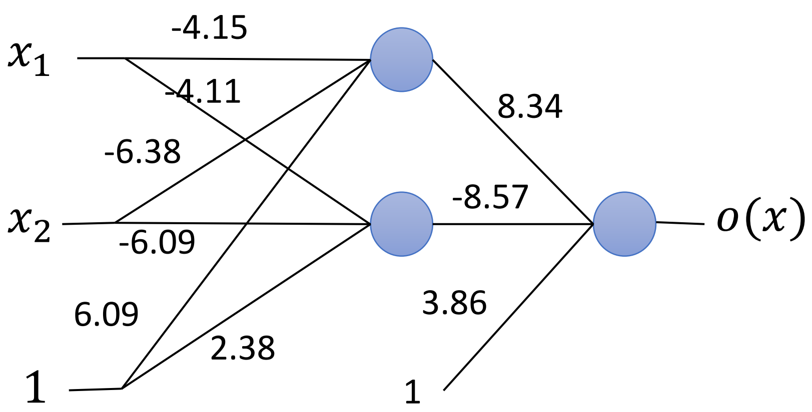
Example: XOR Operator

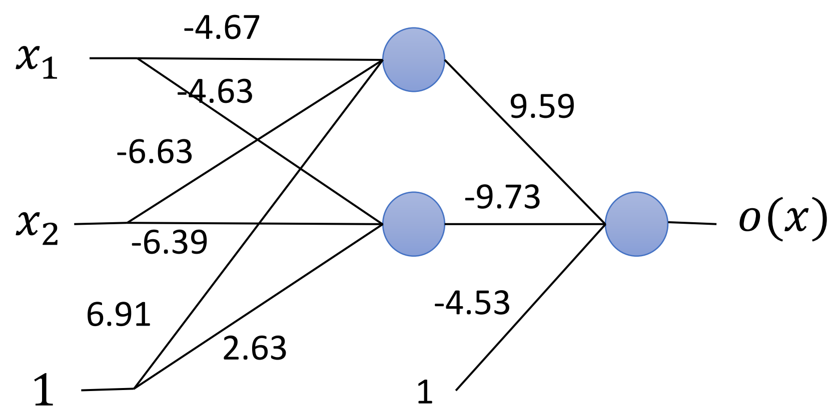
Iteration : 5,000

x_1	x_2	y	o
0	0	0	0.05
0	1	1	0.96
1	0	1	0.96
1	1	0	0.03

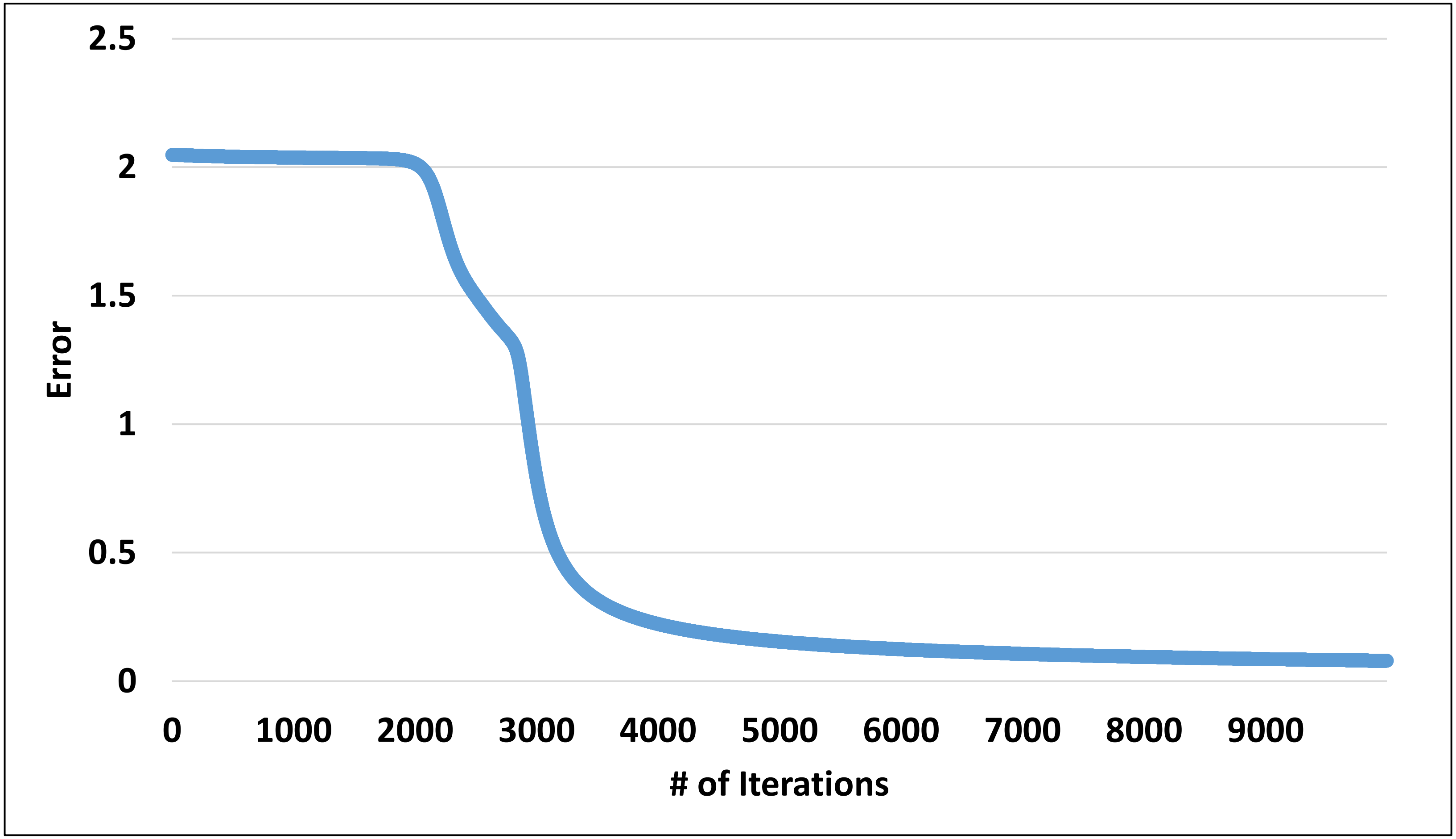
Iteration : 10,000

x_1	x_2	y	o
0	0	0	0.02
0	1	1	0.98
1	0	1	0.98
1	1	0	0.02





Example: XOR Operator



Q&A