Week 3(1/3)

Artificial Neuron

Machine Learning with Python

Handong Global University
Prof. Youngsup Kim
idebtor@gmail.com

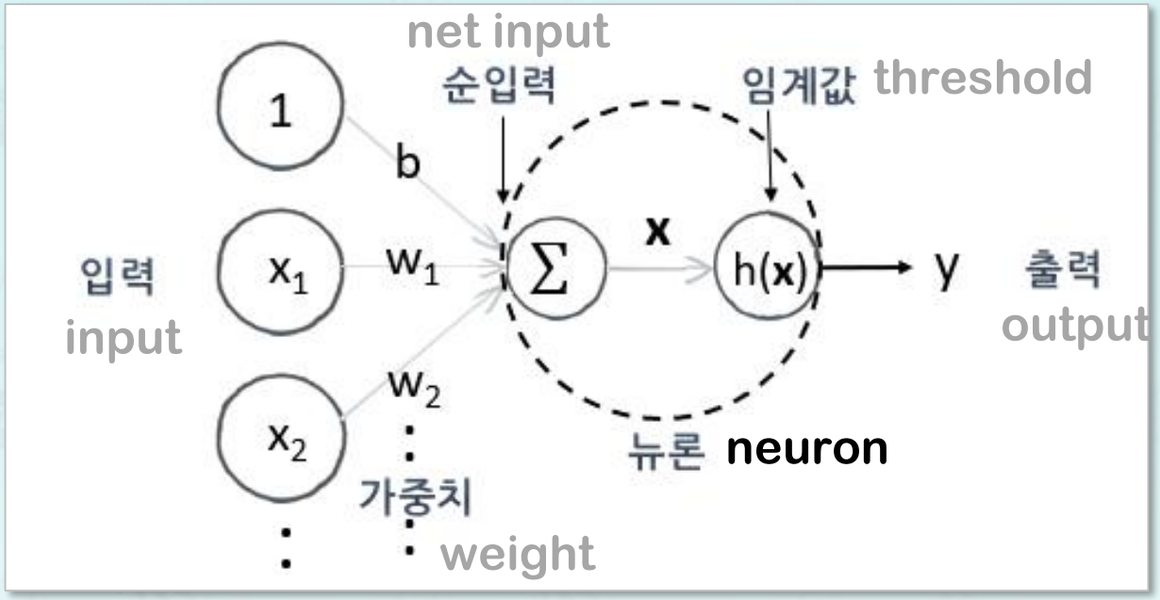
Artificial Neuron

- Goal
  - Understanding Artificial Neuron
  - Implementing AND Neuron
  - Visualizing AND Neuron

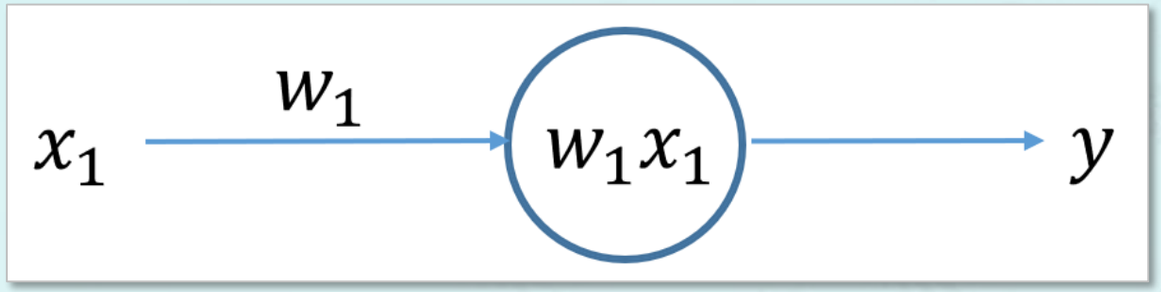
- Content
  - Artificial Neuron Concept
  - AND Gate and AND Neuron
  - AND Neuron Implementaion
  - AND Neuron Visualization

Artificial Neuron

Artificial Neuron

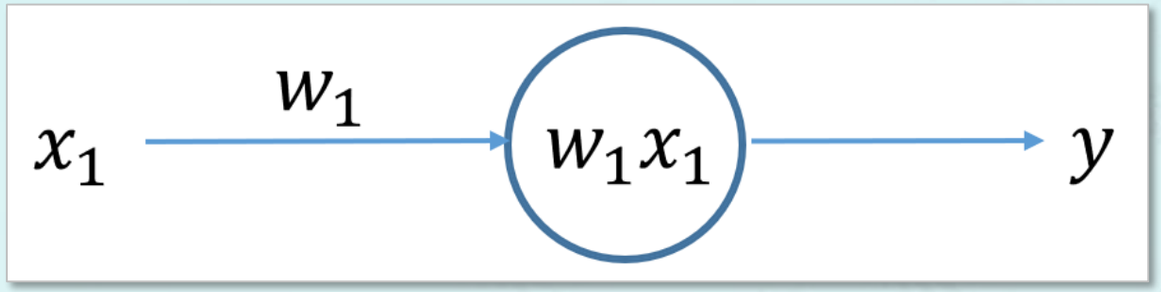


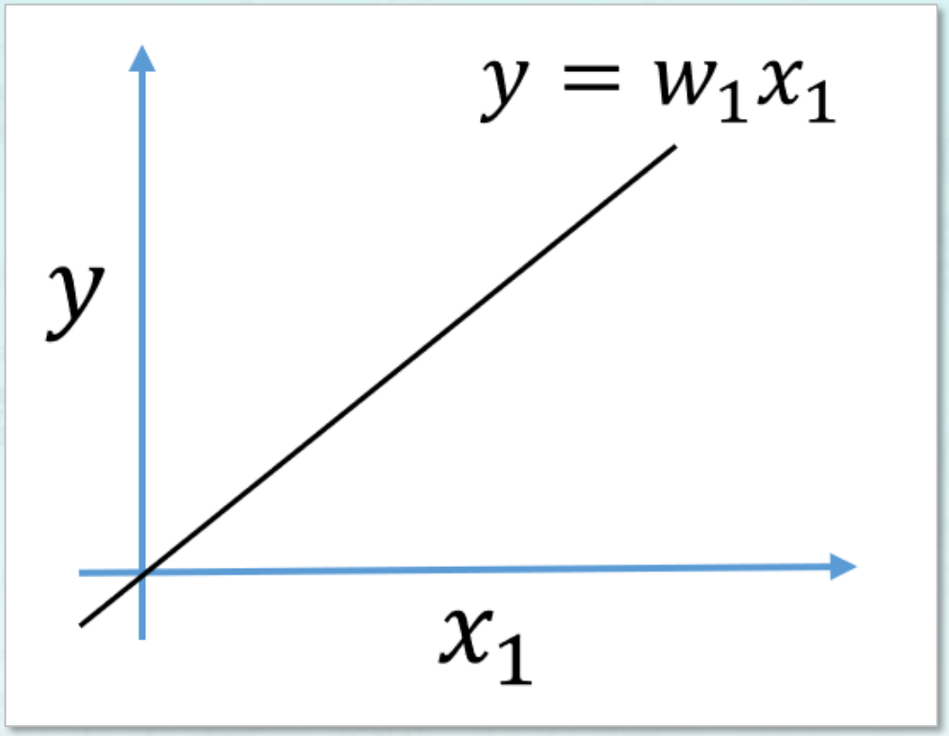
Artificial Neuron



- $y = w_1 x_1$
- $y = ax$

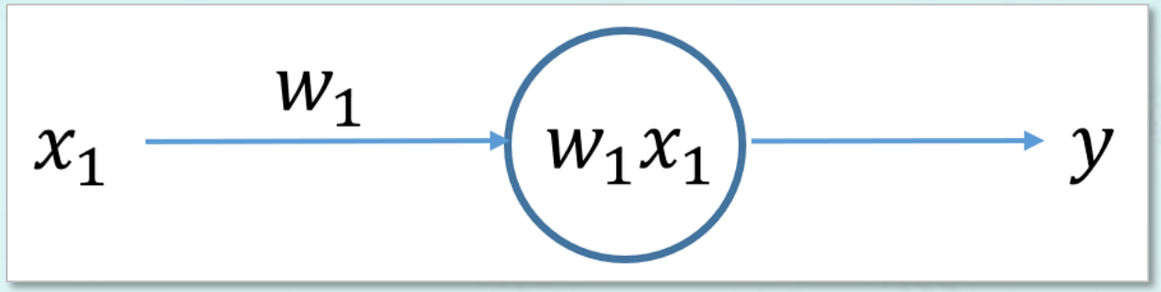
Artificial Neuron

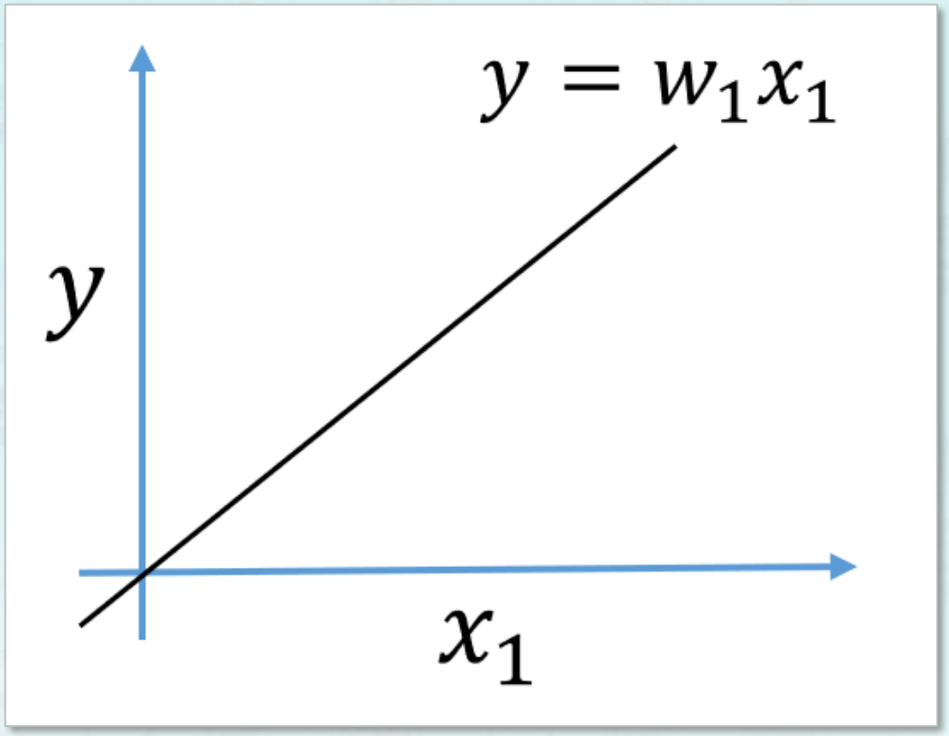




- $y = w_1 x_1$
- $y = ax$

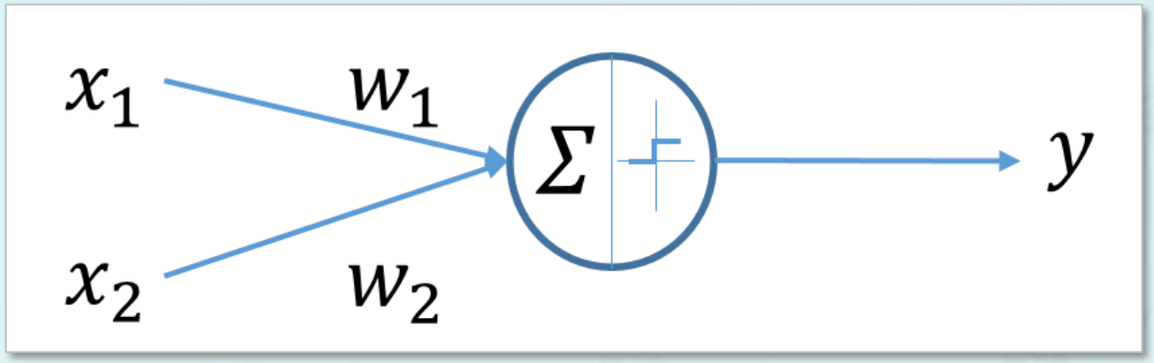
Artificial Neuron





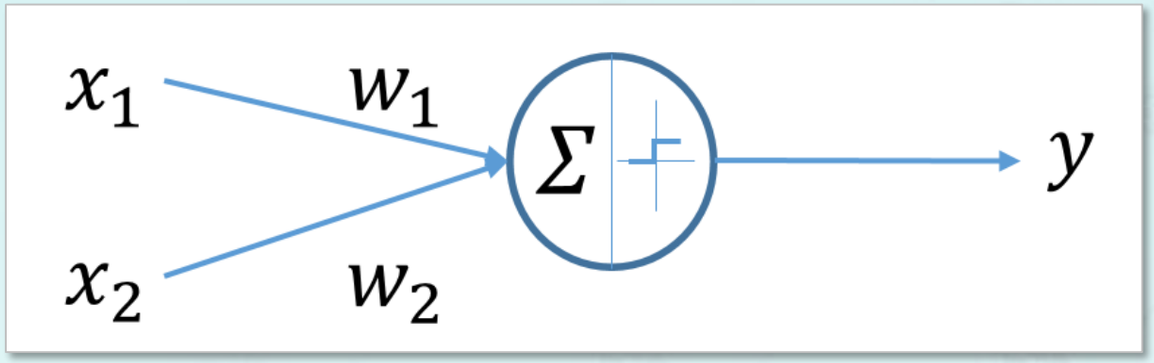
- $y = w_1 x_1$
- $y = ax$
- a → **slope**, w_1 → **weight**

Artificial Neuron



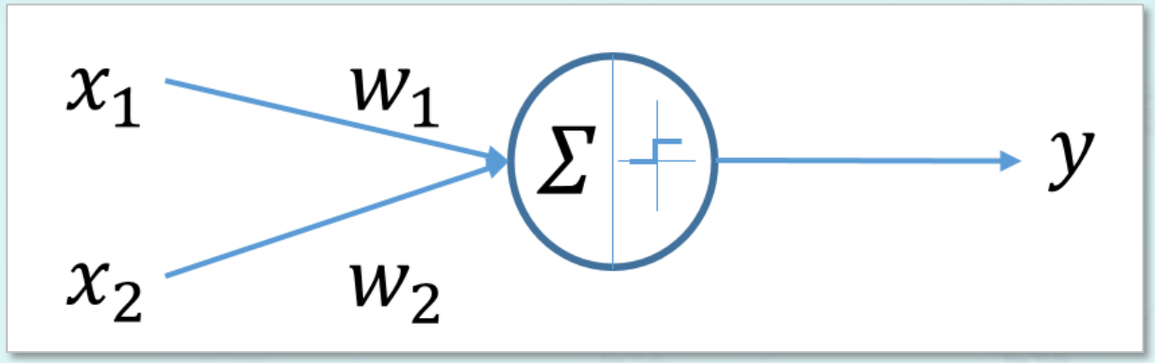
- **net input:**

Artificial Neuron



- **net input:** $w_1 x_1 + w_2 x_2$
- **threshold:** θ
- **activated:** > θ

Artificial Neuron

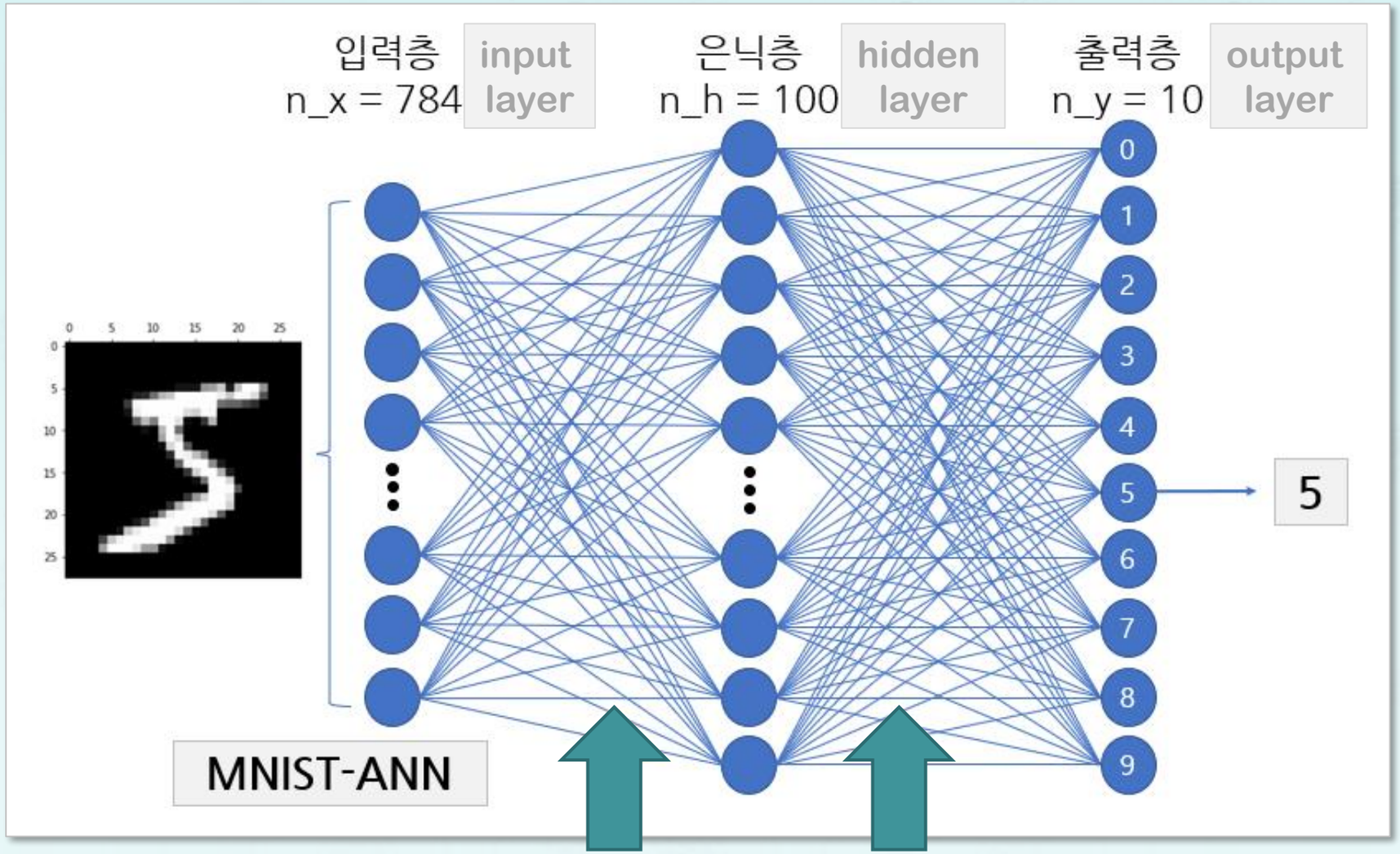


$$y = \begin{cases} 0 & \text{if } (w_1x_1 + w_2x_2 <= \theta) \\ 1 & \text{if } (w_1x_1 + w_2x_2 > \theta) \end{cases} \quad (1)$$

- **net input:** $w_1x_1 + w_2x_2$
- **threshold:** θ
- **activated: >** θ

Artificial Neuron – Computing Weights

Artificial Neuron – Computing Weights



Artificial Neuron

- **bias**

$$y = \begin{cases} 0 & \text{if } (w_1 x_1 + w_2 x_2 <= \theta) \\ 1 & \text{if } (w_1 x_1 + w_2 x_2 > \theta) \end{cases} \quad (1)$$

$$y = \begin{cases} 0 & \text{if } (b + w_1 x_1 + w_2 x_2 <= 0) \\ 1 & \text{if } (b + w_1 x_1 + w_2 x_2 > 0) \end{cases} \quad (2)$$

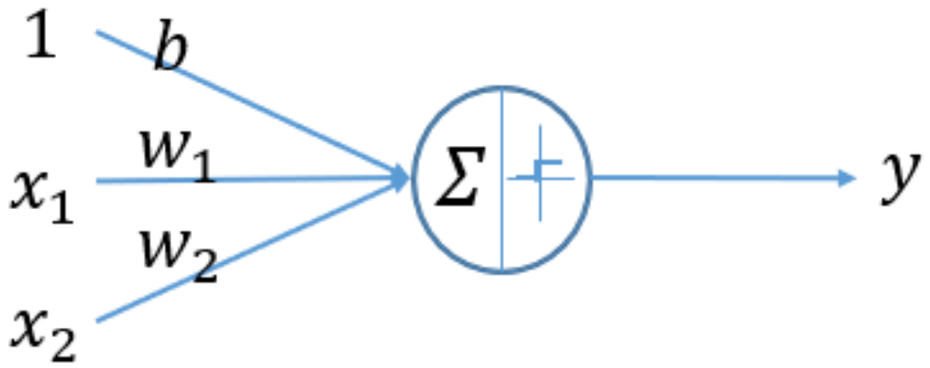
Artificial Neuron

- **bias**

$$y = \begin{cases} 0 & \text{if } (w_1 x_1 + w_2 x_2 <= \theta) \\ 1 & \text{if } (w_1 x_1 + w_2 x_2 > \theta) \end{cases} \quad (1)$$

$$y = \begin{cases} 0 & \text{if } (b + w_1 x_1 + w_2 x_2 <= 0) \\ 1 & \text{if } (b + w_1 x_1 + w_2 x_2 > 0) \end{cases} \quad (2)$$

bias

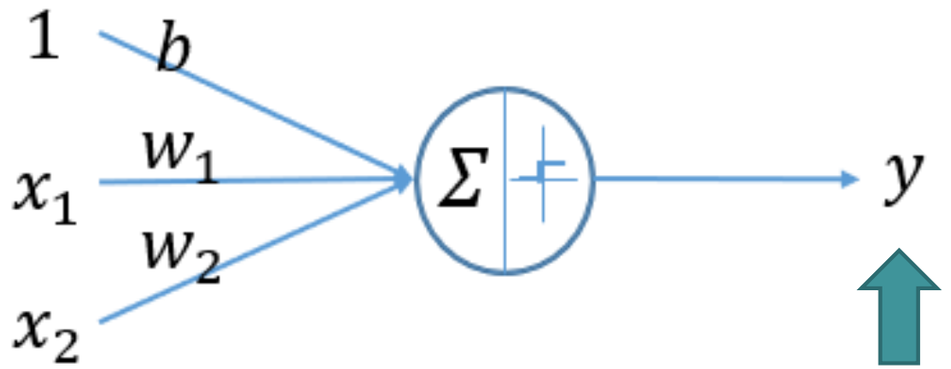


Artificial Neuron

- **bias**

$$y = \begin{cases} 0 & \text{if } (w_1 x_1 + w_2 x_2 <= \theta) \\ 1 & \text{if } (w_1 x_1 + w_2 x_2 > \theta) \end{cases} \quad (1)$$

$$y = \begin{cases} 0 & \text{if } (b + w_1 x_1 + w_2 x_2 <= 0) \\ 1 & \text{if } (b + w_1 x_1 + w_2 x_2 > 0) \end{cases} \quad (2)$$



Artificial Neuron

Artificial Neuron

- **Example 1:**
  - **Assume that weight w is set, threshold θ and input x_1, x_2 are given. Determine whether the neuron is activated or not.**

- $\mathbf{w} = (w_1, w_2) = (0.6, 0.3)$
- $\theta = 0.5$
- $(x_1, x_2) = (0, 1)$

Artificial Neuron

- **Example 1:**
  - Assume that weight w is set, threshold θ and input x_1, x_2 are given. Determine whether the neuron is activated or not.

- $\mathbf{w} = (w_1, w_2) = (0.6, 0.3)$
- $\theta = 0.5$
- $(x_1, x_2) = (0, 1)$

(1) activated
(2) not activated

Artificial Neuron

- **Example 1:**
  - **Assume that weight w is set, threshold θ and input x_1, x_2 are given. Determine whether the neuron is activated or not.**

$$net\ input = w_1 x_1 + w_2 x_2$$

- $\mathbf{w} = (w_1, w_2) = (0.6, 0.3)$
- $\theta = 0.5$
- $(x_1, x_2) = (0, 1)$

(1) activated
(2) not activated

Artificial Neuron

- **Example 1:**
  - **Assume that weight w is set, threshold θ and input x_1, x_2 are given. Determine whether the neuron is activated or not.**

$$
\begin{aligned}
&net\ input \\
&= w_1 x_1 + w_2 x_2 \\
&= 0.6 \times 0 + 0.3 \times 1 \\
&= 0.3 < \theta
\end{aligned}
$$

- $\mathbf{w} = (w_1, w_2) = (0.6, 0.3)$
- $\theta = 0.5$
- $(x_1, x_2) = (0, 1)$

(1) activated
(2) not activated

Artificial Neuron

- **Example 1:**
  - **Assume that weight w is set, threshold θ and input x_1, x_2 are given. Determine whether the neuron is activated or not.**

$$
\begin{aligned}
&net\ input \\
&= w_1 x_1 + w_2 x_2 \\
&= 0.6 \times 0 + 0.3 \times 1 \\
&= 0.3 < \theta
\end{aligned}
$$

- $\mathbf{w} = (w_1, w_2) = (0.6, 0.3)$
- $\theta = 0.5$
- $(x_1, x_2) = (0, 1)$

(1) activated

✓ (2) not activated

Implementing AND Neuron

Implementing AND Neuron

Truth Table

AND 진리표

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

Implementing AND Neuron

$$y = \begin{cases} 0 & \text{if } (b + w_1 x_1 + w_2 x_2 <= 0) \\ 1 & \text{if } (b + w_1 x_1 + w_2 x_2 > 0) \end{cases} \quad (2)$$

Truth Table

AND 진리표

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

Implementing AND Neuron

Truth Table

AND 진리표

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

$$y = \begin{cases} 0 & \text{if } (b + w_1 x_1 + w_2 x_2 <= 0) \\ 1 & \text{if } (b + w_1 x_1 + w_2 x_2 > 0) \end{cases} \quad (2)$$

- **Example:**

Implementing AND Neuron

Truth Table

AND 진리표

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

$$y = \begin{cases} 0 & \text{if } (b + w_1 x_1 + w_2 x_2 <= 0) \\ 1 & \text{if } (b + w_1 x_1 + w_2 x_2 > 0) \end{cases} \quad (2)$$

- **Example:**

  Among the following combinations of weight and bias, find one that satisfies the equations (2) above and makes AND Neuron.

$$(1)\ (w_1, w_2) = (0.5, 0.5),\ b = -0.7$$
$$(2)\ (w_1, w_2) = (0.5, 0.5),\ b = -0.3$$
$$(3)\ (w_1, w_2) = (0.5, 0.5),\ b = 0.2$$

Implementing AND Neuron

$$y = \begin{cases} 0 & \text{if } (b + w_1 x_1 + w_2 x_2 <= 0) \\ 1 & \text{if } (b + w_1 x_1 + w_2 x_2 > 0) \end{cases} \quad (2)$$

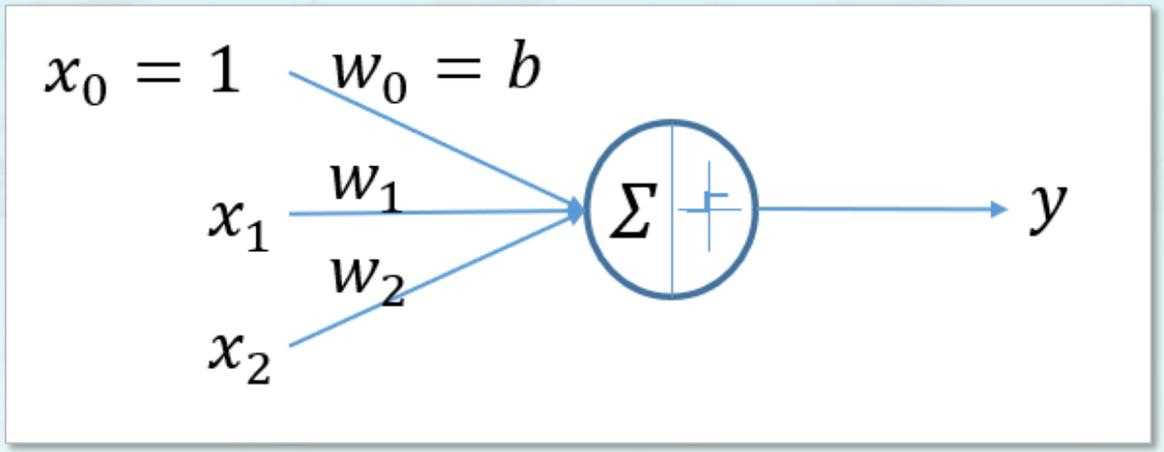
- **Example:**

  Among the following combinations of weight and bias, find one that satisfies the equations (2) above and makes AND Neuron.

(1) $(w_1, w_2) = (0.5, 0.5), b = -0.7$
(2) $(w_1, w_2) = (0.5, 0.5), b = -0.3$
(3) $(w_1, w_2) = (0.5, 0.5), b = 0.2$

Implementing AND Neuron

$$(1)\ (w_1, w_2) = (0.5, 0.5),\ b = -0.7$$

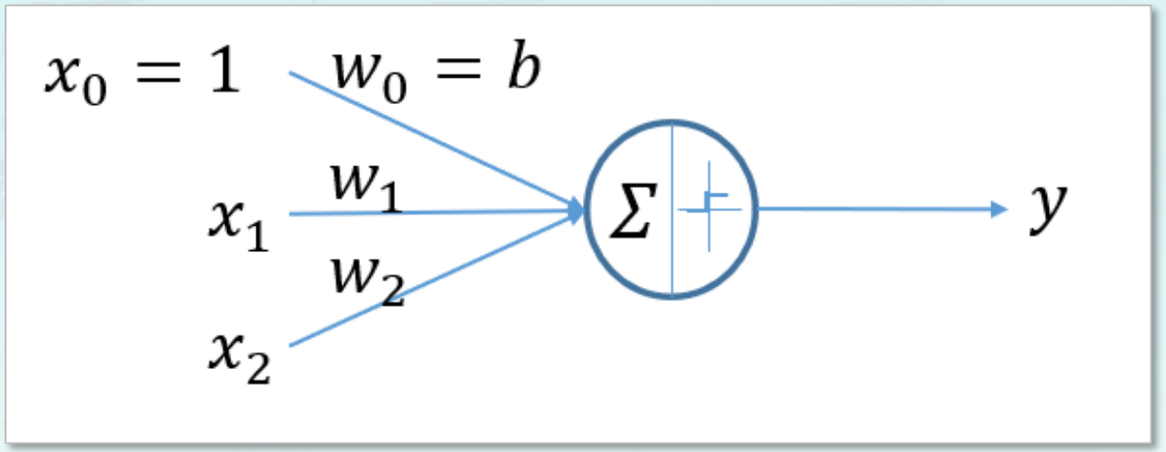



```
def AND
```

Implementing AND Neuron

1. AND(x1, x2)
2. AND(x0, x1, x2)
3. AND(1, x1, x2)

$$(1)\ (w_1, w_2) = (0.5, 0.5),\ b = -0.7$$

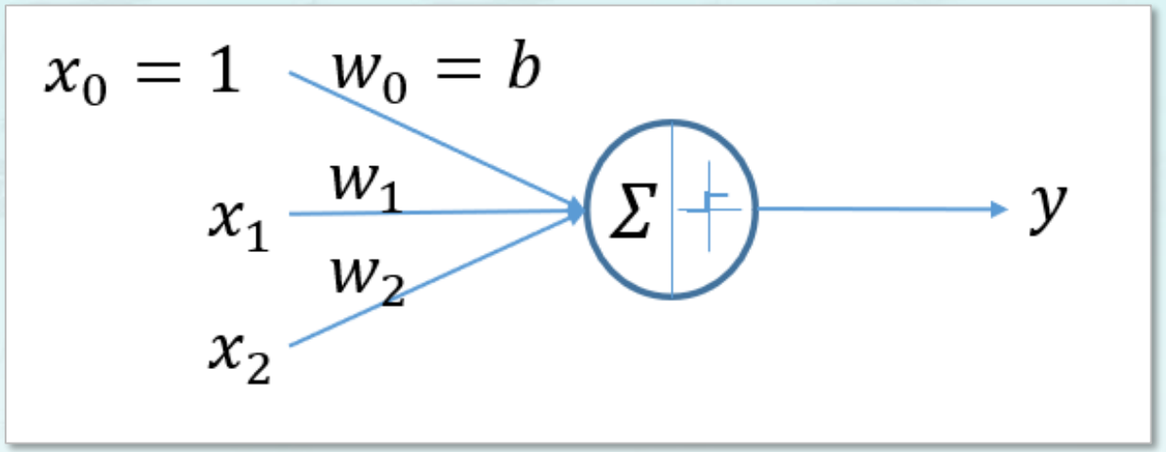



```
def AND
```

Implementing AND Neuron

1. AND(x1, x2)
2. AND(x0, x1, x2)
3. AND(1, x1, x2)

$$(1)\ (w_1, w_2) = (0.5, 0.5),\ b = -0.7$$

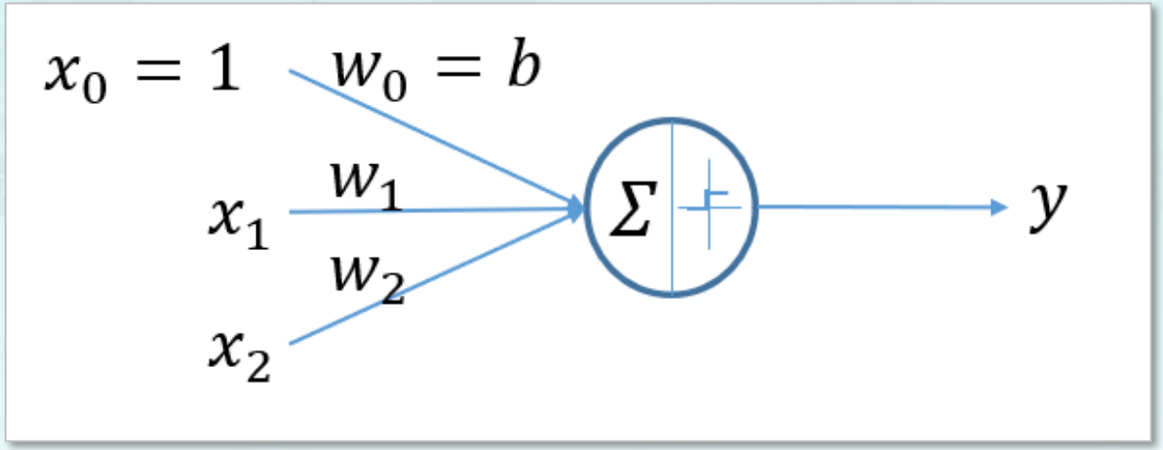



```
def AND(x1, x2):
```

Implementing AND Neuron

1. AND(x1, x2)
2. AND(x0, x1, x2)
3. AND(1, x1, x2)

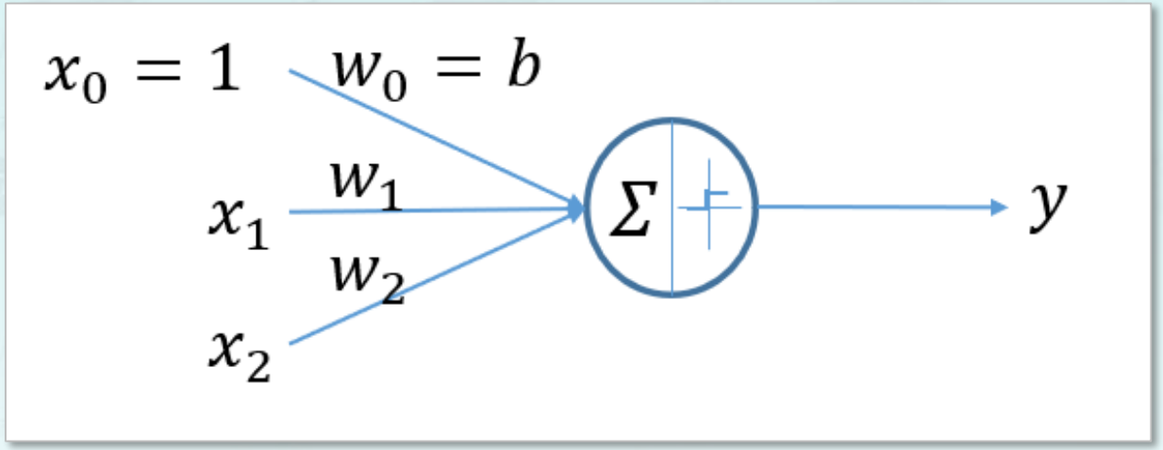
$$(1)\ (w_1, w_2) = (0.5, 0.5),\ b = -0.7$$




```
def AND(x1, x2):
```

Implementing AND Neuron

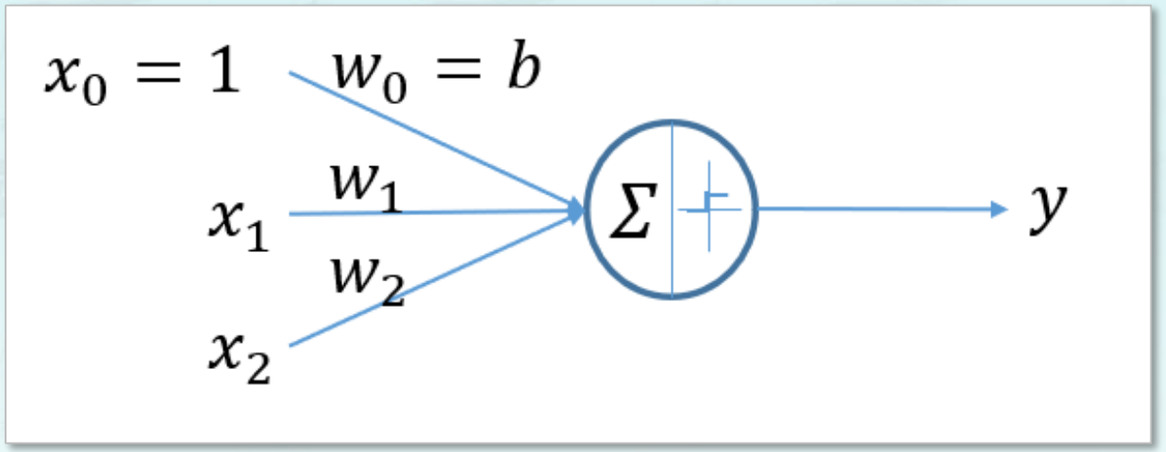
$$(1)\ (w_1, w_2) = (0.5, 0.5),\ b = -0.7$$




```
def AND(x1, x2):
    x = np.array([1, x1, x2])          # input
```

Implementing AND Neuron

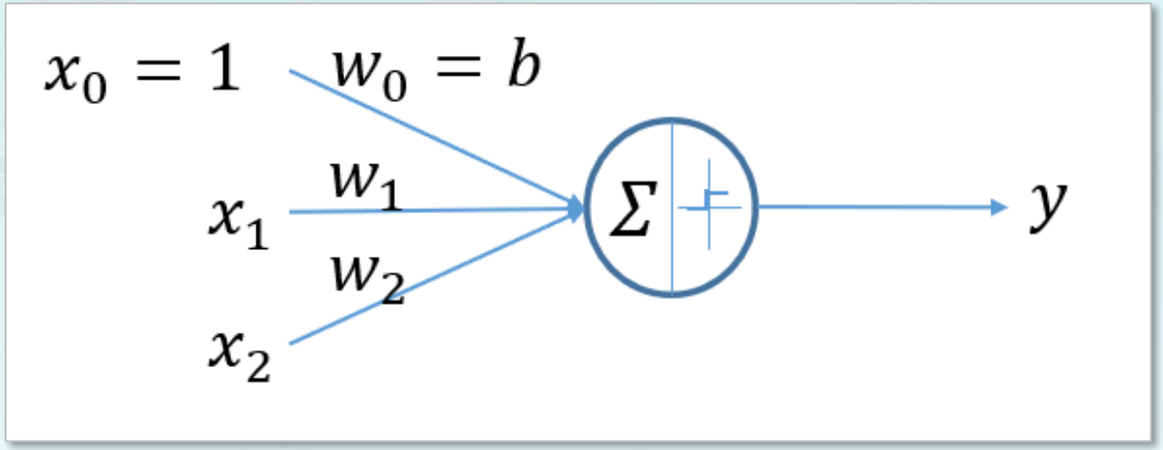
$$(1)\ (w_1, w_2) = (0.5, 0.5),\ b = -0.7$$




```
def AND(x1, x2):
    x = np.array([1, x1, x2])              # input
    w = np.array([-0.7, 0.5, 0.5])         # bias + weight
```

Implementing AND Neuron

$$(1)\ (w_1, w_2) = (0.5, 0.5),\ b = -0.7$$

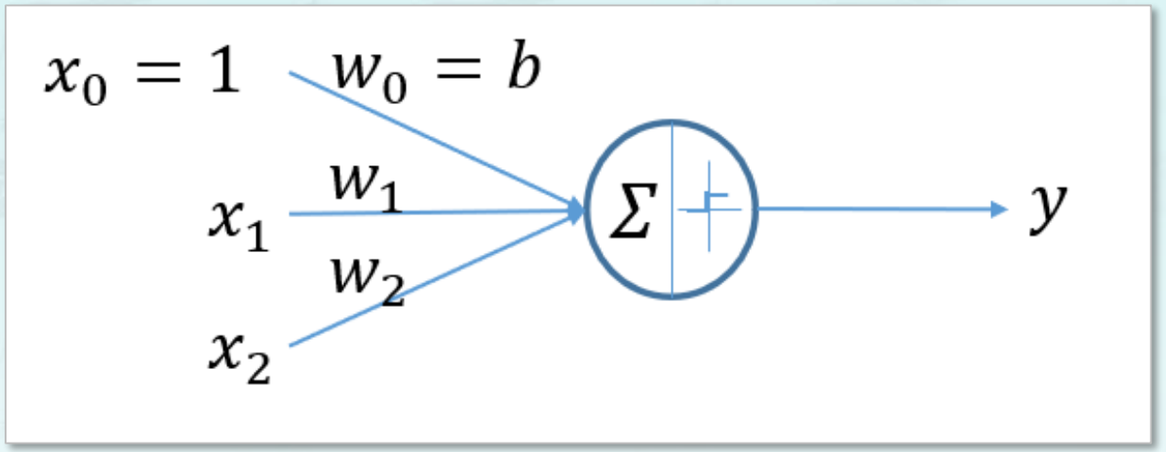



```
def AND(x1, x2):
    x = np.array([1, x1, x2])              # input
    w = np.array([-0.7, 0.5, 0.5])         # bias + weight
```

return

Implementing AND Neuron

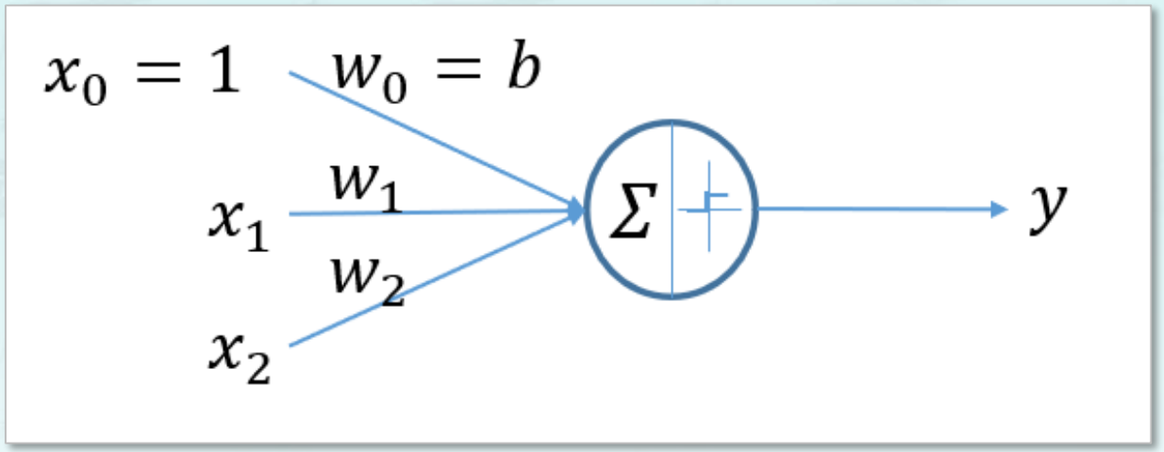
$$(1)\ (w_1, w_2) = (0.5, 0.5),\ b = -0.7$$




```
def AND(x1, x2):
    x = np.array([1, x1, x2])              # input
    w = np.array([-0.7, 0.5, 0.5])         # bias + weight
          return  np.dot(w, x)
```

Implementing AND Neuron

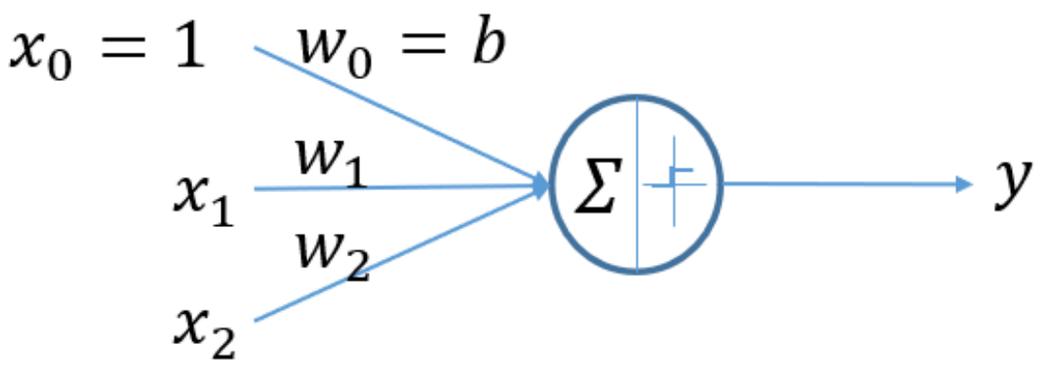
$$(1)\ (w_1, w_2) = (0.5, 0.5),\ b = -0.7$$




```
def AND(x1, x2):
    x = np.array([1, x1, x2])          # input
    w = np.array([-0.7, 0.5, 0.5])     # bias + weight
    return  np.dot(w, x) > 0
```

Implementing AND Neuron

$$(1)\ (w_1, w_2) = (0.5, 0.5),\ b = -0.7$$




```
def AND(x1, x2):
    x = np.array([1, x1, x2])              # input
    w = np.array([-0.7, 0.5, 0.5])         # bias + weight
    return int(np.dot(w, x) > 0)
```

Implementing AND Neuron

```
def AND(x1, x2):
    x = np.array([1, x1, x2])          # input
    w = np.array([-0.7, 0.5, 0.5])     # bias + weight
    return int(np.dot(w, x) > 0)
```

Implementing AND Neuron

```
print("AND(0, 0) = ", AND(0, 0))
print("AND(0, 1) = ", AND(0, 1))
print("AND(1, 0) = ", AND(1, 0))
print("AND(1, 1) = ", AND(1, 1))
```

```
AND(0, 0) =  0
AND(0, 1) =  0
AND(1, 0) =  0
AND(1, 1) =  1
```

```
def AND(x1, x2):
    x = np.array([1, x1, x2])              # input
    w = np.array([-0.7, 0.5, 0.5])         # bias + weight
    return int(np.dot(w, x) > 0)
```

Visualizing AND Neuron

$$AND(x_1, x_2) \begin{cases} -0.7 + 0.5x_1 + 0.5x_2 <= 0 \\ -0.7 + 0.5x_1 + 0.5x_2 > 0 \end{cases} \quad (3)$$

$$y = ax + b$$

Visualizing AND Neuron

$$AND(x_1, x_2) \begin{cases} -0.7 + 0.5x_1 + 0.5x_2 <= 0 \\ -0.7 + 0.5x_1 + 0.5x_2 > 0 \end{cases} \quad (3)$$

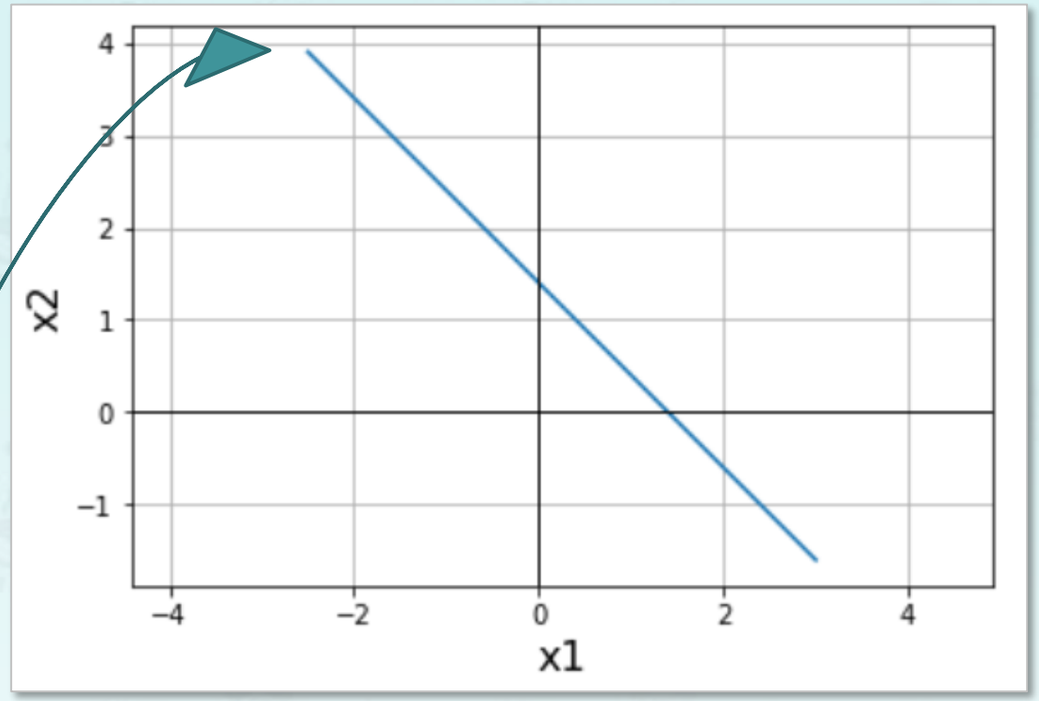
$$y = ax + b$$

$$-0.7 + 0.5x_1 + 0.5x_2 = 0$$

$$x_2 = -\frac{0.5}{0.5}x_1 + \frac{0.7}{0.5}$$

$$x_2 = -x_1 + 1.4 \quad (4)$$

Visualizing AND Neuron

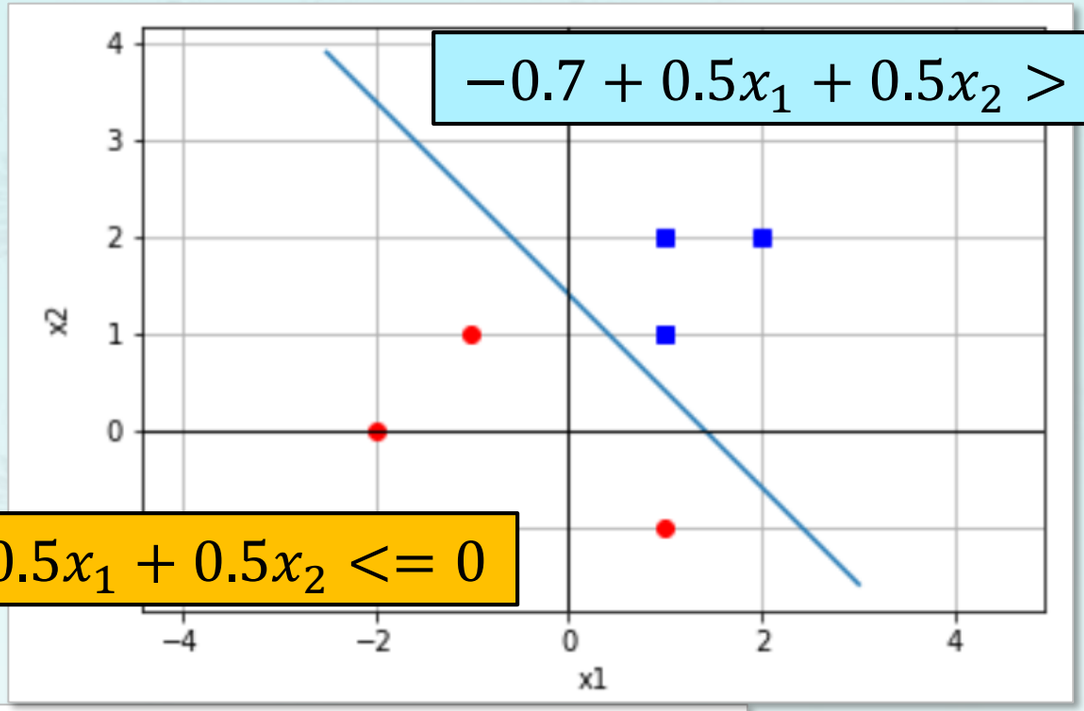


$$-0.7 + 0.5x_1 + 0.5x_2 = 0$$

$$x_2 = -\frac{0.5}{0.5}x_1 + \frac{0.7}{0.5}$$

$$x_2 = -x_1 + 1.4 \qquad (4)$$

Visualizing AND Neuron



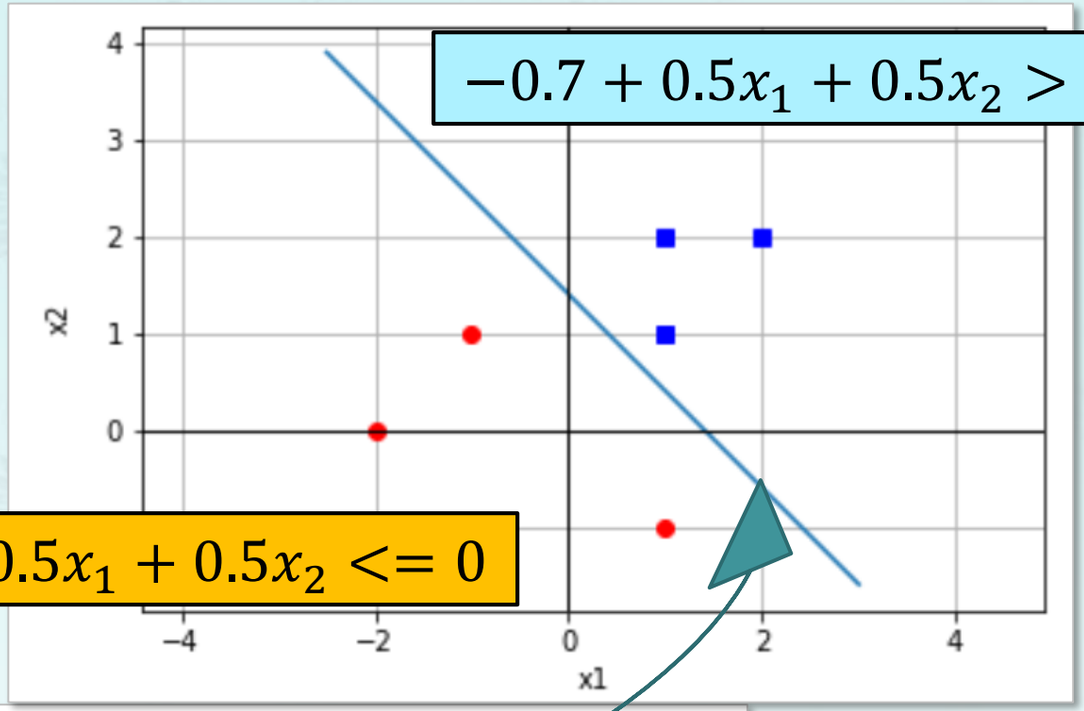
$$-0.7 + 0.5x_1 + 0.5x_2 = 0$$

$$x_2 = -\frac{0.5}{0.5}x_1 + \frac{0.7}{0.5}$$

$$x_2 = -x_1 + 1.4 \qquad (4)$$

Visualizing AND Neuron

- Classification
- Classifier



$$-0.7 + 0.5x_1 + 0.5x_2 = 0$$

$$x_2 = -\frac{0.5}{0.5}x_1 + \frac{0.7}{0.5}$$

$$x_2 = -x_1 + 1.4 \qquad (4)$$

Artificial Neuron

- **Summary:**
  - Understanding Artificial Neuron
  - Implementing AND Neuron
  - Visualizing AND Neuron

- **Next**
  - 3-2 Derivatives

Week3(1/3)

Artificial Neuron

Machine Learning with Python

Handong Global University
Prof. Youngsup Kim
idebtor@gmail.com