Deep Learning

Single Perceptron Architecture
Single Perceptron Capabilities
Single Perceptron Learning

Machine Learning – Ex.1

(1) x=3 → y=5

(2) x=10 → y=19

(3) x=-1 → y=-3

x=7 → y=?

Machine Learning – Ex.2

(1) $x_1 = 3,\ \ x_2 = 3 \ \ \rightarrow\ \ y = 6$

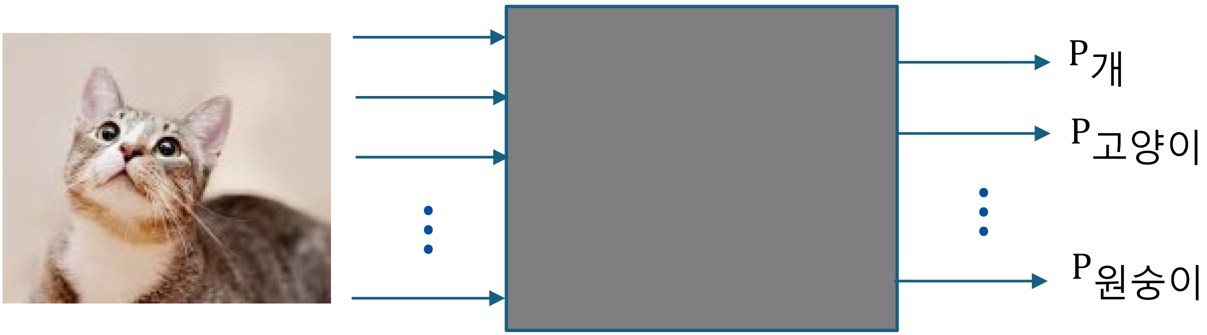
(2) $x_1 = 1,\ \ x_2 = 2 \ \ \rightarrow\ \ y = -1$

(3) $x_1 = 0,\ \ x_2 = 0 \ \ \rightarrow\ \ y = 0$

(4) $x_1 = 5,\ \ x_2 = -1 \ \ \rightarrow\ \ y = 26$

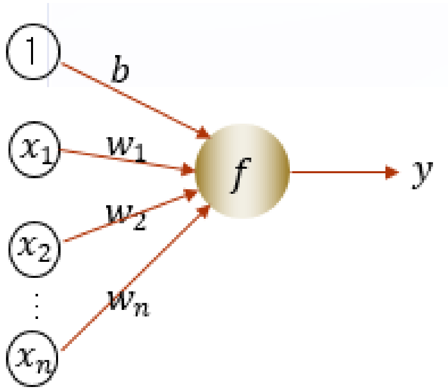
$x_1 = 100,\ \ x_2 = 200 \ \ \rightarrow\ \ y = ?$

Machine Learning – Ex.3



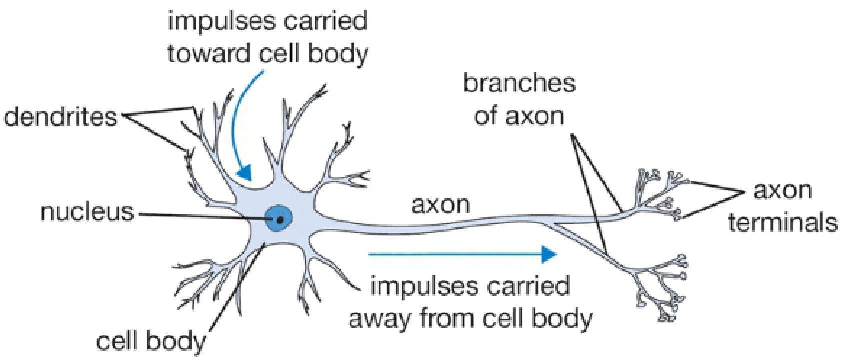
Single Perceptron – Contents

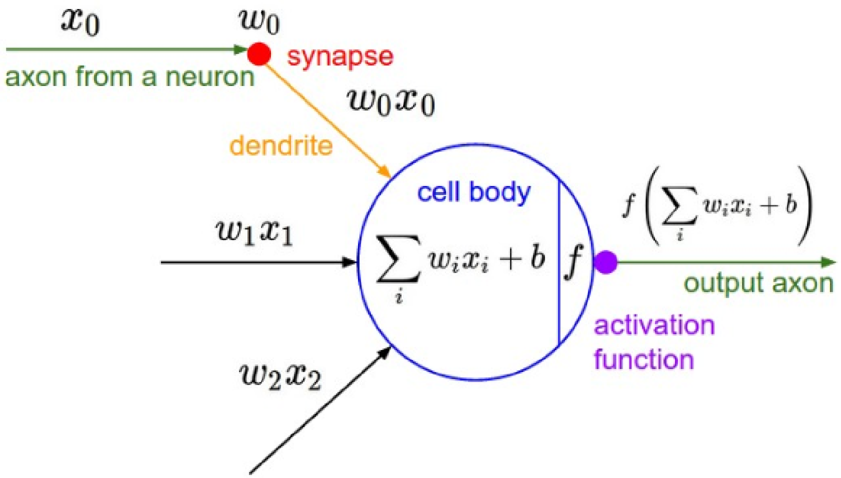
- Architecture
- Capability
- Learning



Perceptron – Architecture

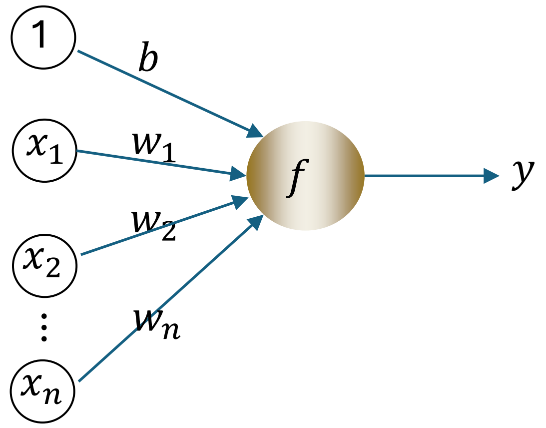
- **Biological neuron vs Artificial neuron**





Perceptron – Architecture

- **Architecture**

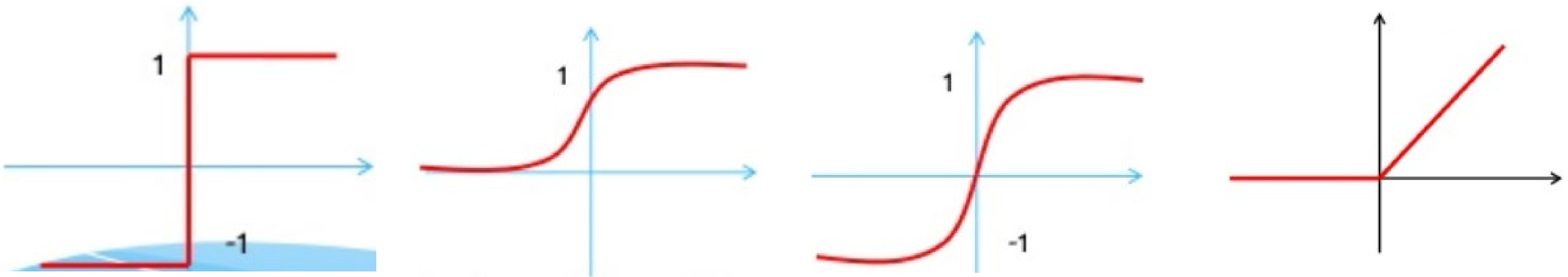


- input: $\mathbf{x}=[x_1, x_2, \dots, x_n]^T$
- weights: $\mathbf{w}=[w_1, w_2, \dots, w_n]^T$
- bias: b
- activation function: f
- output: $y = f(\mathbf{w}^\mathrm{T}\mathbf{x} + b) = f(w_1x_1 + w_2x_2 + \cdots + w_nx_n + b)$

Activation Functions

- **Activation functions for $y = f(\mathbf{w}^\mathsf{T}\mathbf{x} + b) = f(z)$**

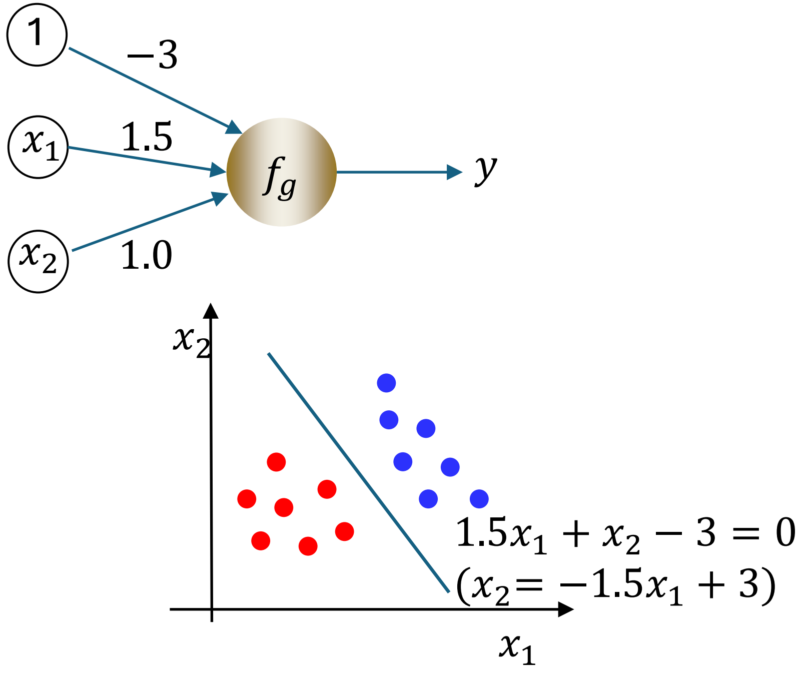
  - **sign:** $f_g(z) = \begin{cases} +1, & z \geq 0 \\ -1, & z < 0 \end{cases}$
  - **sigmoid:** $f_s(z) = \frac{1}{1+e^{-z}},\ 0 \leq f \leq 1$
  - **tanh:** $f_t(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}},\ -1 \leq f \leq 1$
  - **ReLU:** $f_R(z) = max(0, z),\ 0 \leq f \leq \infty$

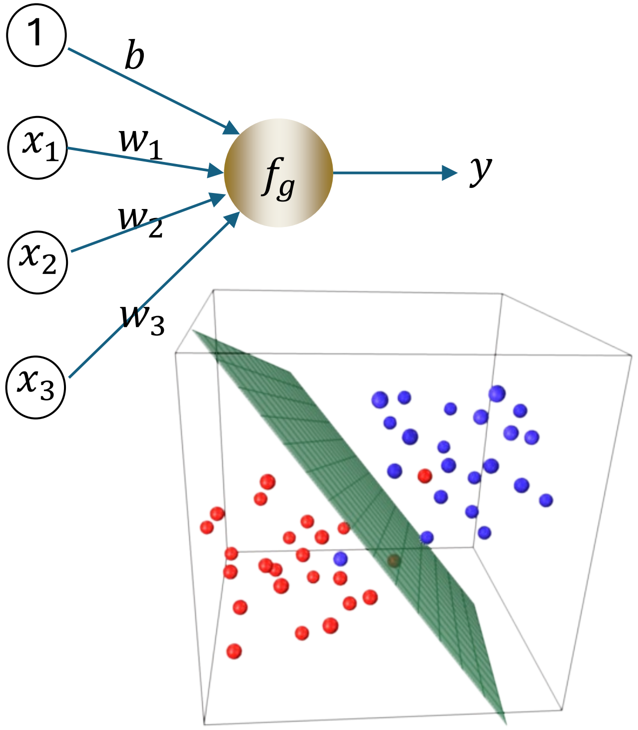


sign **sigmoid** **tanh** **ReLU**

Perceptron - Capability

- **Single Perceptron with f_g represents a linear classifier in R^n**
  - $y = f_g(\mathbf{w}^\mathsf{T}\mathbf{x} + b) = \begin{cases} +1, & \mathbf{w}^\mathsf{T}\mathbf{x} + b \geq 0 \\ -1, & \mathbf{w}^\mathsf{T}\mathbf{x} + b < 0 \end{cases}$

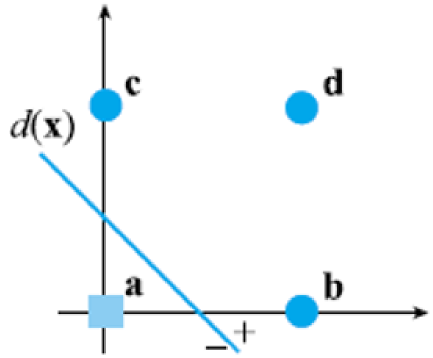


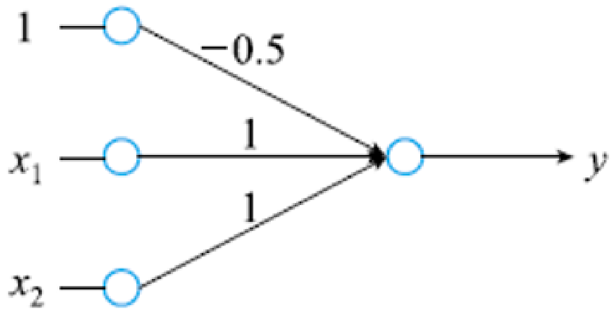


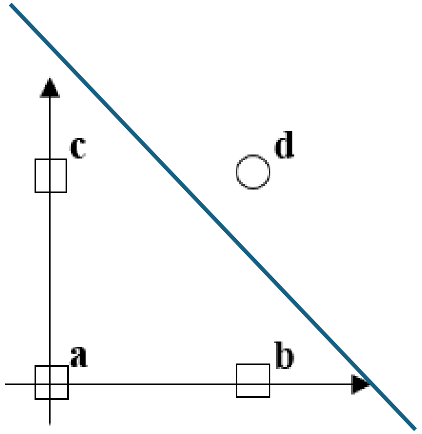
$w_1x_1 + w_2x_2 + w_3x_3 + b = 0$

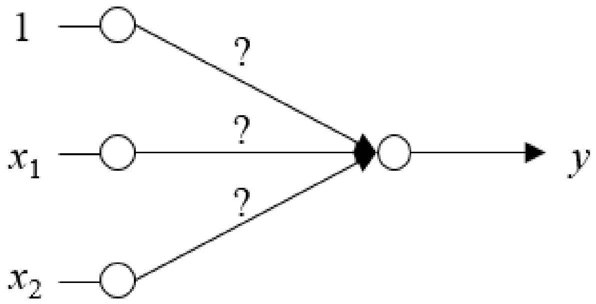
Perceptron - Capability

- **Logic functions: AND, OR, etc**



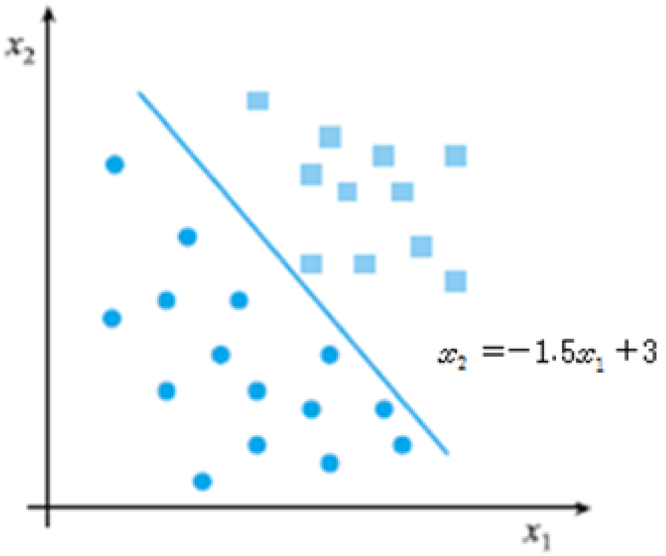


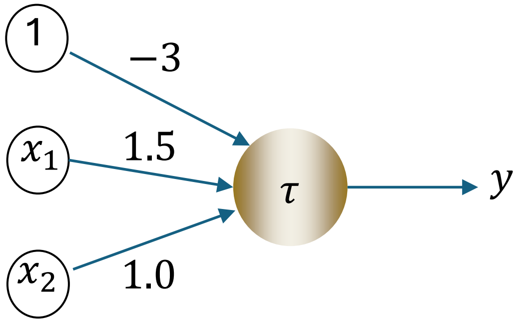




Perceptron - Exercise

Design a single perceptron for the following decision boundary in a 2-D space: $x_2 = -1.5x_1 + 3$. Here, the activation function for the perceptron is a step function, $\tau(s)$

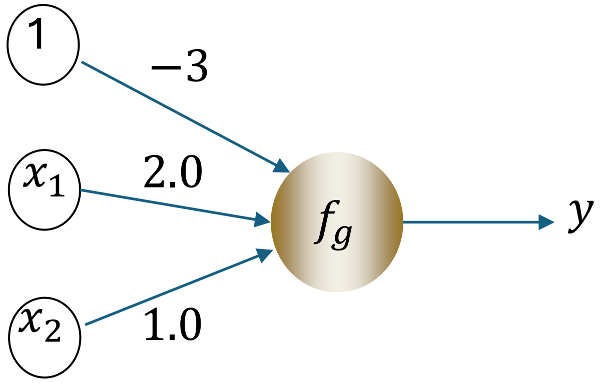


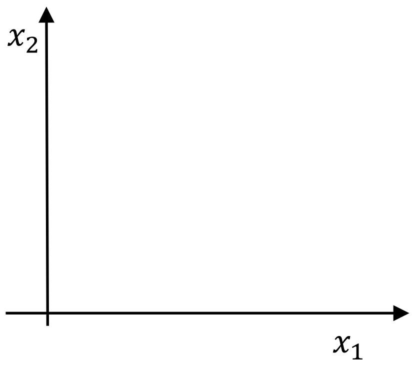


$$\tau(s) = \begin{cases} +1, & s \geq 0 \\ -1, & s < 0 \end{cases}$$

Perceptron - Exercise

- **Single Perceptron with f_g represents a linear classifier(or line)**
  - $y = f_g(\mathbf{w}^\mathsf{T}\mathbf{x} + b) = \begin{cases} +1, & \mathbf{w}^\mathsf{T}\mathbf{x} + b \geq 0 \\ -1, & \mathbf{w}^\mathsf{T}\mathbf{x} + b < 0 \end{cases}$
  - What is the line for the following perceptron?

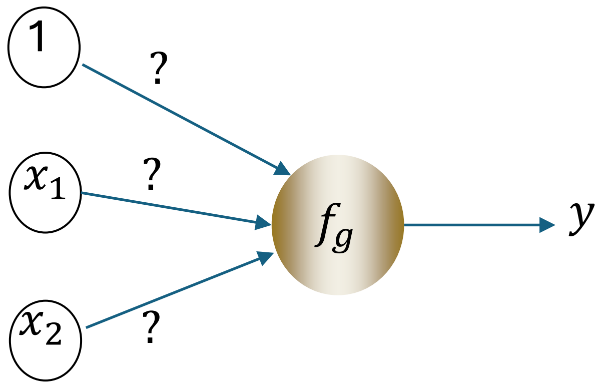


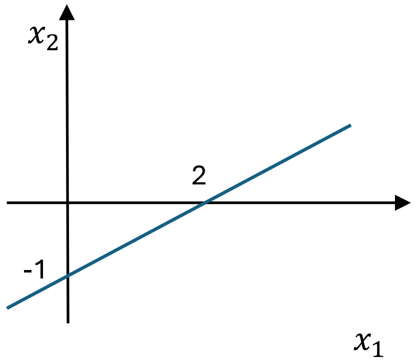


- What is the output value for the input x1 =(-1, 0), and x2=(2, 0)?

Perceptron - Exercise

- **Single Perceptron with f_g represents a linear classifier(or line)**
  - $y = f_g(\mathbf{w}^\mathsf{T}\mathbf{x} + b) = \begin{cases} +1, & \mathbf{w}^\mathsf{T}\mathbf{x} + b \geq 0 \\ -1, & \mathbf{w}^\mathsf{T}\mathbf{x} + b < 0 \end{cases}$
  - What is the perceptron for the following line?





- What is the output value for the input x1 =(-1, 0), and x2=(0 -2)?

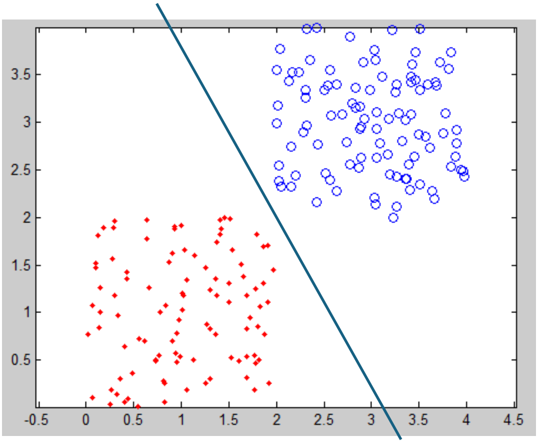
Perceptron – Learning

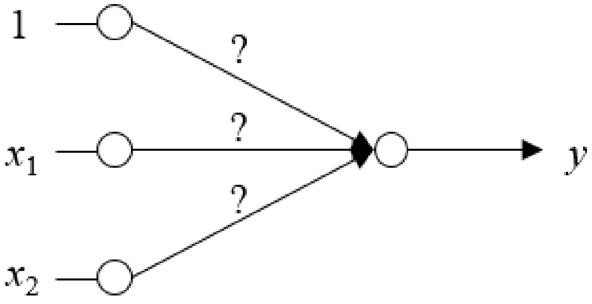
- **Problem definition: given a set of N training samples,**

  $\mathbf{X}$={ ($\mathbf{x}_1$, t_1), ($\mathbf{x}_2$, t_2), …, ($\mathbf{x}_N$, t_N) },

  where $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]^T$ and $t_i \in \{+1, -1\}$

☞ **find *a possible* $\mathbf{w}$=$[w_1, w_2, \dots, w_n]^T$ and b which separates samples with $t = +1$ from samples with $t = -1$**

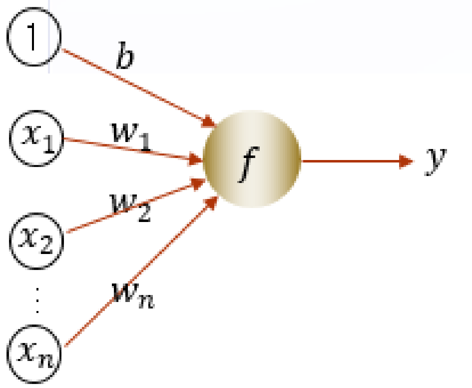




Perceptron – Learning

- **Given a $(\mathbf{x}_k, t_k)$, test the goodness for $\theta = \{\mathbf{w}, b\}$**

$t_k = +1$	$\mathrm{w}^T \mathrm{x}_k + b \geq 0$	$y = +1$	correct
	$\mathrm{w}^T \mathrm{x}_k + b < 0$	$y = -1$	error
$t_k = -1$	$\mathrm{w}^T \mathrm{x}_k + b \geq 0$	$y = +1$	error
	$\mathrm{w}^T \mathrm{x}_k + b < 0$	$y = -1$	correct



(*) assume $f = f_g$

- **Cost(Error) function for the goodness of $\theta = \{\mathbf{w}, b\}$**

$$J(\theta) = \sum_{\mathbf{x}_k \in Y} (-t_k)\,(\mathbf{w}^\mathsf{T}\mathbf{x}_k + b)$$

where Y is set of mis-classified samples

Perceptron – Learning

- **Gradient Descent Algorithm to find the optimal $\theta = \{\mathbf{w}, b\}$**

  - **initialize θ as $\theta(0)$ randomly**
  - **iterates**

$$\theta(t+1) = \theta(t) - \rho \frac{\partial J}{\partial \theta}$$

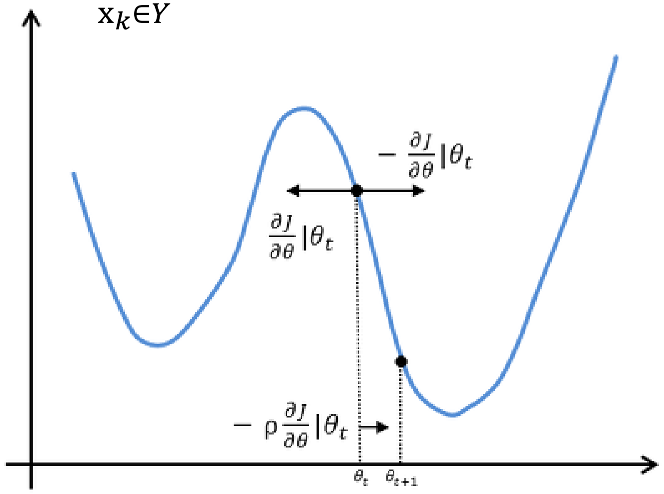
☞ $$\mathbf{w}(t+1) = \mathbf{w}(t) - \rho \sum_{\mathbf{x}_k \in Y} (-t_k)\,(\mathbf{x}_k)$$

☞ $$b(t+1) = b(t) - \rho \sum_{\mathbf{x}_k \in Y} (-t_k)$$

where ρ is a learning rate ($0 < \rho \ll 1$), and $\dfrac{\partial J}{\partial \theta}$ is the gradient

  - **until convergence to a local minimum**

$$J(\theta) = \sum_{\mathrm{x}_k \in Y} (-t_k)(\mathbf{w}^T \mathbf{x}_k + b)$$



Perceptron – Learning

Algorithm [4.1]

Perceptron Learning (Batch Mode)

Input: Training set $X = \{(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)\}$, learning rate ρ

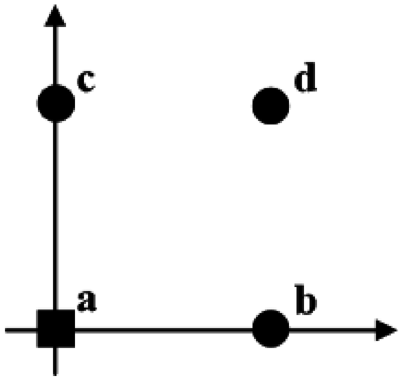
Output: Perceptron parameters w, b

Algorithm:

1. Initialize w and b.
2. **repeat** {
3. $\quad Y = \emptyset$;
4. $\quad$ **for** $i = 1$ to N **do** {
5. $\quad\quad y = \tau(w^T x_i + b)$ // Perform classification using (4.2)
6. $\quad\quad$ **if** $y \neq t_i$ **then** $Y = Y \cup x_i$; // Collect misclassified samples
7. $\quad$ }
8. $\quad w = w + \rho \sum_{x_k \in Y} t_k x_k$; // Update parameters using (4.7)
9. $\quad b = b + \rho \sum_{x_k \in Y} t_k$;
10. } **until** $Y = \emptyset$;
11. Store w and b.

Source: Pattern Recognition (Oil Seok, Kyobo Bookstore, 2008)

Perceptron Learning Example (1)



Training Samples

$\mathbf{a}=(0,0)^T$, $t_a=-1$

$\mathbf{b}=(1,0)^T$, $t_b=1$

$\mathbf{c}=(0,1)^T$, $t_c=1$

$\mathbf{d}=(1,1)^T$, $t_d=1$

Initialization

$\rho=0.4$

$\mathbf{w}(0)=(-0.5,0.75)^T$, $b(0)=0.375$ ①

Error Patterns

$d(\mathbf{x})=-0.5x_1+0.75x_2+0.375$

$Y=\{\mathbf{a}, \mathbf{b}\}$

$d(\mathbf{x})=-0.1x_1+0.75x_2+0.375$

$Y=\{\mathbf{a}\}$

Weight Updates

$$\mathbf{w}(1)=\mathbf{w}(0)+0.4(t_a\cdot\mathbf{a}+t_b\cdot\mathbf{b})=\begin{pmatrix}-0.5\\0.75\end{pmatrix}+0.4\left[-\begin{pmatrix}0\\0\end{pmatrix}+\begin{pmatrix}1\\0\end{pmatrix}\right]=\begin{pmatrix}-0.1\\0.75\end{pmatrix}$$ ②

$$b(1)=b(0)+0.4(t_a+t_b)=0.375+0.4*0=0.375$$

$$\mathbf{w}(2)=\mathbf{w}(1)+0.4(t_a\mathbf{a})=\begin{pmatrix}-0.1\\0.75\end{pmatrix}+0.4\left[-\begin{pmatrix}0\\0\end{pmatrix}\right]=\begin{pmatrix}-0.1\\0.75\end{pmatrix}$$ ③

$$b(2)=b(1)+0.4(t_a)=0.375-0.4=-0.025$$

Perceptron Learning Example (2)

Misclassification pattern set

$d(\mathbf{x})$=-0.1x_1+0.75x_2-0.025

Y={**b**}

$d(\mathbf{x})$=-0.3x_1+0.75x_2+0.375

Y={**a**}

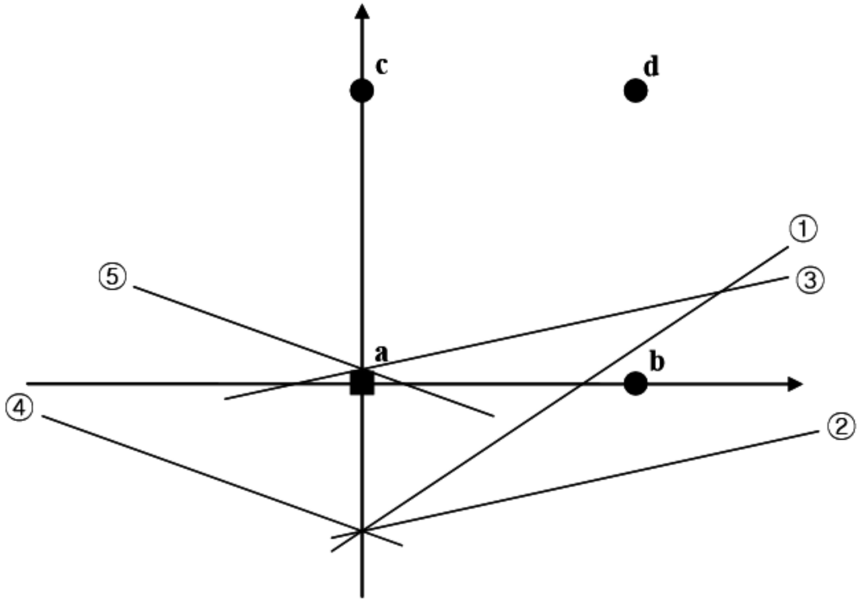
Weight update

$$\mathbf{w}(3)=\mathbf{w}(2)+0.4(t_b\mathbf{b})=\begin{pmatrix}-0.1\\0.75\end{pmatrix}+0.4\left[\begin{pmatrix}1\\0\end{pmatrix}\right]=\begin{pmatrix}0.3\\0.75\end{pmatrix} \quad ④$$

$$b(3)=b(2)+0.4(t_b)=-0.025+0.4=0.375$$

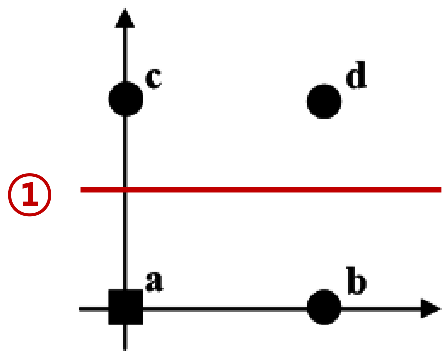
$$\mathbf{w}(4)=\mathbf{w}(3)+0.4(t_a\mathbf{a})=\begin{pmatrix}0.3\\0.75\end{pmatrix}+0.4\left[-\begin{pmatrix}0\\0\end{pmatrix}\right]=\begin{pmatrix}0.3\\0.75\end{pmatrix} \quad ⑤$$

$$b(4)=b(3)+0.4(t_a)=0.375-0.4=-0.025$$



Perceptron Learning - Exercise

☞ Do the previous calculations again with another initialization:



Training Samples

$\mathbf{a}=(0,0)^T$, $t_a=-1$

$\mathbf{b}=(1,0)^T$, $t_b=1$

$\mathbf{c}=(0,1)^T$, $t_c=1$

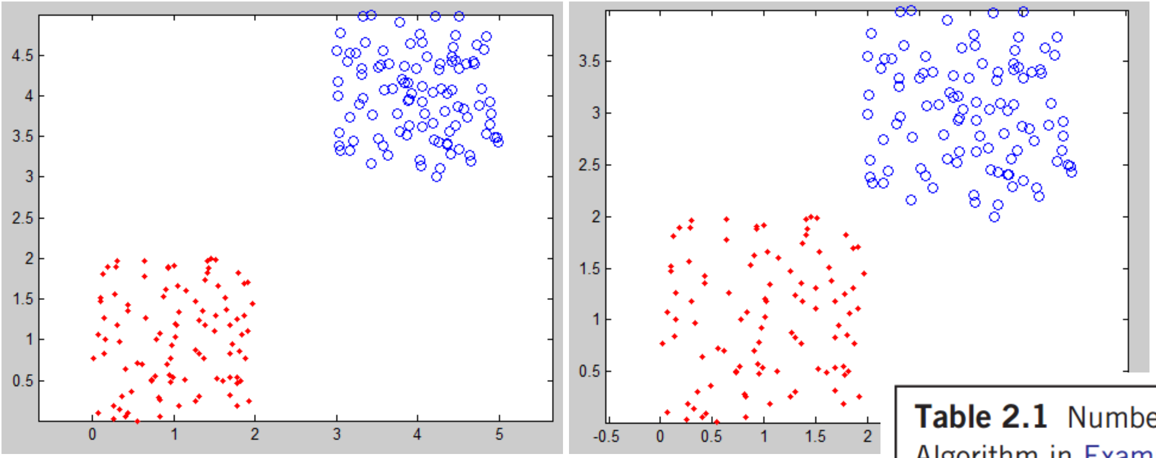
$\mathbf{d}=(1,1)^T$, $t_d=1$

Initialization

$\rho = 0.4$

$\mathrm{w}(0) = (0, -1.0),\ b(0) = 0.5$ ①

Perceptron Learning Example (3)



X1 X2

X3 X4

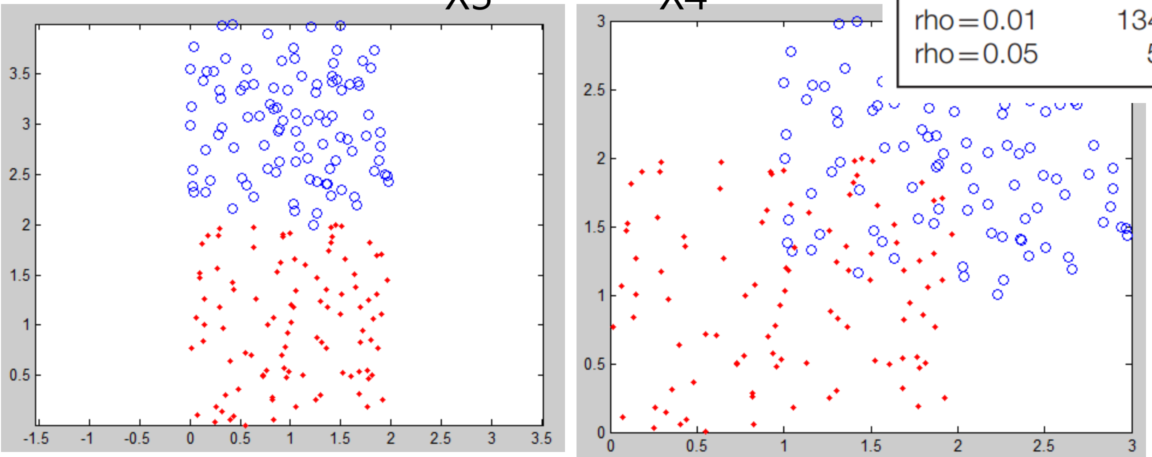


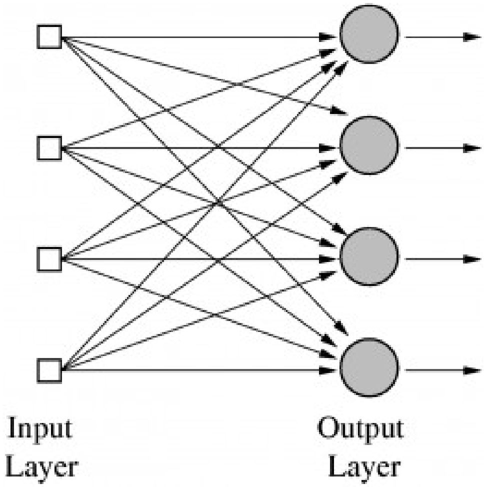
Table 2.1 Number of Iterations Performed by the Perceptron Algorithm in Example 2.2.1

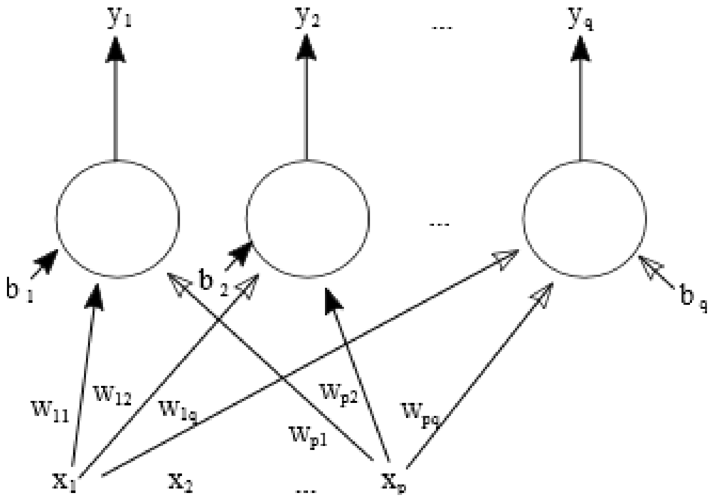
	X_1	X_2	X_3	X_4
rho=0.01	134	134	5441	No convergence
rho=0.05	5	5	252	No convergence

출처: Introduction to PR – A MATLAB Approach(S. Theodoritis et al, AP, 2010)

Perceptron – Extension

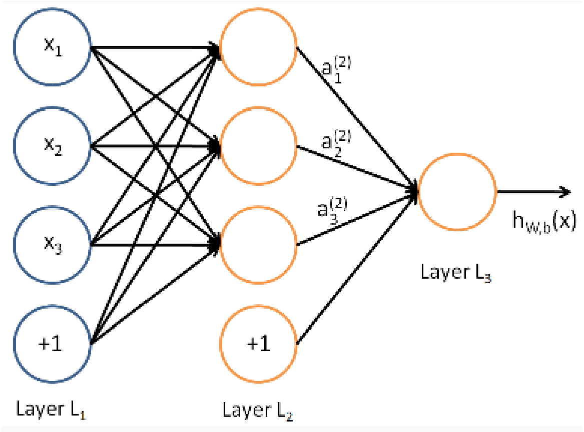
- **Single Layer Perceptron**
  - **Input layer & Output layer**
  - **linear function with p-D input and q-D output**
  - **q linear functions in p-Dimension**



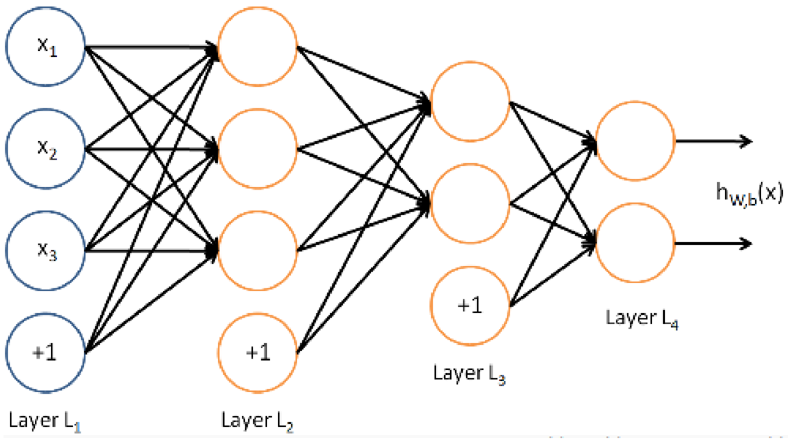


Perceptron – Extension

- **Multi-Layer Perceptron**
  - input layer, hidden layer(s), output layer
  - feed-forward
  - fully-connected



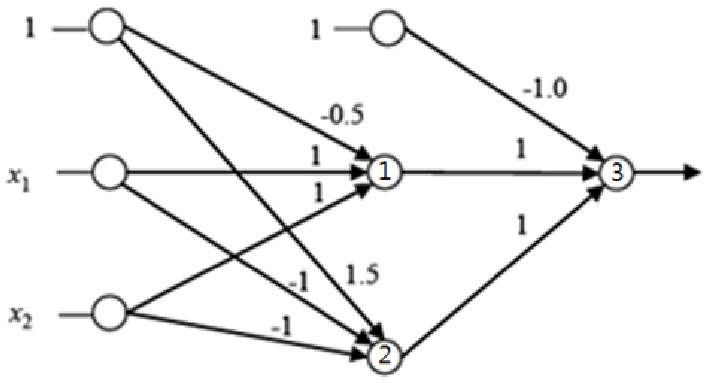
3×3×1 MLP



3×3×2×2 MLP

Exercise

Given the following MLP(Multi-Layer Perceptron), compute the outputs from individual neurons, i.e. neurons ①, ②, and ③, for the input pattern $x=(1,\ 1)^T$. Assume that the activation function in every neuron is a step function $\tau(s)$.



$$\tau(s) = \begin{cases} +1, & s \geq 0 \\ -1, & s < 0 \end{cases}$$

THANK YOU!