CZ4041/CE4041: Machine Learning

Lesson 7b: Neural Networks (Multi-Layer)

Kelly KE

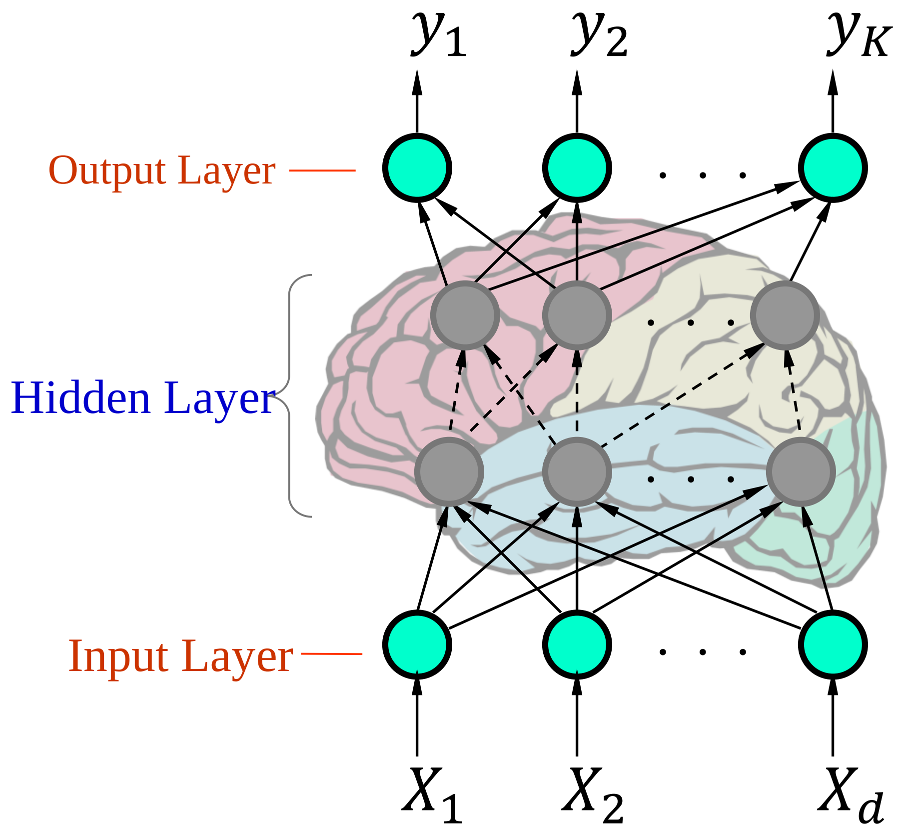
School of Computer Science and Engineering, NTU, Singapore

Acknowledgements: Some figures are adapted from the lecture notes of the books "Introduction to Machine Learning" (Chap. 11) and "Introduction to Data Mining" (Chap. 5). Slides are modified from the version prepared by Dr. Sinno Pan.

Outline

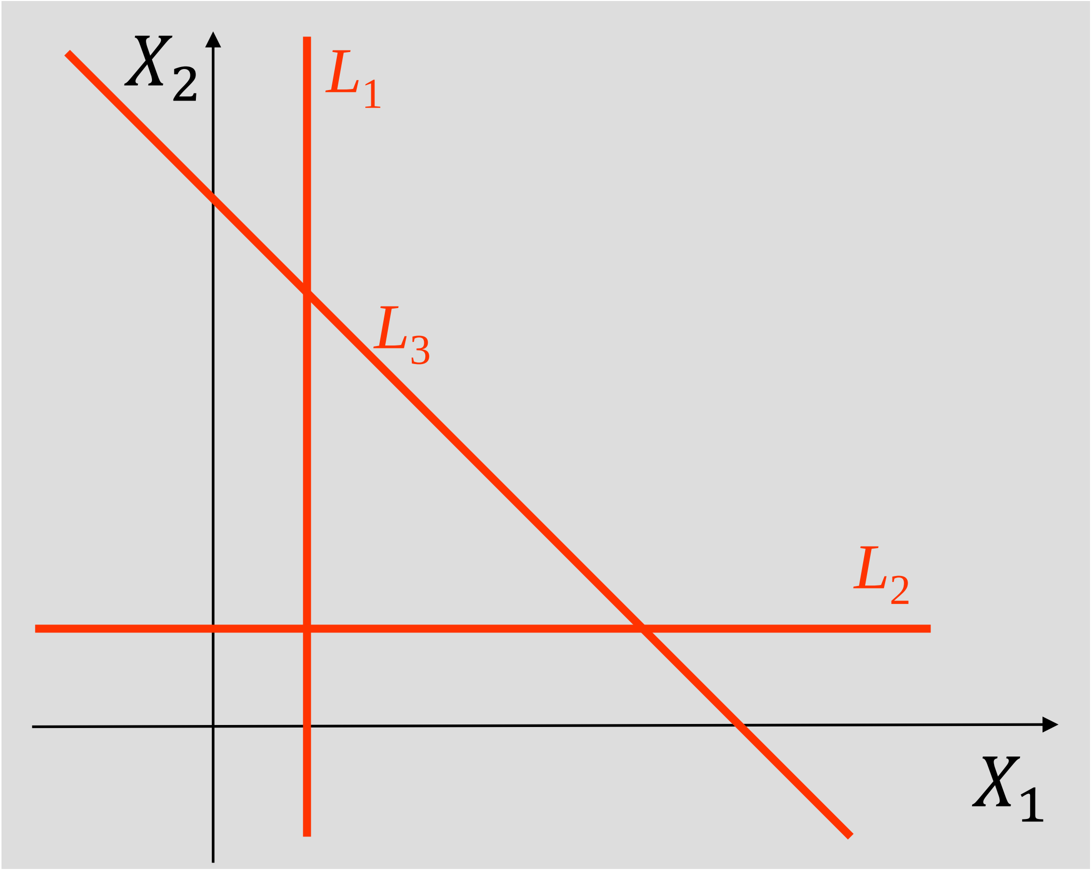
- Artificial Neural Networks
  - Perceptrons
  - Multi-layer Neural Networks

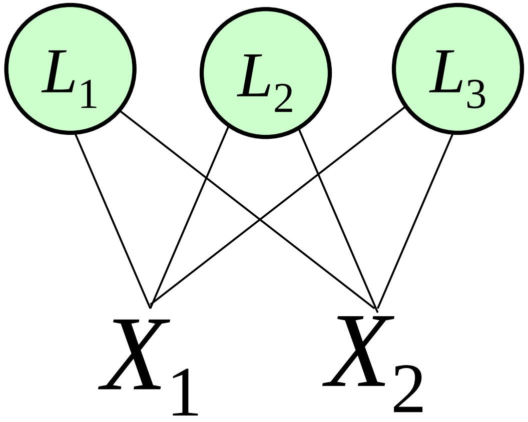
General Structure: Multilayer ANN



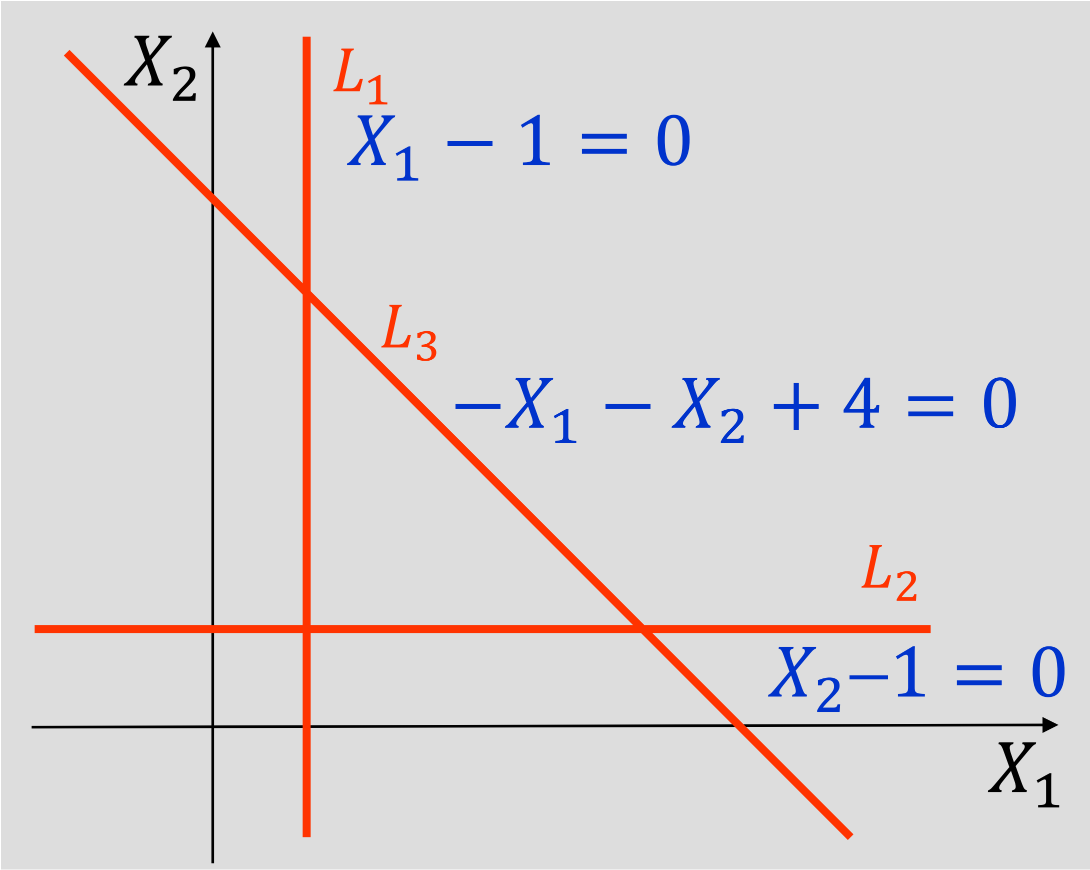
A feed-forward neural network: the nodes in one layer are connected only to the nodes in the next layer

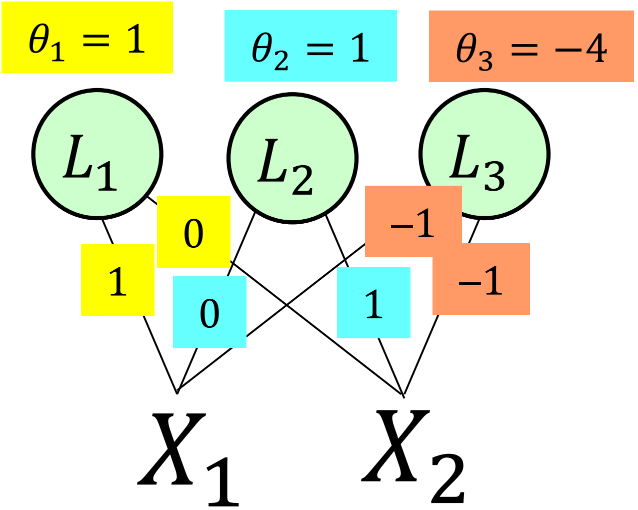
Example: Not Linearly Separable



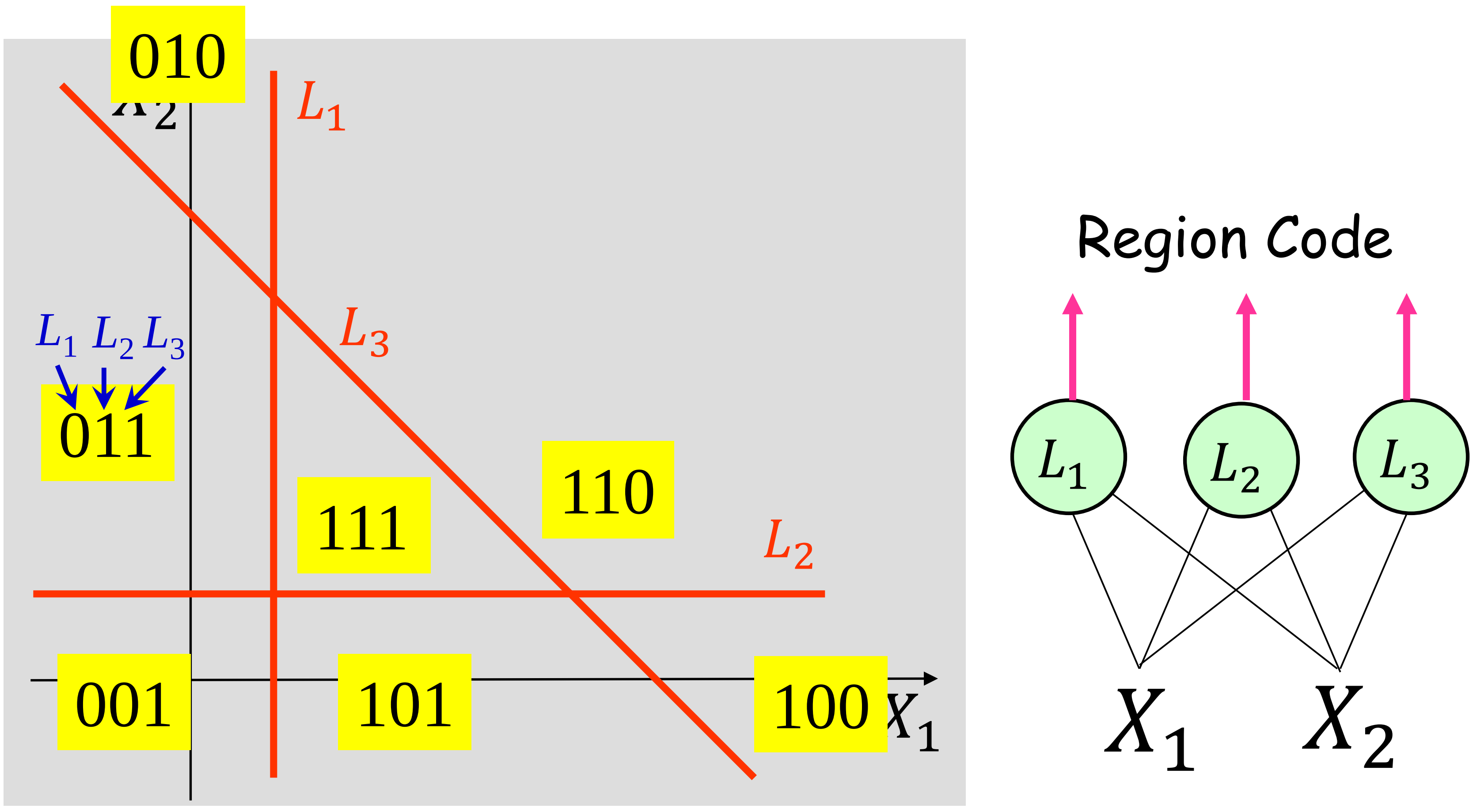


Example: Not Linearly Separable

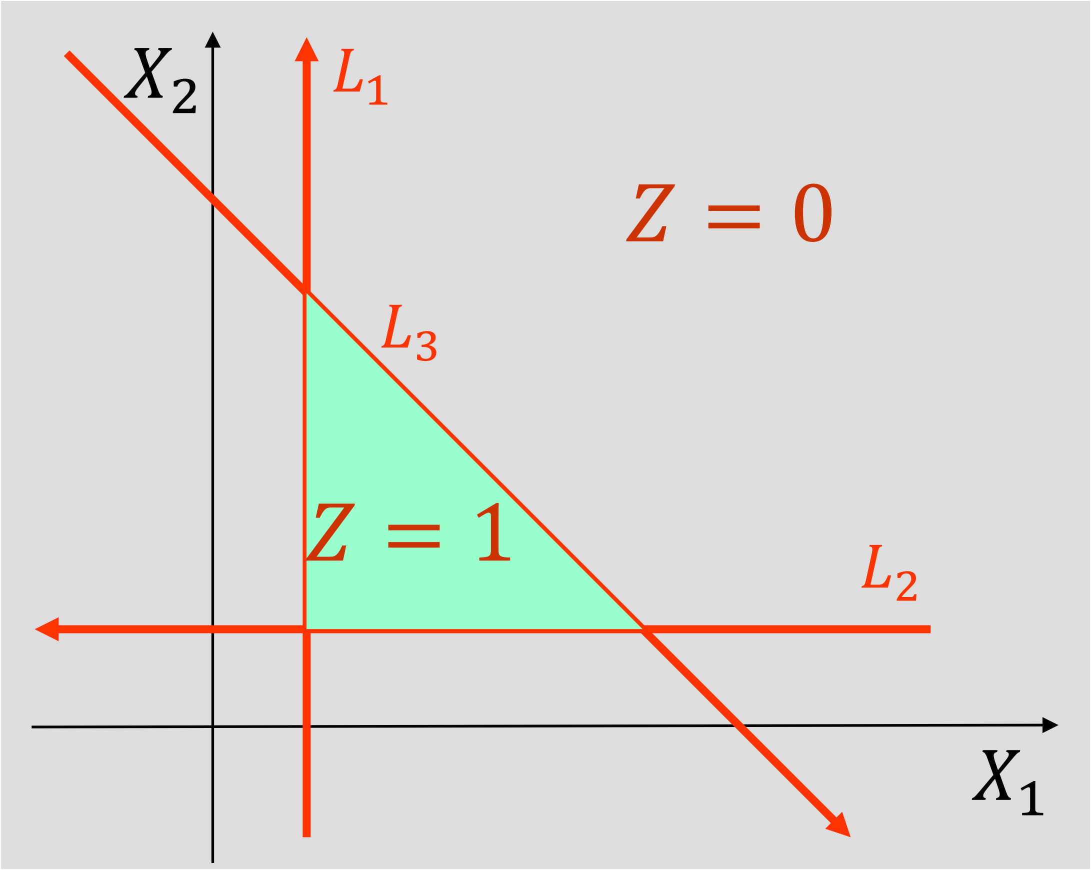


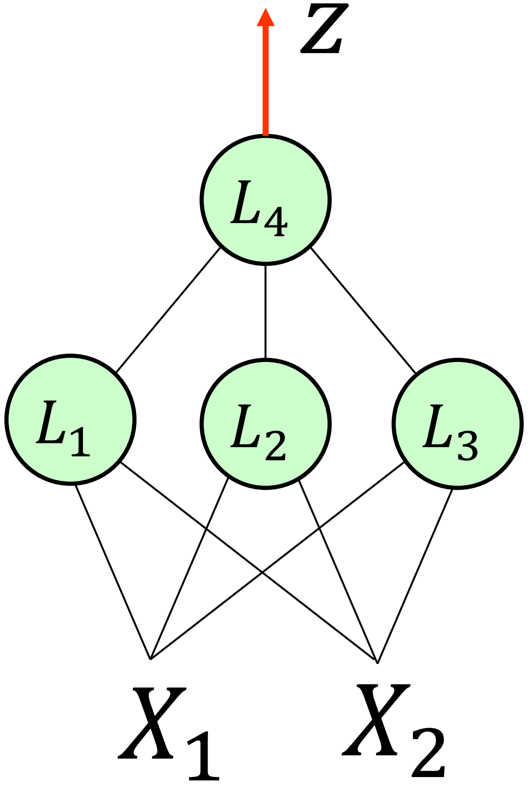


Example: Not Linearly Separable

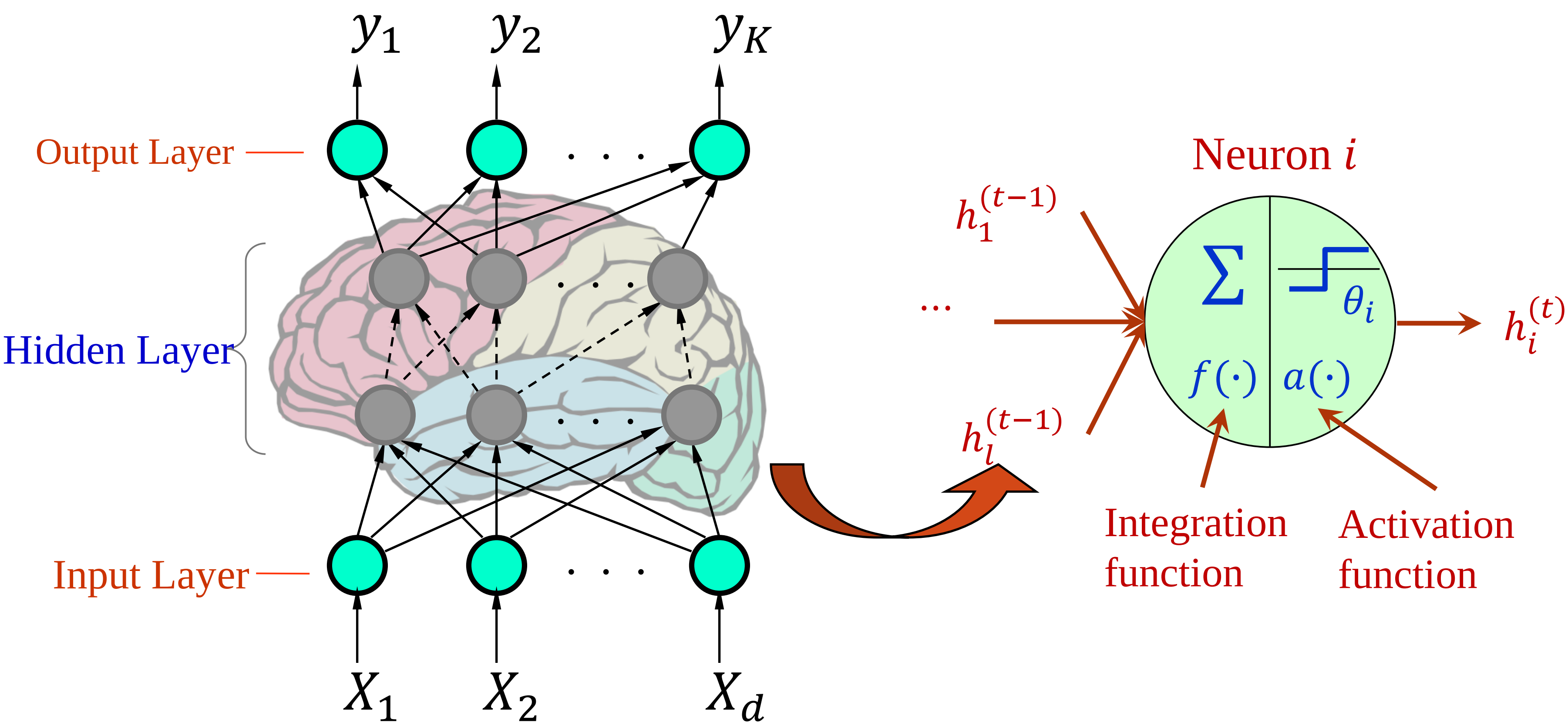


Example: Not Linearly Separable …





General Structure: Multilayer ANN



Integration Functions

- Weighted sum:

$$\sum_{i=1}^{d} w_i X_i - \theta$$

- Quadratic function

$$\sum_{i=1}^{d} w_i X_i^2 - \theta$$

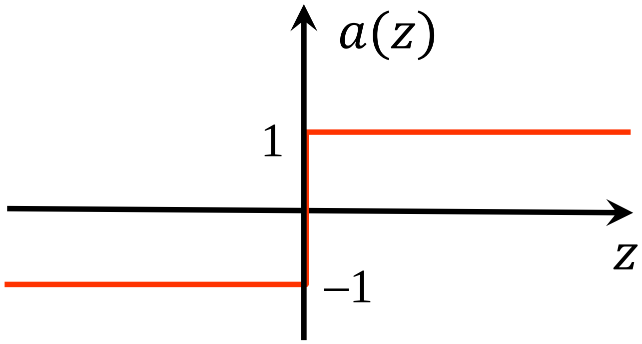
- Spherical function

$$\sum_{i=1}^{d} (X_i - w_i)^2 - \theta$$

Activation Functions

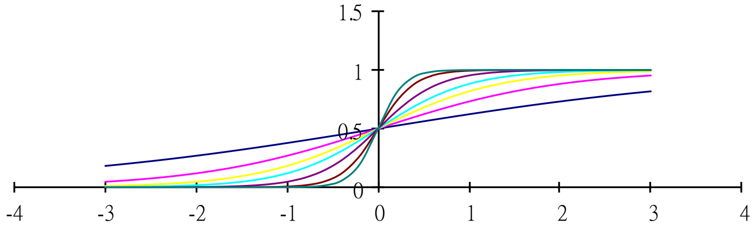
- Sign function (Threshold function)

$$a(z) = \mathrm{sign}(z) = \begin{cases} 1 & z \geq 0 \\ -1 & z < 0 \end{cases}$$



- Unipolar sigmoid function:

$$a(z) = \frac{1}{1 + e^{-\lambda z}}$$



When $\lambda = 1$, it is called sigmoid function

Update Weights for Multi-layer NNs

- Initialize the weights in each layer $\left(\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(k)}, \ldots, \mathbf{w}^{(m)}\right)$
- Adjust the weights such that the output of ANN is consistent with class labels of training examples
  - Loss function for each training instance:

$$E = \frac{1}{2}(y_i - \hat{y}_i)^2$$

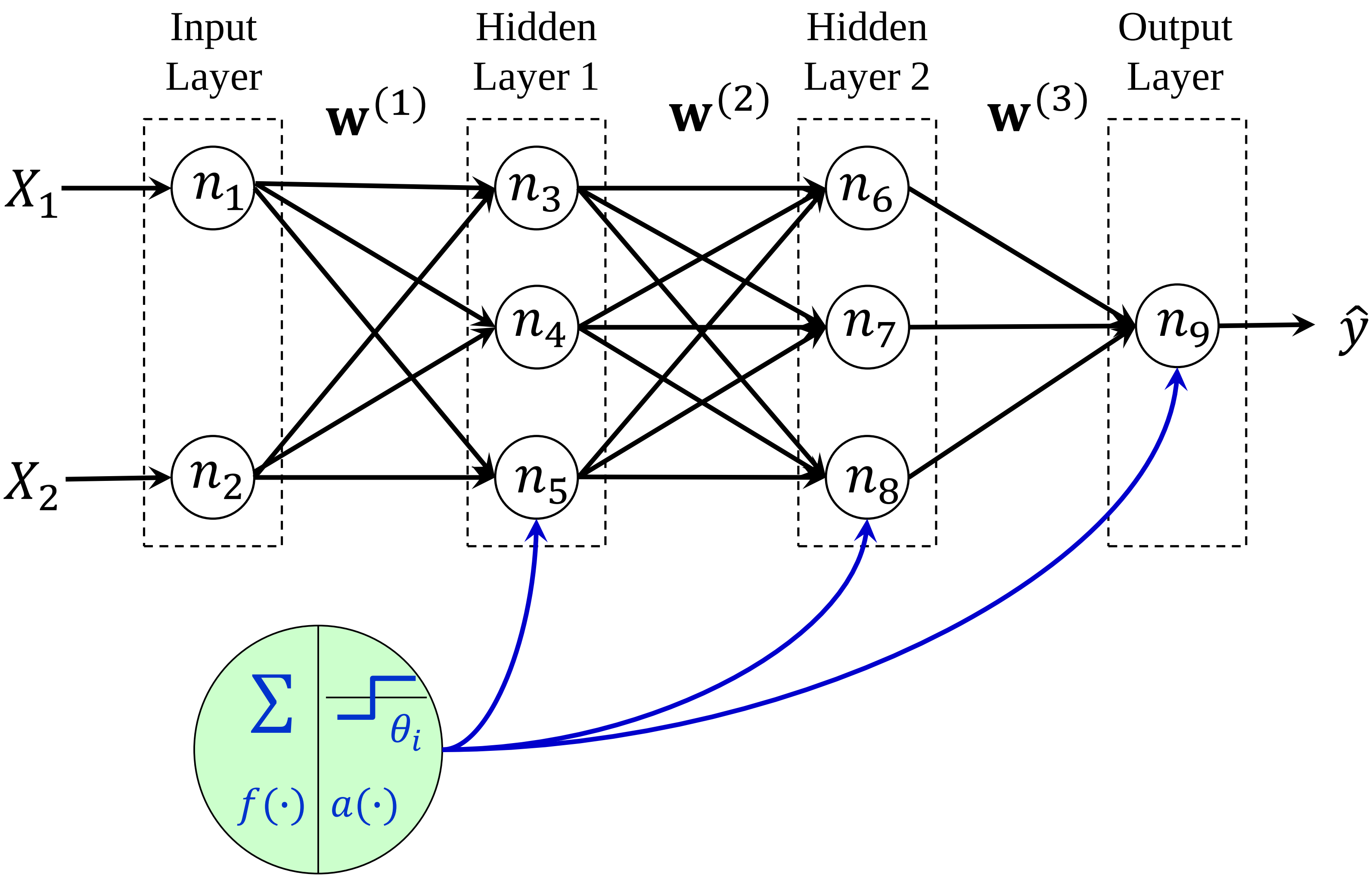
  - For each layer k, update the weights, $\mathbf{w}^{(k)}$, by gradient descent at each iteration t:

$$\mathbf{w}_{t+1}^{(k)} = \mathbf{w}_t^{(k)} - \lambda \frac{\partial E}{\partial \mathbf{w}^{(k)}}$$

  - Computing an analytical expression for the gradient w.r.t. weights in each layer is computationally expensive!!!

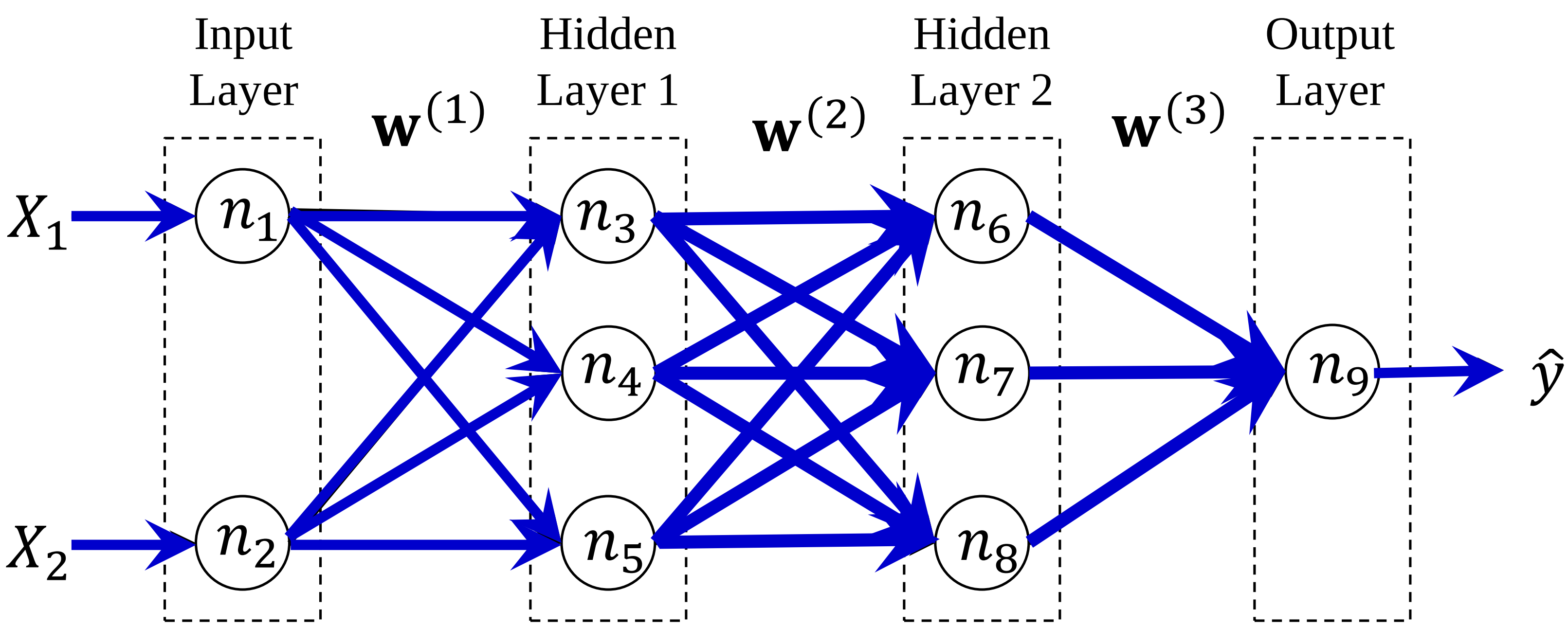
Backpropagation algorithm

A Multi-layer Feed-forward NN



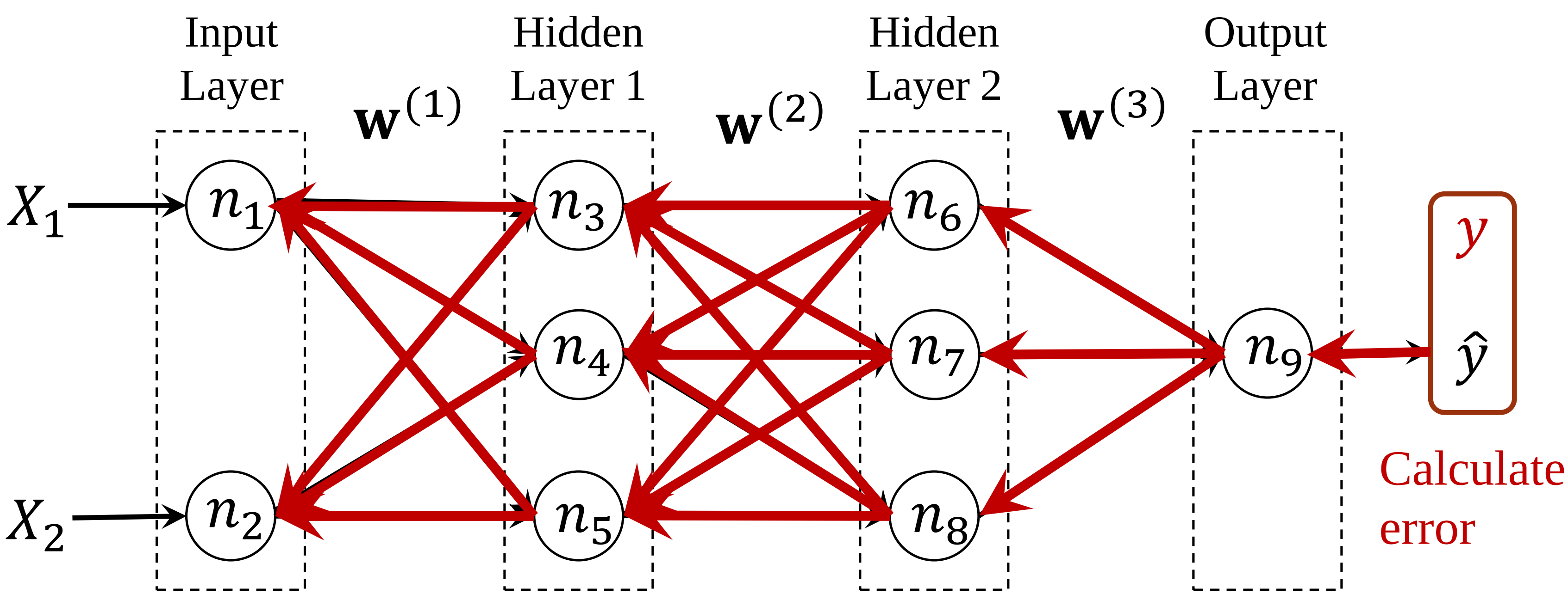
Backpropagation: Basic Idea

- Initialize the weights ($\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(3)}$)
- **Forward pass:** each training examples$(\boldsymbol{x}_i, y_i)$ is used to compute outputs of each hidden layer and generate the final output $\hat{y}_i$ based on the ANN

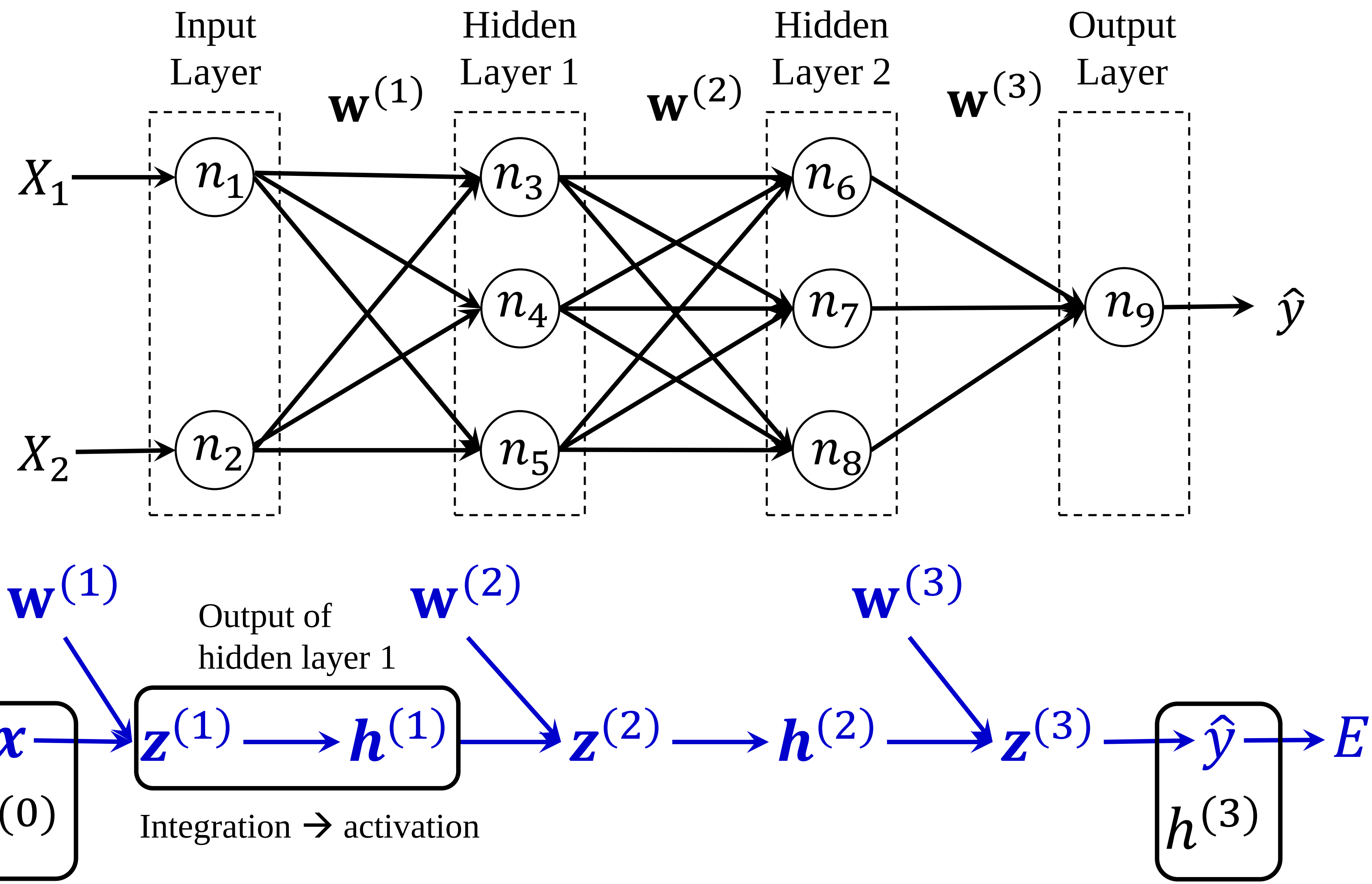


Backpropagation: Basic Idea (cont.)

- **Backpropagation:** Starting with the output layer, to propagate error back to the previous layer in order to update the weights between the two layers, until the earliest hidden layer is reached



The Computational Graph



Backpropagation (BP)

- Gradient of E w.r.t. $w^{(3)}$: $\frac{\partial E}{\partial w^{(3)}} = \boxed{\frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z^{(3)}}} \frac{\partial z^{(3)}}{\partial w^{(3)}}$
- Gradient of E w.r.t. $w^{(2)}$:

$$\frac{\partial E}{\partial w^{(2)}} = \boxed{\boxed{\frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z^{(3)}}} \frac{\partial z^{(3)}}{\partial h^{(2)}} \frac{\partial h^{(2)}}{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial w^{(2)}}$$

- Gradient of E w.r.t. $w^{(1)}$:

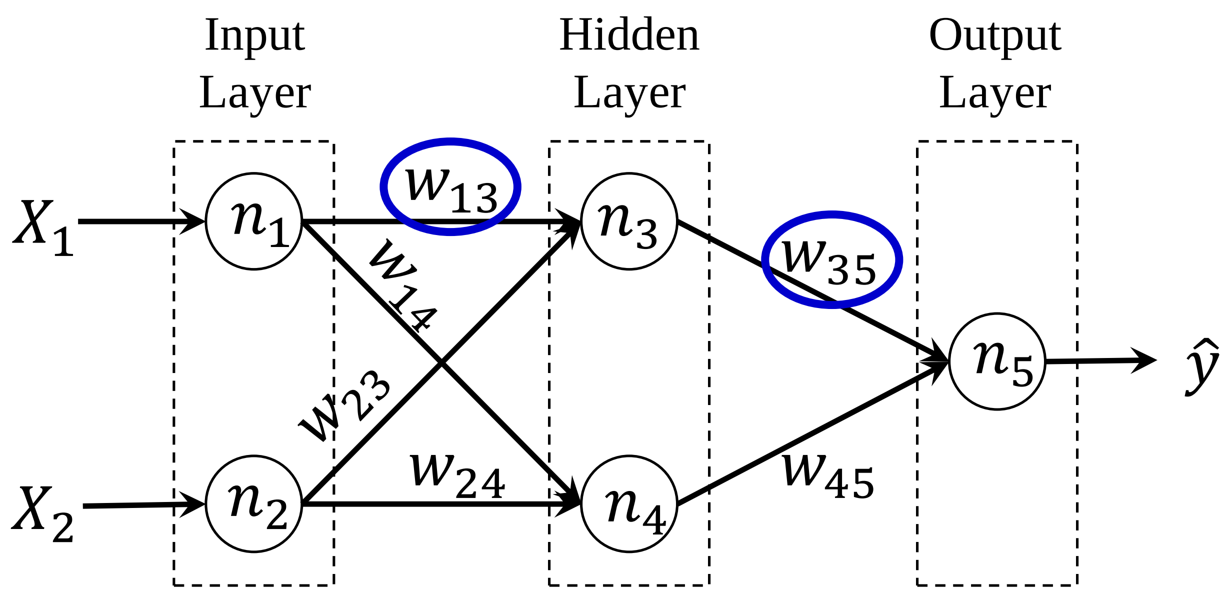
$$\frac{\partial E}{\partial w^{(1)}} = \boxed{\frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial h^{(2)}} \frac{\partial h^{(2)}}{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial h^{(1)}} \frac{\partial h^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial w^{(1)}}$$

Consider each layer contains a single unit

$$\begin{tikzcd}
 & w^{(1)} \arrow[d] & & w^{(2)} \arrow[d] & & w^{(3)} \arrow[d] & & \\
x \arrow[r] & z^{(1)} \arrow[r] & h^{(1)} \arrow[r] & z^{(2)} \arrow[r] & h^{(2)} \arrow[r] & z^{(3)} \arrow[r] & \hat{y} \arrow[r] & E
\end{tikzcd}$$

An Example

- Consider an ANN of 1 hidden layer as follows. Suppose the sign function and the weighted sum function are used for both hidden and output nodes



$$\begin{tikzcd} & w_{13} \arrow[d] & & w_{35} \arrow[d] & & \\ X_1 \arrow[r] & z_3 \arrow[r] & h_3 \arrow[r] & z_5 \arrow[r] & \hat{y} \arrow[r] & E \end{tikzcd}$$

$$w'_{35} = w_{35} + \lambda E_i h_3$$

$$w'_{35} = w_{35} - \lambda \frac{\partial E}{\partial w_{35}} = w_{35} - \lambda \boxed{\frac{\partial E}{\partial \hat{y}}}\frac{\partial \hat{y}}{\partial z_5}\boxed{\frac{\partial z_5}{\partial w_{35}}}$$

$$E = \frac{1}{2}E_i^2 = \frac{1}{2}(y_i - \hat{y}_i)^2$$

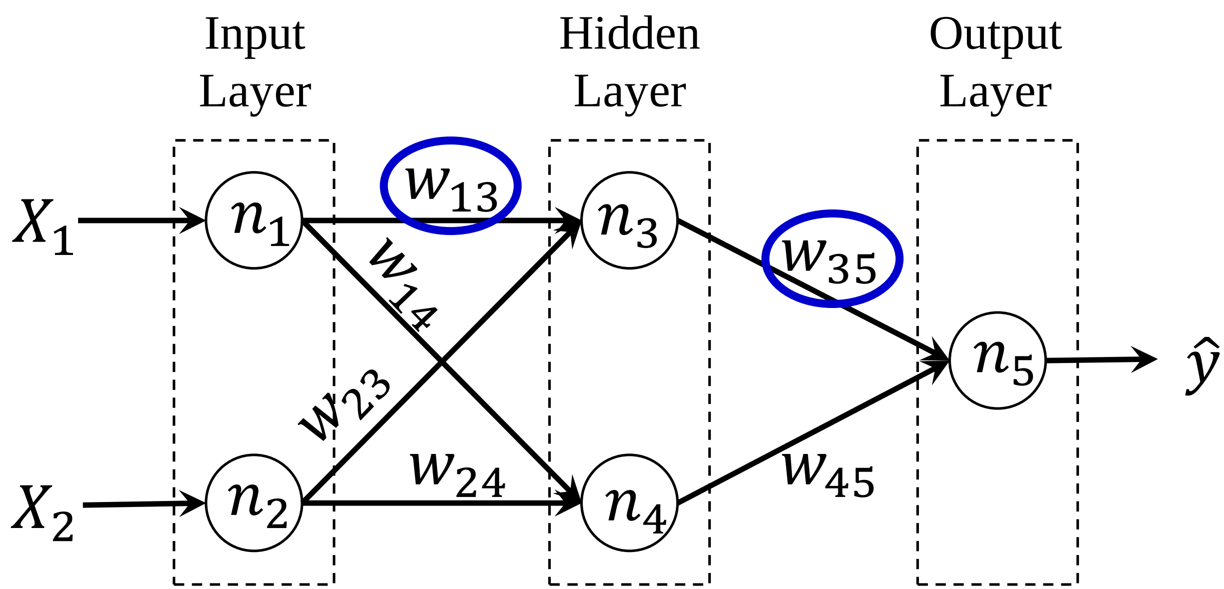
$$-1 \times (y_i - \hat{y}_i) = -E_i$$

$$\cancel{\hat{y} = \mathrm{sgn}(z_5)}$$

$$\hat{y} = z_5 \qquad \frac{\partial \hat{y}}{\partial z_5} = 1$$

$$z_5 = w_{35}h_3 + w_{45}h_4$$

$$h_3$$



$$X_1 \longrightarrow \underset{\nwarrow w_{13}}{z_3} \longrightarrow h_3 \longrightarrow \underset{\nwarrow w_{35}}{z_5} \longrightarrow \hat{y} \longrightarrow E$$

$$w'_{13} = w_{13} + \lambda E_i w_{35} X_1$$

$$w'_{13} = w_{13} - \lambda \frac{\partial E}{\partial w_{13}} = w_{13} - \lambda \boxed{\frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_5}} \boxed{\frac{\partial z_5}{\partial h_3}} \frac{\partial h_3}{\partial z_3} \boxed{\frac{\partial z_3}{\partial w_{13}}}$$

Obtained when updating w_{35}

$\boxed{-E_i}$

$$z_5 = w_{35} h_3 + w_{45} h_4$$

w_{35}

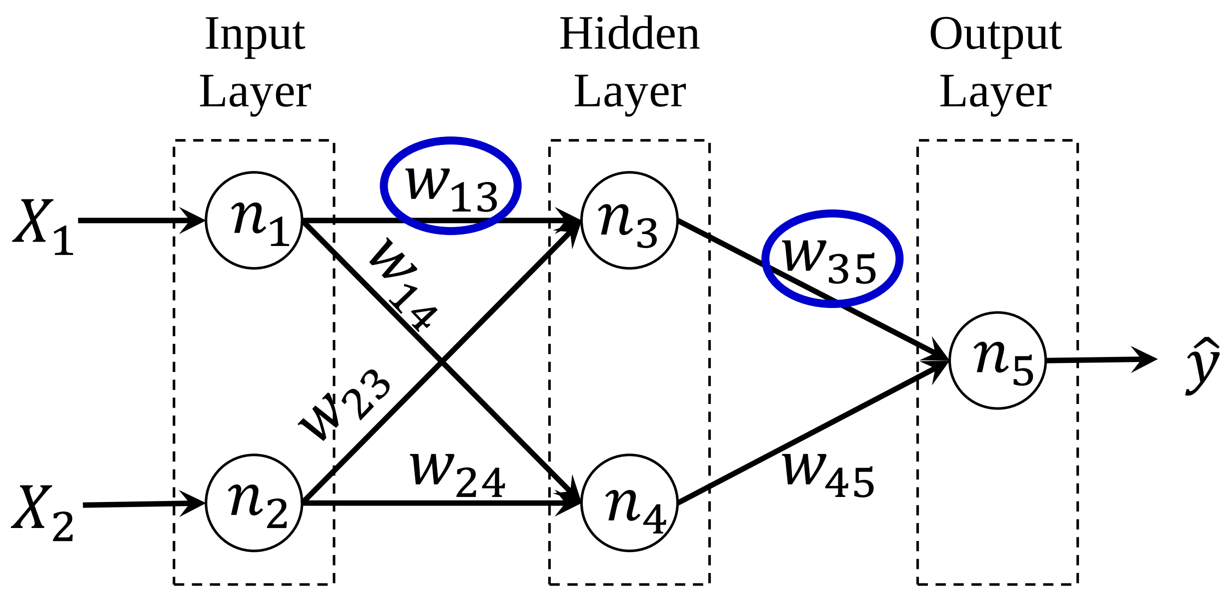
$\cancel{h_3 = sgn(z_3)}$

$h_3 = z_3$

$$\frac{\partial h_3}{\partial z_3} = 1$$

$$z_3 = w_{13} X_1 + w_{23} X_2$$

X_1



$$X_1 \xrightarrow{w_{13}} z_3 \longrightarrow h_3 \xrightarrow{w_{35}} z_5 \longrightarrow \hat{y} \longrightarrow E$$

$$w'_{23} = w_{23} + \lambda E_i w_{35} X_2$$

$$w'_{23} = w_{23} - \lambda \frac{\partial E}{\partial w_{23}} = w_{23} - \lambda \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_5} \frac{\partial z_5}{\partial h_3} \frac{\partial h_3}{\partial z_3} \frac{\partial z_3}{\partial w_{23}}$$

Obtained when updating w_{35}

$-E_i$

$$z_5 = w_{35}h_3 + w_{45}h_4$$

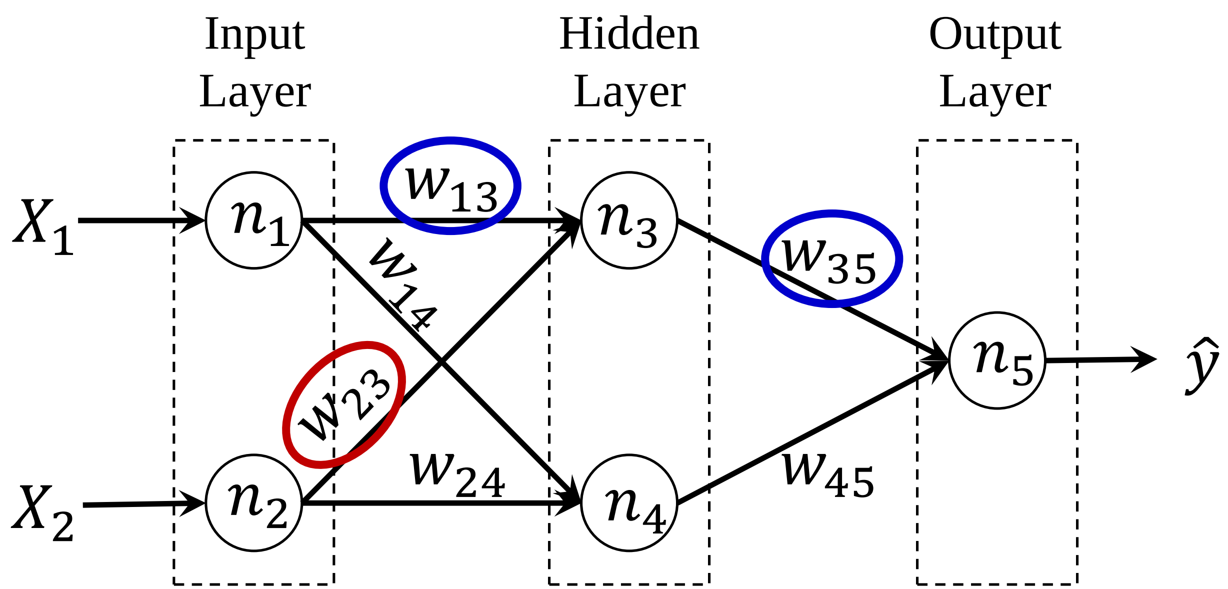
w_{35}

$\cancel{h_3 = \mathrm{sgn}(z_3)}$

$h_3 = z_3$ $\qquad \frac{\partial h_3}{\partial z_3} = 1$

$$z_3 = w_{13}X_1 + w_{23}X_2$$

X_2

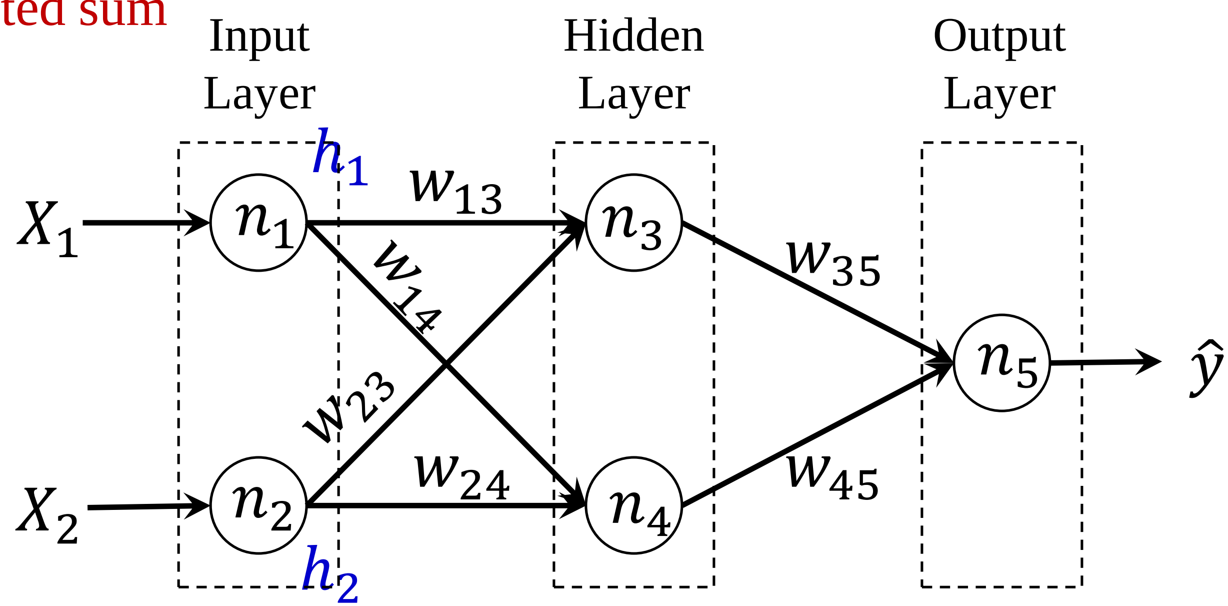


$$\begin{array}{ccccccccccc} & & w_{23} & & & & w_{35} & & & & \\ X_2 & \rightarrow & z_3 & \rightarrow & h_3 & \rightarrow & z_5 & \rightarrow & \hat{y} & \rightarrow & E \end{array}$$

BP Algorithm: Example

Activation function: sign()
Integration function: weighted sum
$\lambda = 0.4, \; \theta = 0$

X_1	X_2	y
0	0	-1
1	0	1
0	1	1
1	1	1



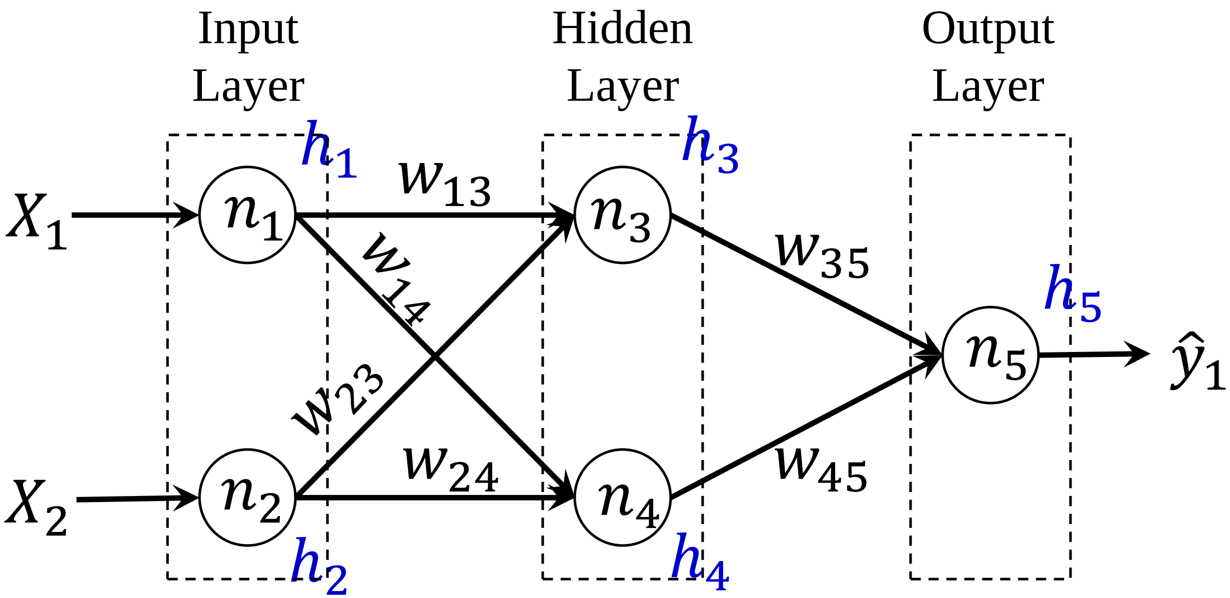
- **Initialization:**

  $(w_{13} = 1, w_{14} = 1, w_{23} = 1, w_{24} = 1, w_{35} = 1, w_{45} = 1)$

  For the 1st example: $h_1 = 0$ and $h_2 = 0$

BP Algorithm: Example (cont.)

X_1	X_2	y
0	0	-1
1	0	1
0	1	1
1	1	1



Forward pass:

$h_3 = \text{sign}(0 \times 1 + 0 \times 1) = 1$ and $h_4 = \text{sign}(0 \times 1 + 0 \times 1) = 1$

Then $\hat{y}_1 = h_5 = \text{sign}(1 \times 1 + 1 \times 1) = 1$

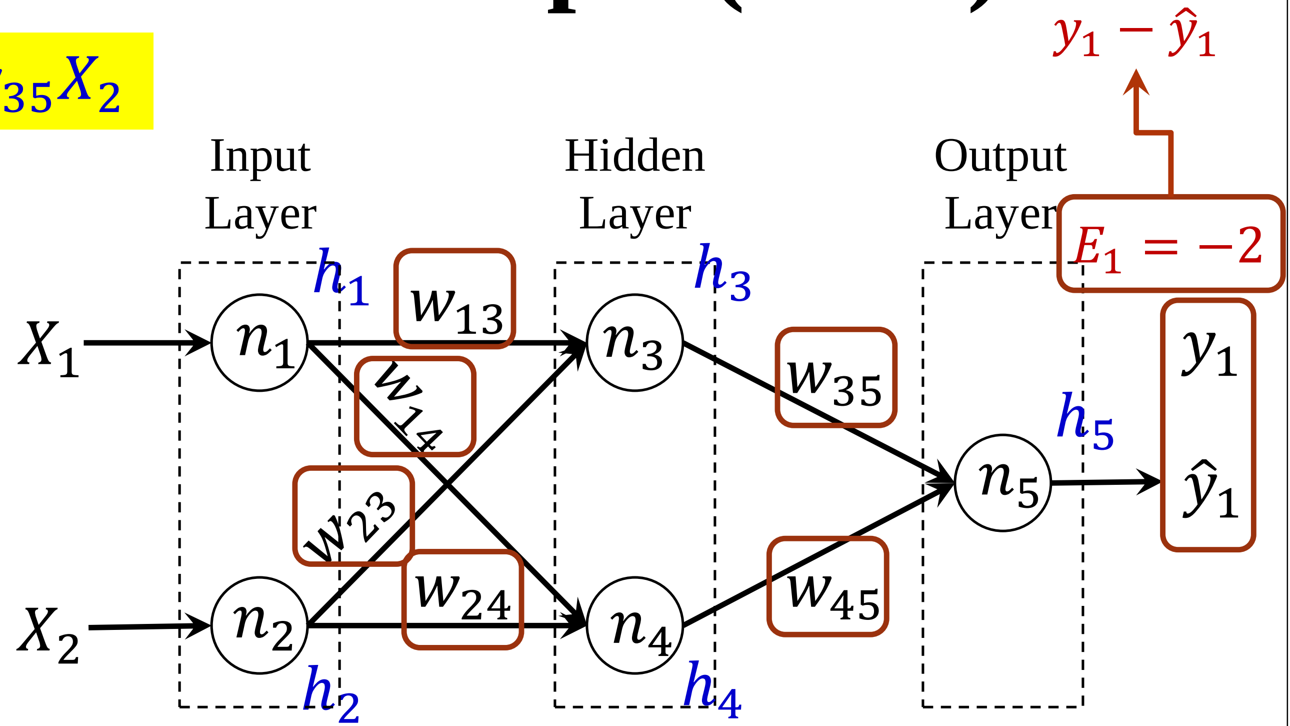
$w'_{13} = w_{13} + \lambda E_1 w_{35} X_1$

$w'_{35} = w_{35} + \lambda E_1 h_3$

BP Algorithm: Example (cont.)

$w'_{23} = w_{23} + \lambda E_i w_{35} X_2$

X_1	X_2	y
0	0	-1
1	0	1
0	1	1
1	1	1



Backpropagation:

$w_{35} = 1 + 0.4 \times (-2) \times 1 = 0.2$

$w_{13} = 1 + 0.4 \times (-2) \times 1 \times 0 = 1$

$w_{23} = 1 + 0.4 \times (-2) \times 1 \times 0 = 1$

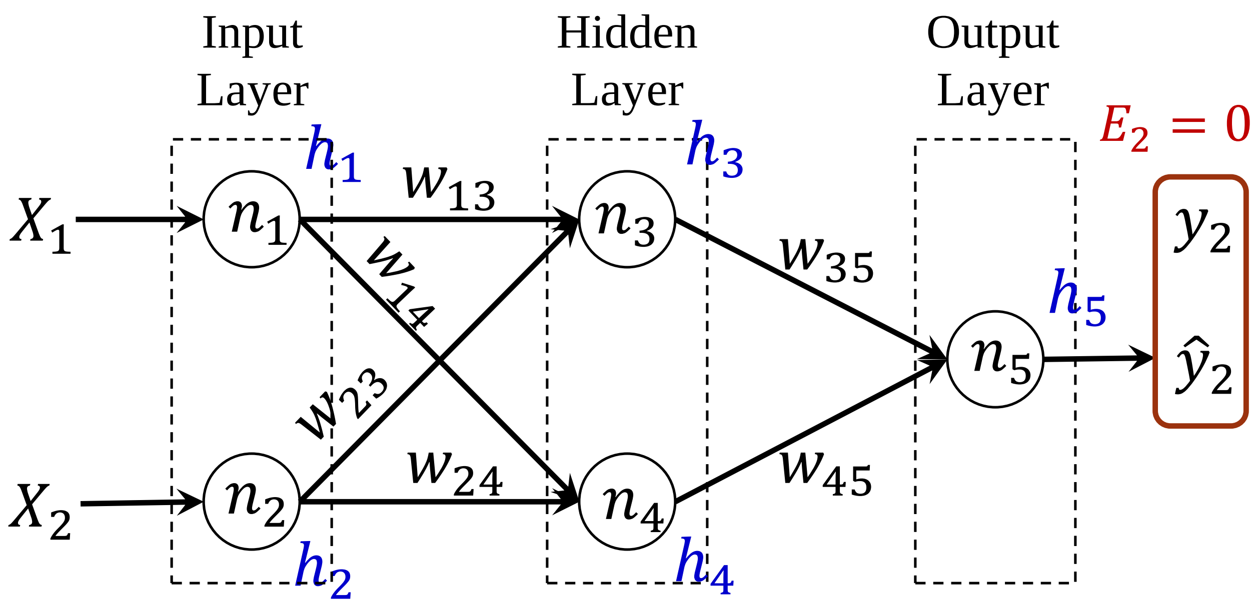
$w_{45} = 1 + 0.4 \times (-2) \times 1 = 0.2$

$w_{14} = 1 + 0.4 \times (-2) \times 1 \times 0 = 1$

$w_{24} = 1 + 0.4 \times (-2) \times 1 \times 0 = 1$

BP Algorithm: Example (cont.)

X_1	X_2	y
0	0	-1
1	0	1
0	1	1
1	1	1



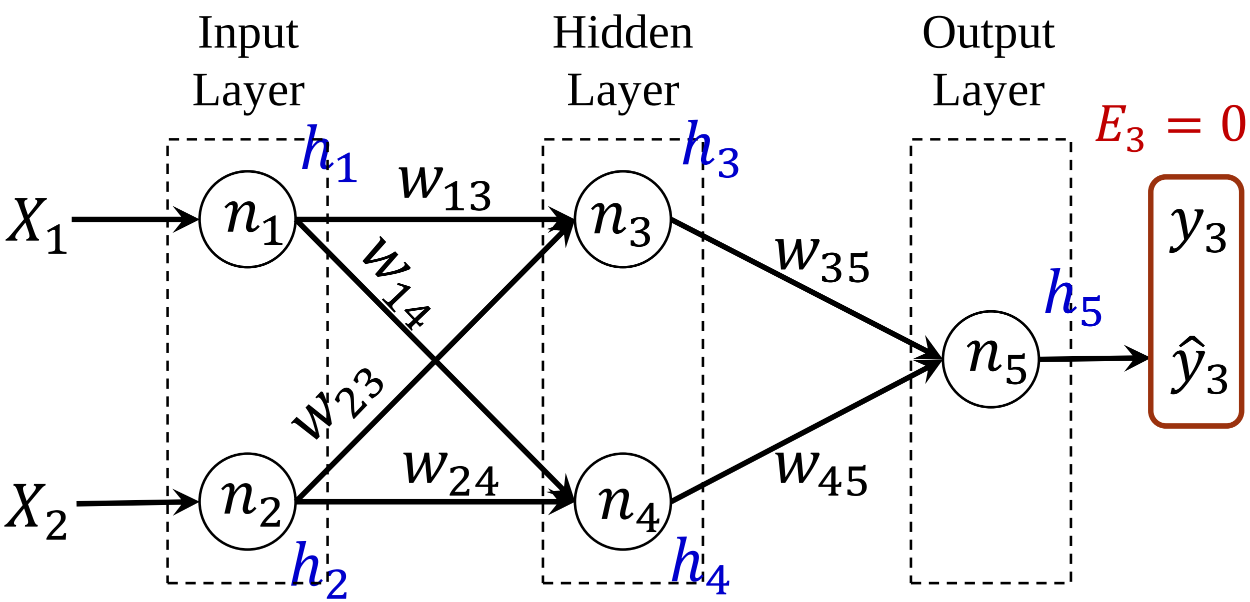
For the 2nd example: $h_1 = 1$ and $h_2 = 0$

$h_3 = \text{sign}(1 \times 1 + 0 \times 1) = 1$ and $h_4 = \text{sign}(1 \times 1 + 0 \times 1) = 1$

Then $\hat{y}_2 = h_5 = \text{sign}(1 \times 0.2 + 1 \times 0.2) = 1$

BP Algorithm: Example (cont.)

X_1	X_2	y
0	0	-1
1	0	1
0	1	1
1	1	1



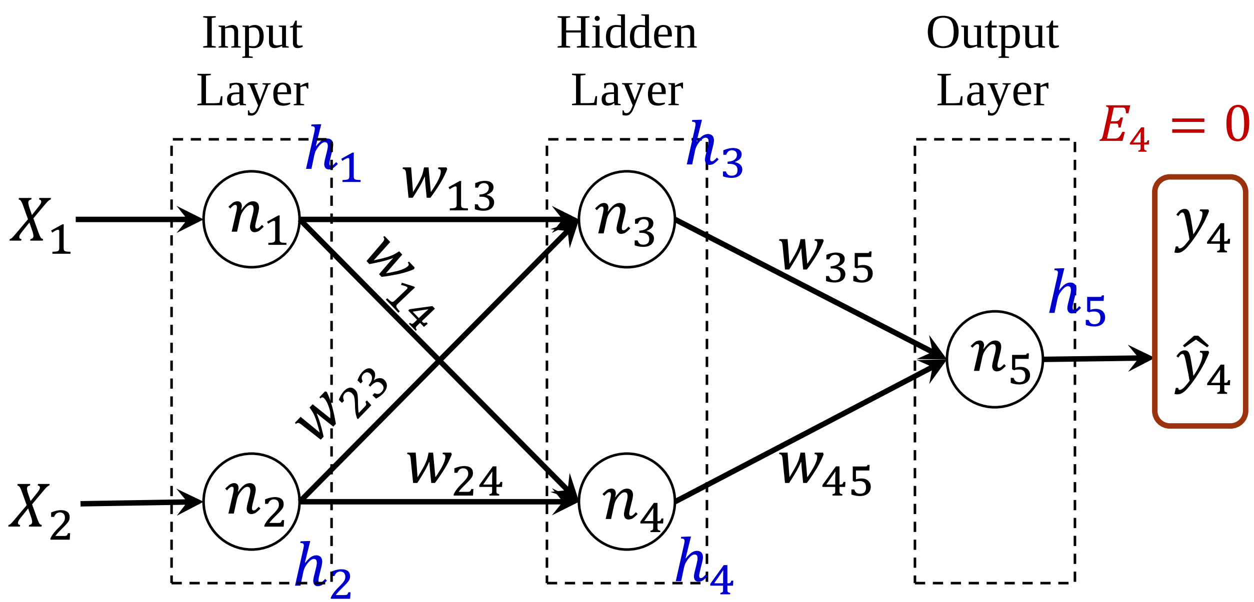
For the 3rd example: $h_1 = 0$ and $h_2 = 1$

$h_3 = \text{sign}(0 \times 1 + 1 \times 1) = 1$ and $h_4 = \text{sign}(0 \times 1 + 1 \times 1) = 1$

Then $\hat{y}_3 = h_5 = \text{sign}(1 \times 0.2 + 1 \times 0.2) = 1$

BP Algorithm: Example (cont.)

X_1	X_2	y
0	0	-1
1	0	1
0	1	1
1	1	1



For the 4th example: $h_1 = 1$ and $h_2 = 1$

$h_3 = \text{sign}(1 \times 1 + 1 \times 1) = 1$ and $h_4 = \text{sign}(1 \times 1 + 1 \times 1) = 1$

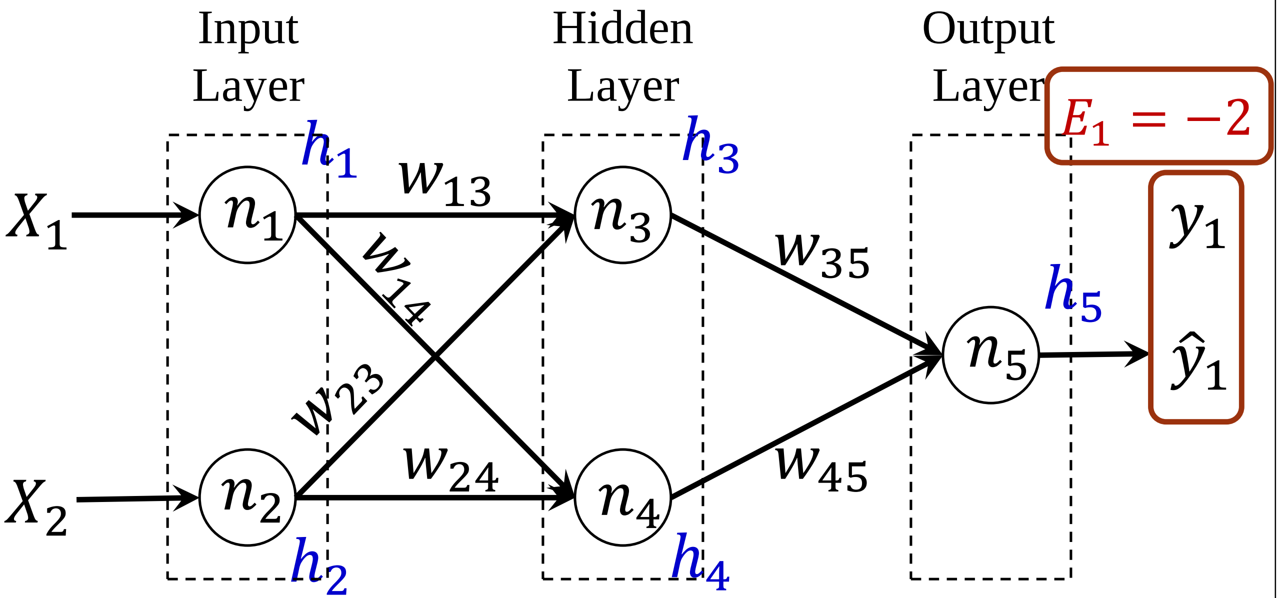
Then $\hat{y}_4 = h_5 = \text{sign}(1 \times 0.2 + 1 \times 0.2) = 1$

End of the 1st Epoch

The 2nd Epoch starts

BP Algorithm: Example (cont.)

X_1	X_2	y
0	0	-1
1	0	1
0	1	1
1	1	1



For the 1st example again: $h_1 = 0$ and $h_2 = 0$

$h_3 = \text{sign}(0 \times 1 + 0 \times 1) = 1$ and $h_4 = \text{sign}(0 \times 1 + 0 \times 1) = 1$

Then $\hat{y}_1 = h_5 = \text{sign}(0.2 \times 1 + 0.2 \times 1) = 1$

Weights need to be further updated via backpropagation

Design Issues for ANN

- The number of nodes in the input layer
  - Assign an input node to each numerical or binary input variable
- The number of nodes in the output layer
  - Binary class problem → single node
  - C-class problem → C output nodes

Thank you!