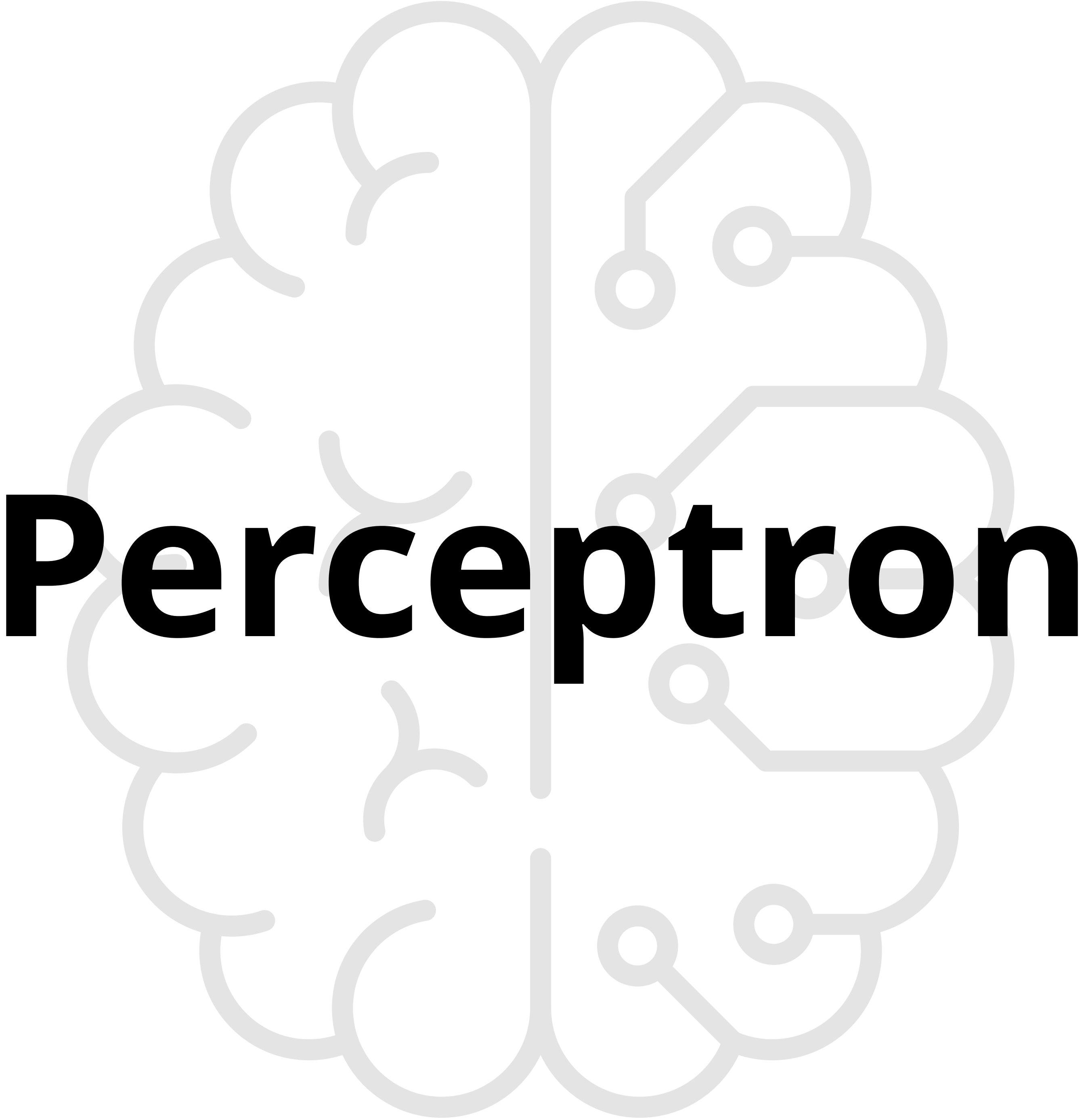
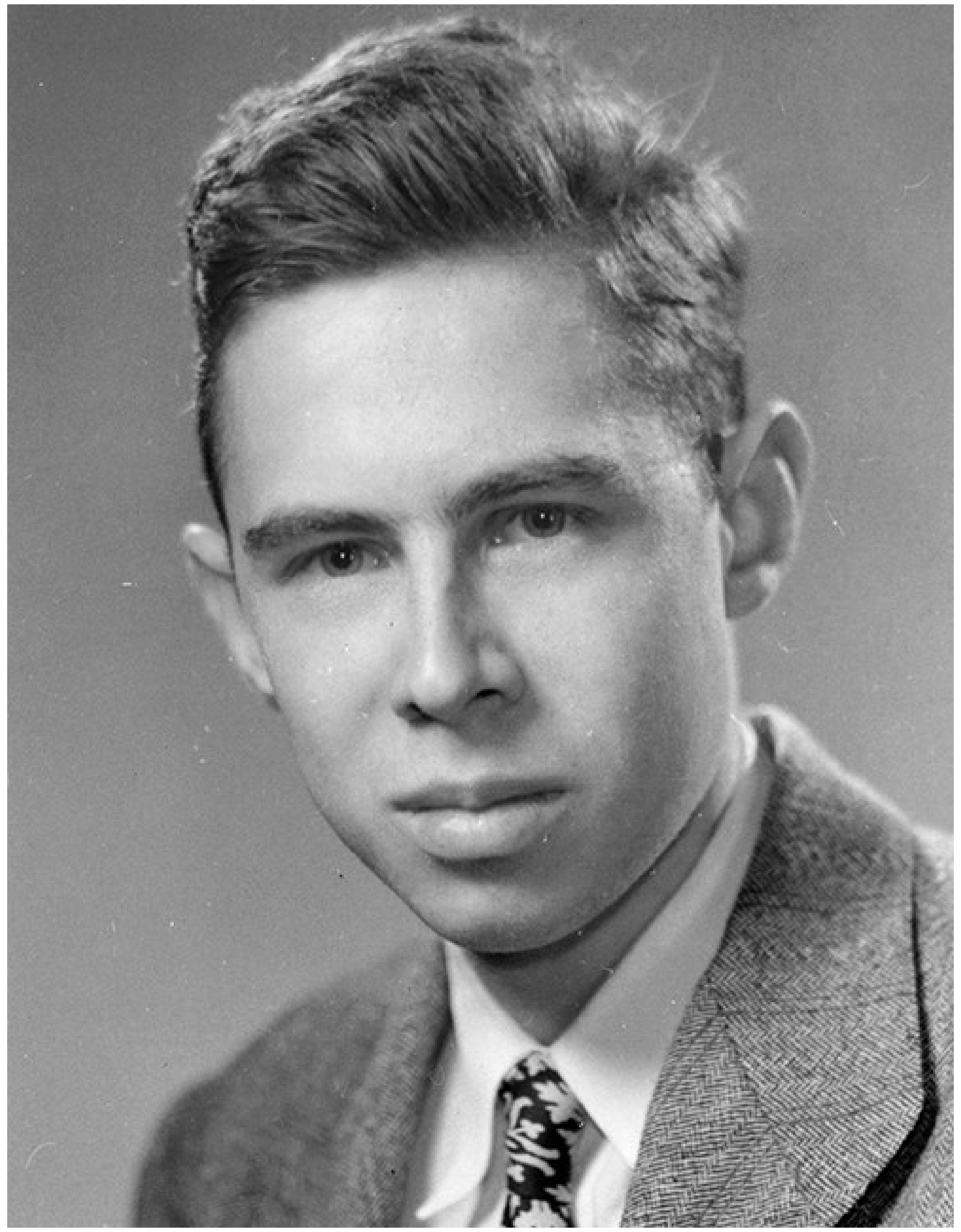
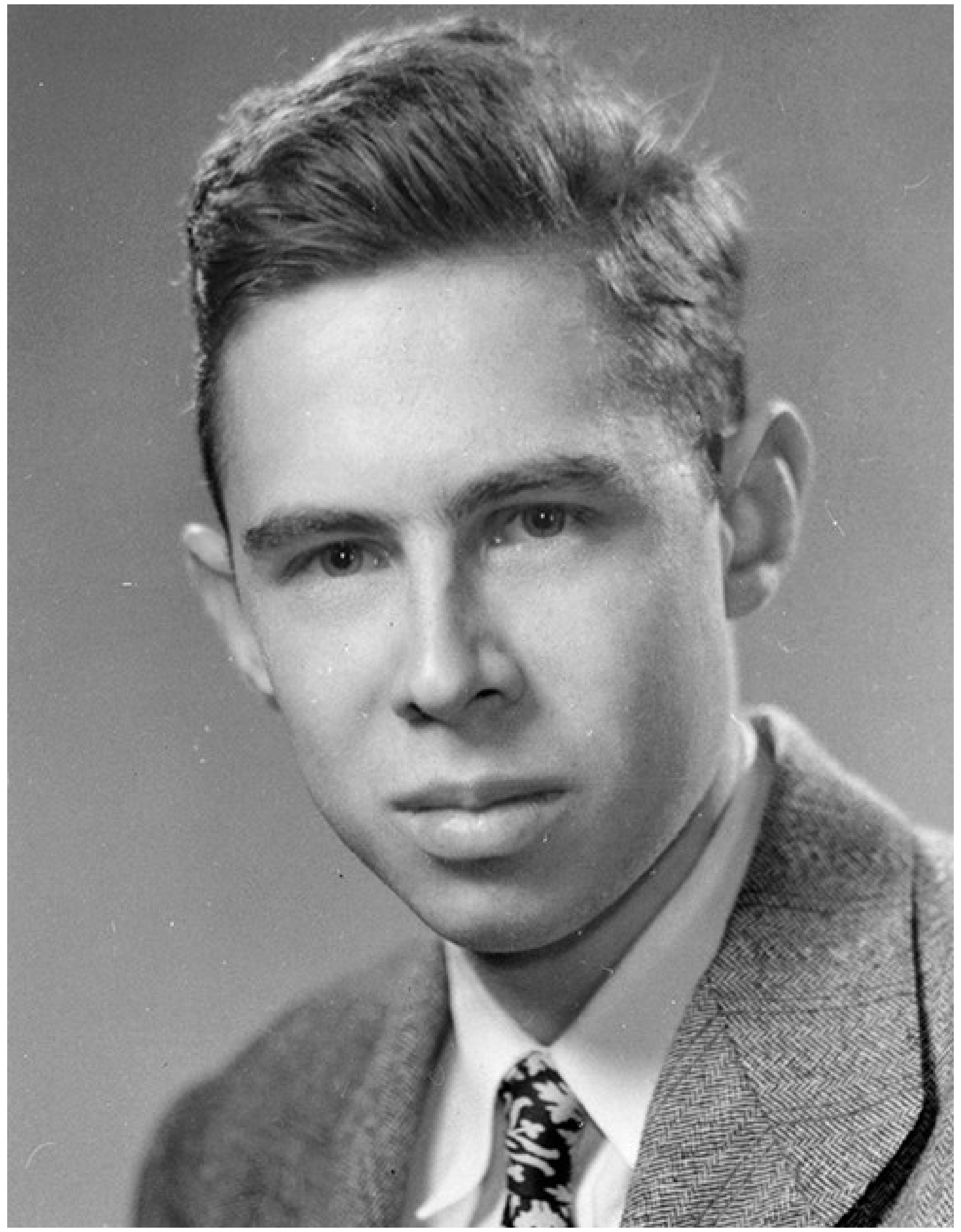


Perceptron





Frank Rosenblatt

What is a Perceptron?

Mathematical Model of a Biological Neuron

Building Block of an Artificial Neural Network.

Supervised Learning Algorithm for Binary Classifier.

The Perceptron

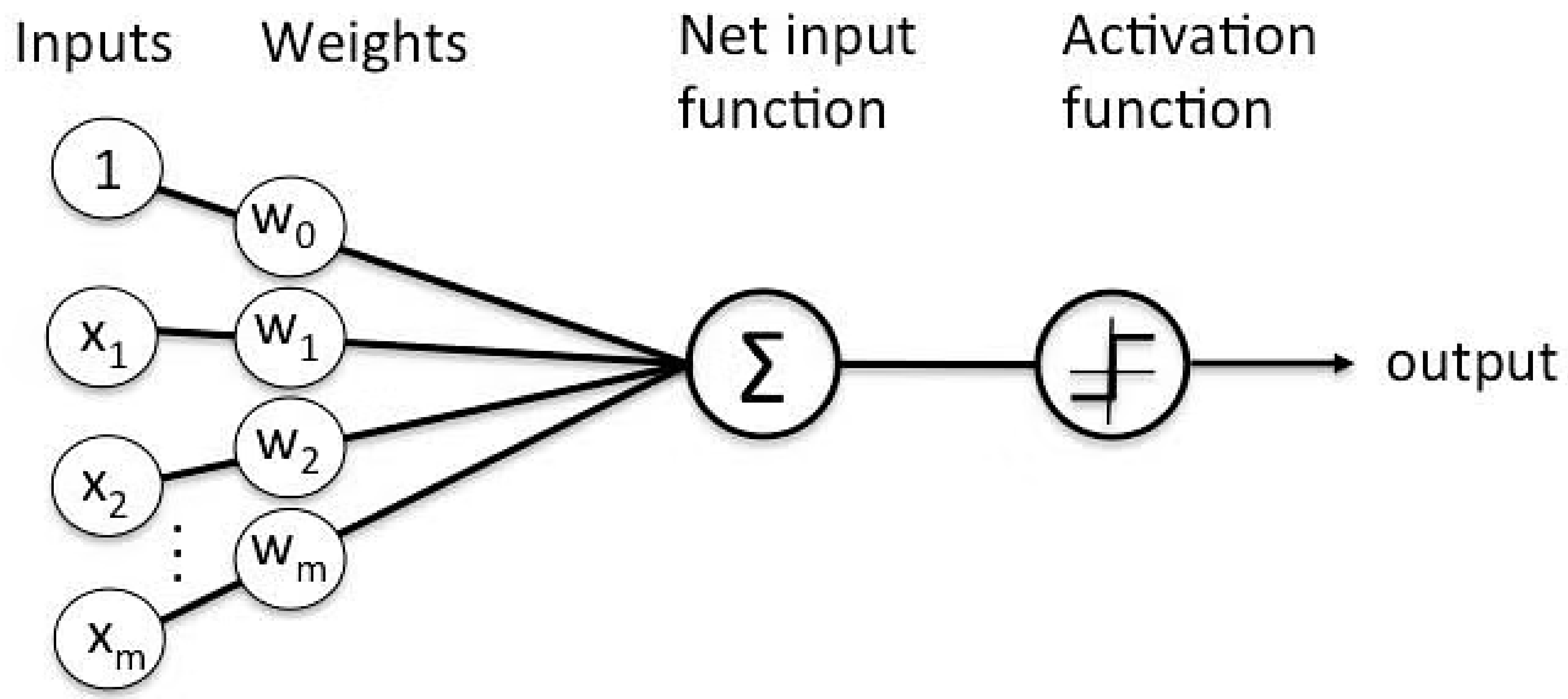
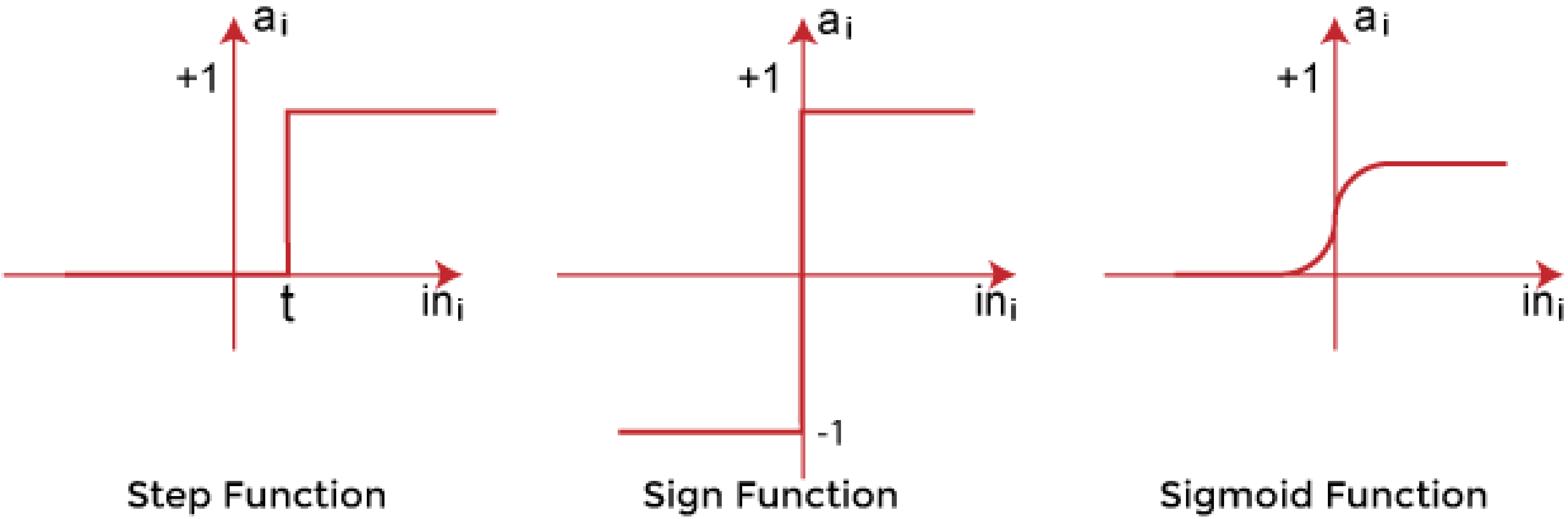


Image: https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron

Activation Functions



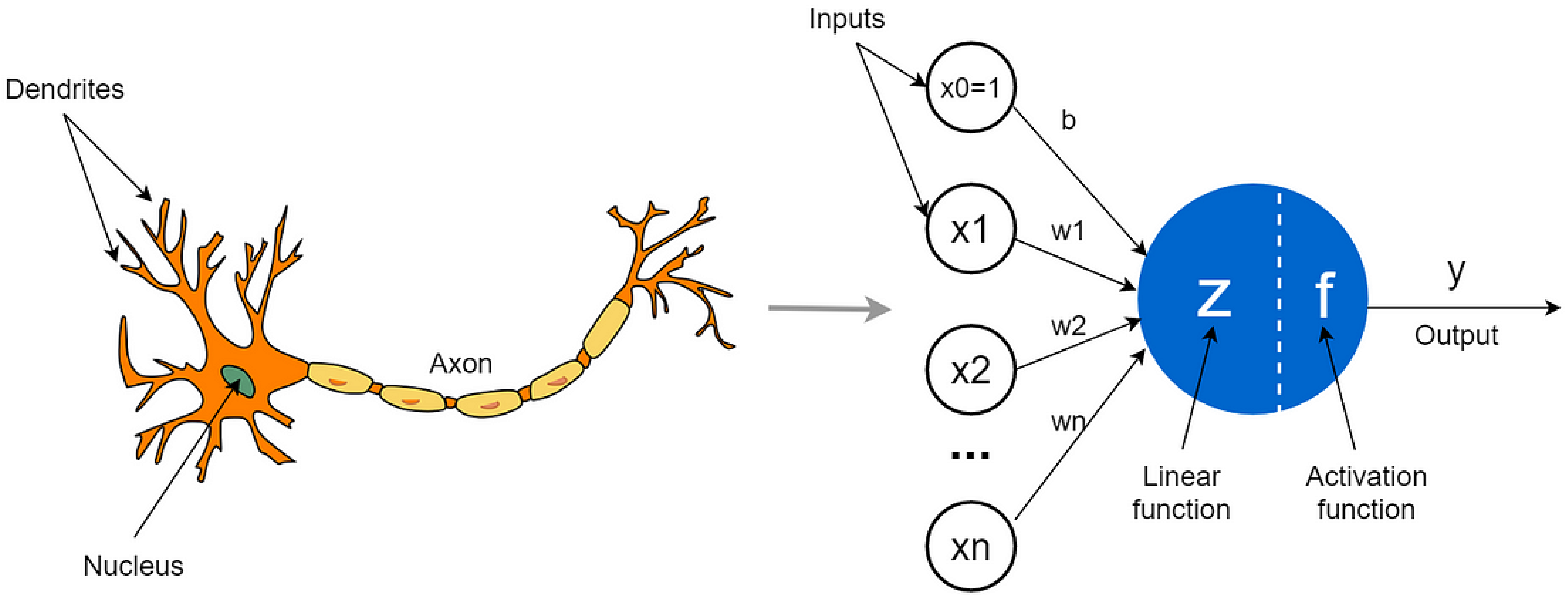
Step Function

Sign Function

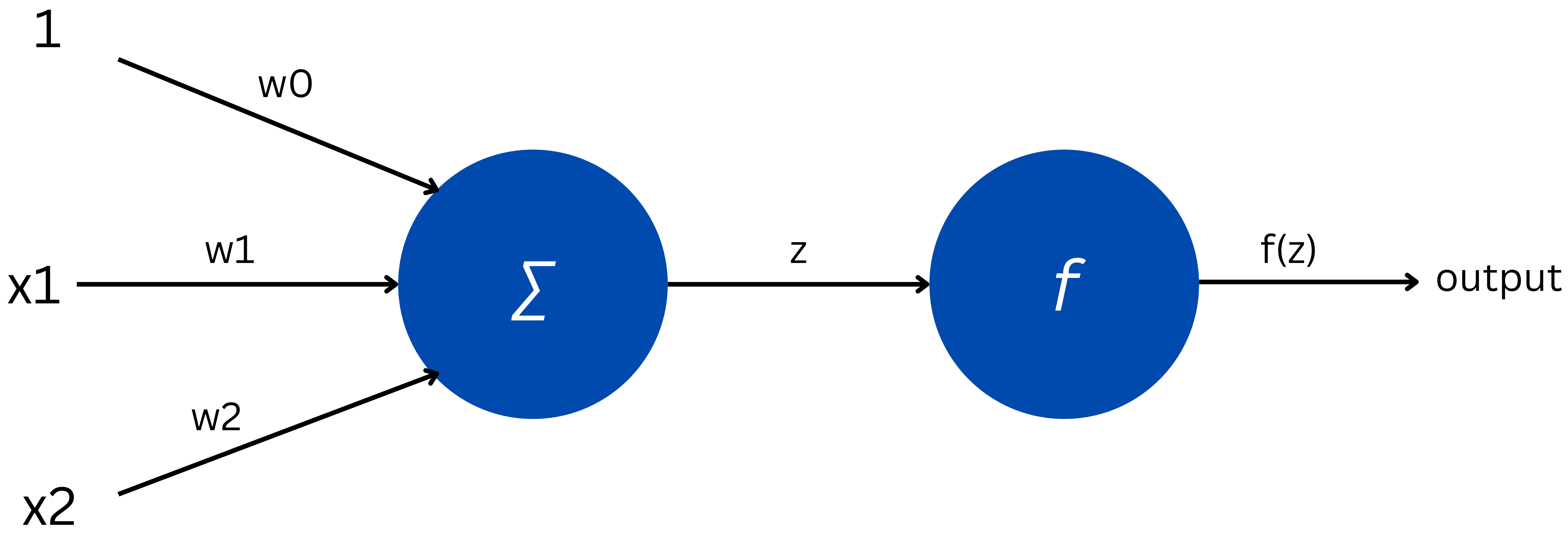
Sigmoid Function

Image: https://www.javatpoint.com/perceptron-in-machine-learning

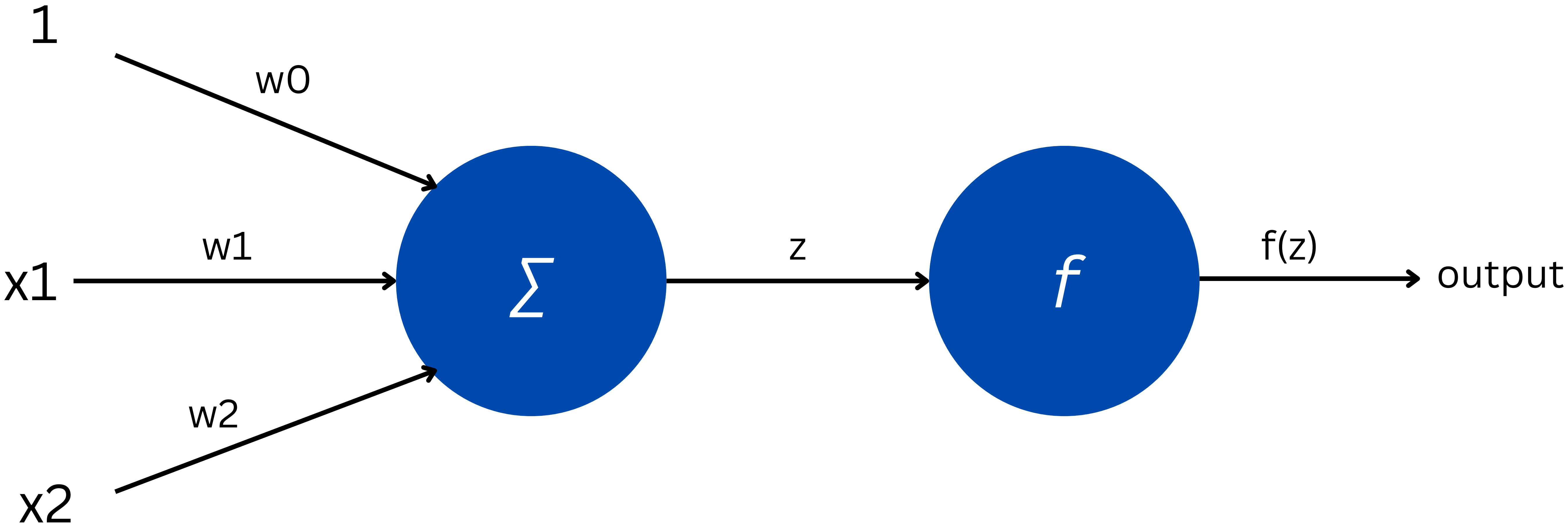
Biological Neuron vs Perceptron



2 Input Perceptron

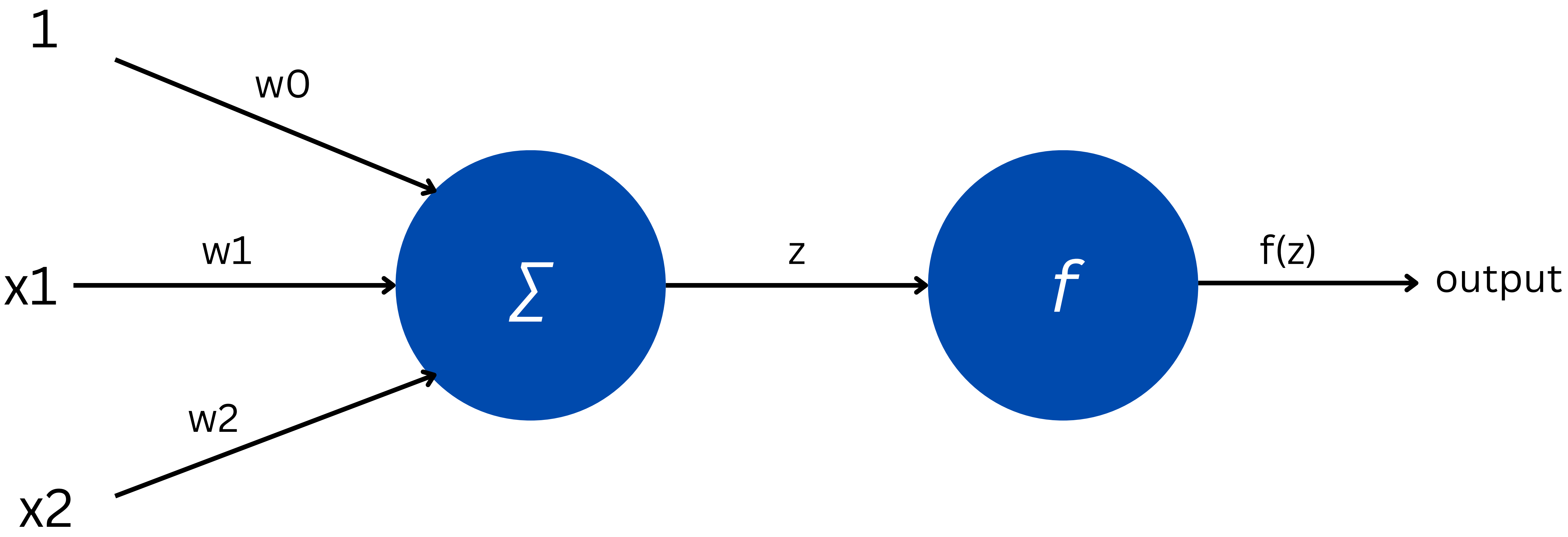


2 Input Perceptron



$$z = w0 + w1 \cdot x1 + w2 \cdot x2$$

2 Input Perceptron

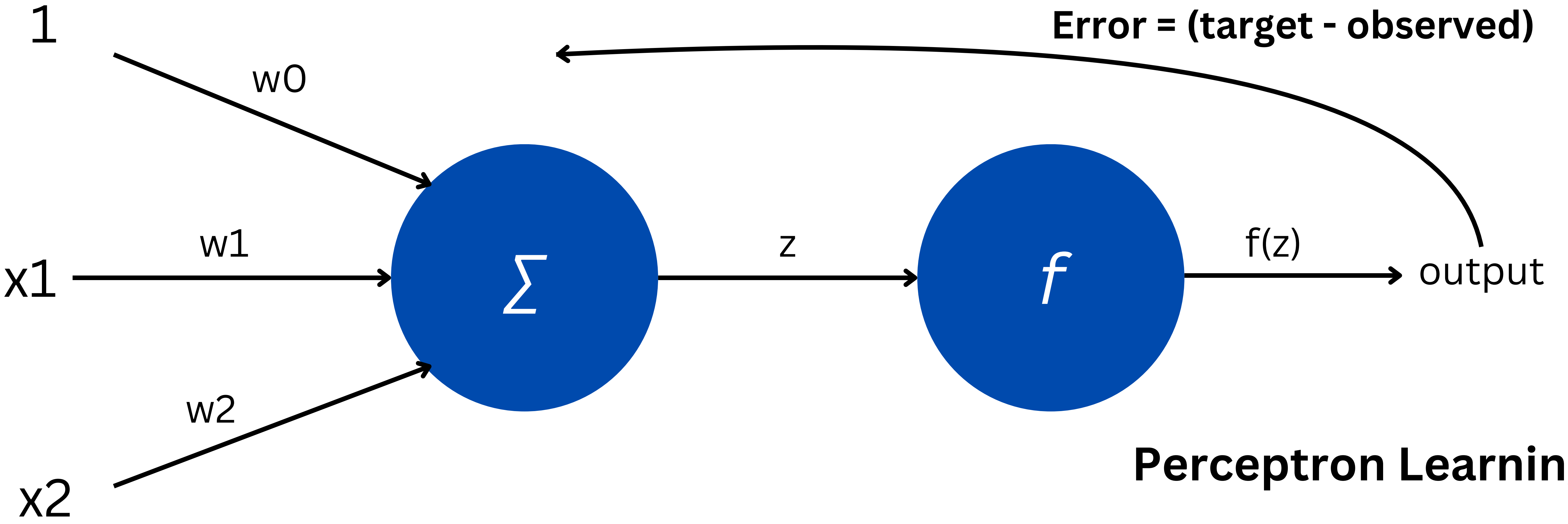


$$z = w0 + w1 \cdot x1 + w2 \cdot x2$$

$$f(z) = 1 \quad \text{if } z >= 0$$
$$= 0 \quad \text{if } z < 0$$

Error

Perceptron Training



Error = (target - observed)

1

w0

x1

w1

∑

z

f

f(z)

output

w2

x2

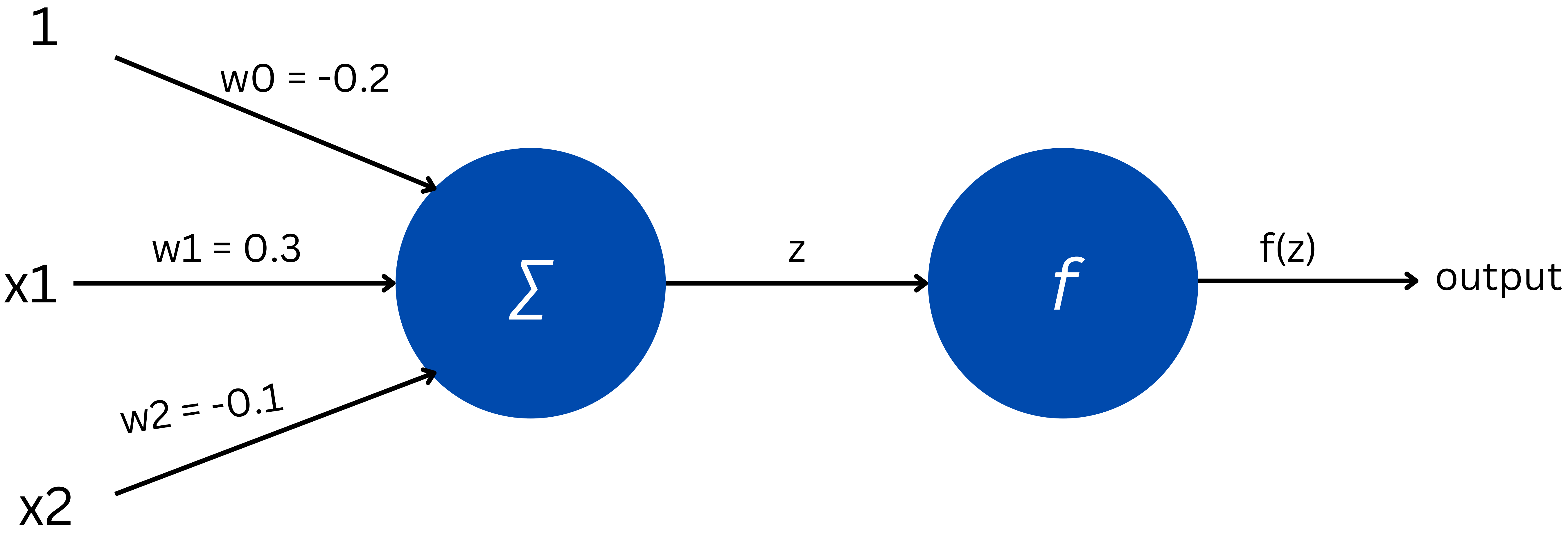
Perceptron Learning Rule

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i = \eta(t - o)x_i$$

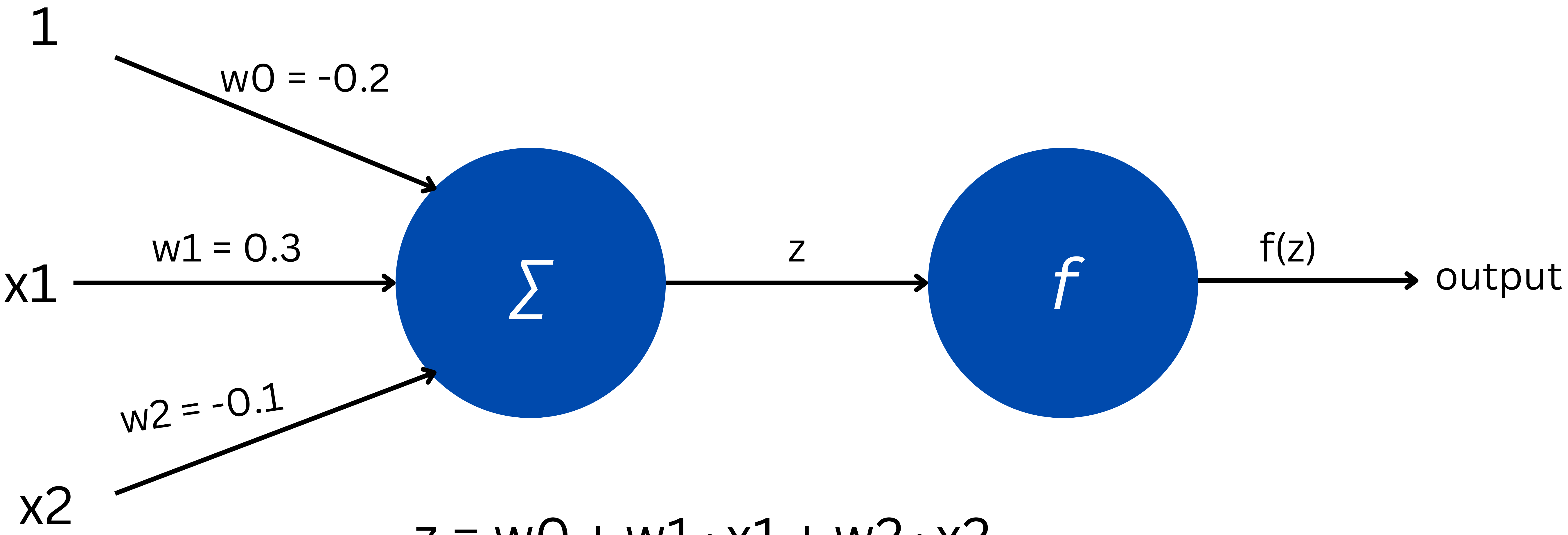
Training Algorithm

Step 1: Initialization



Training Algorithm

Step 2: Activation



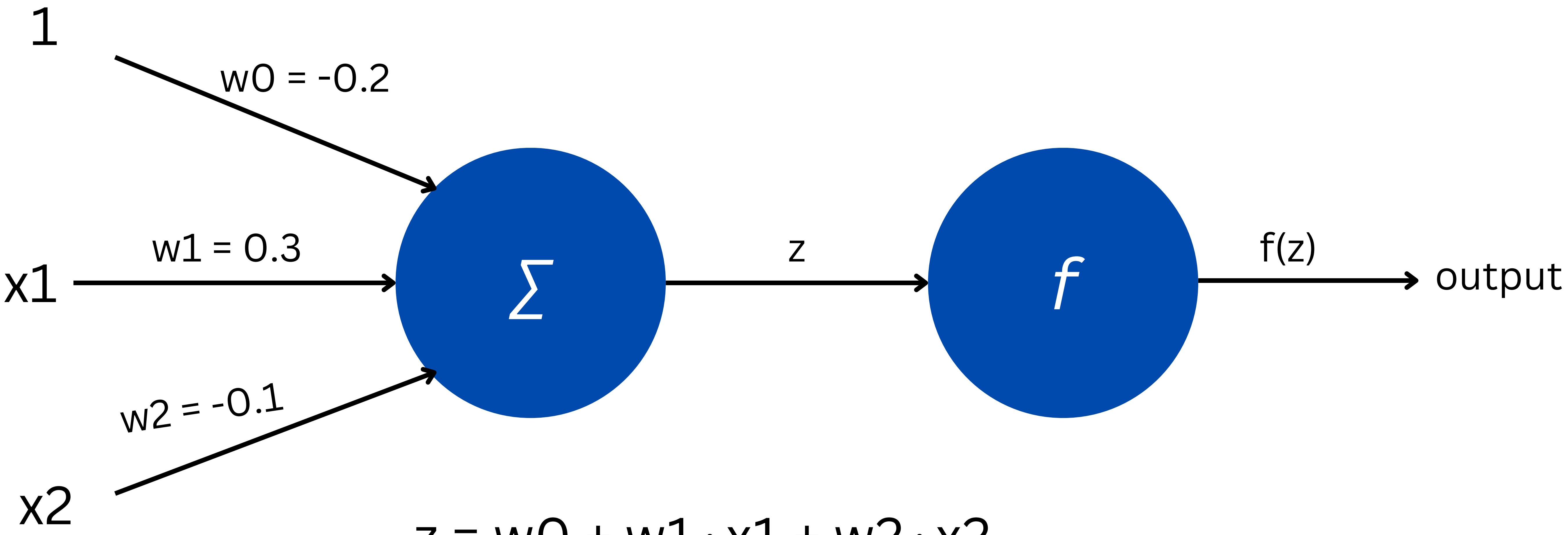
$$z = w0 + w1 \cdot x1 + w2 \cdot x2$$

Error = target - observed

$$f(z) = \begin{cases} 1 & \text{if } z >= 0 \\ 0 & \text{if } z < 0 \end{cases}$$

Training Algorithm

Step 3: Weight Update



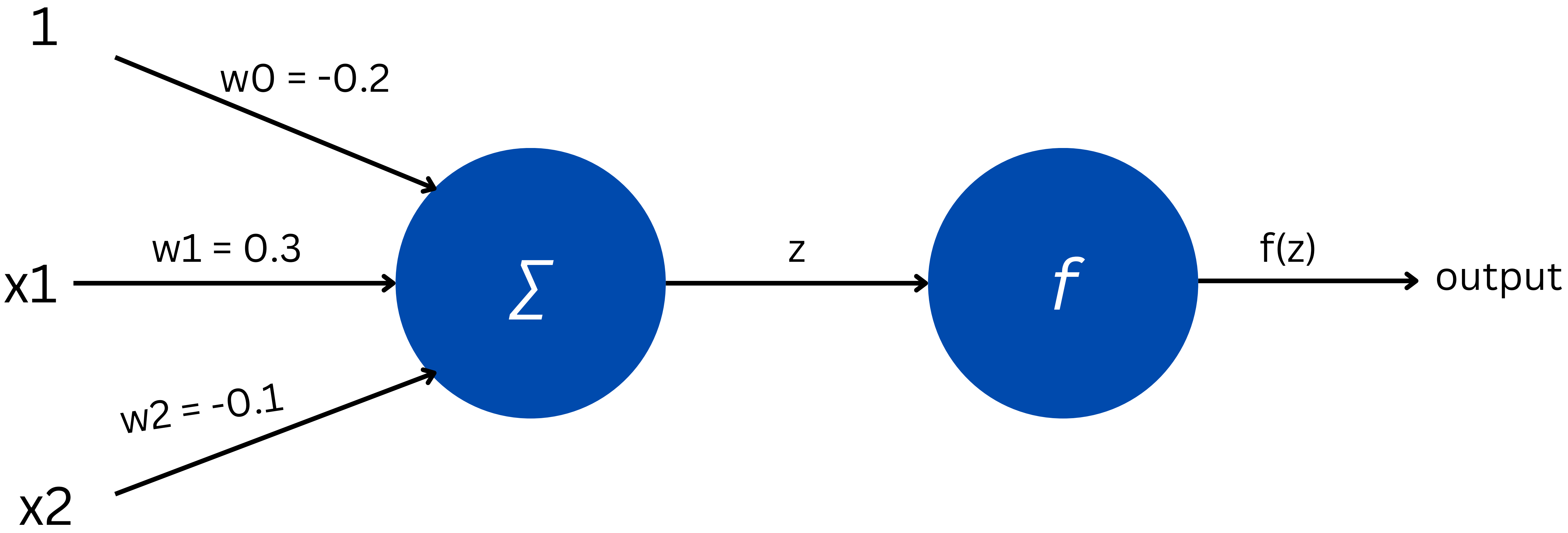
$$z = w0 + w1 \cdot x1 + w2 \cdot x2$$

Error = target - observed

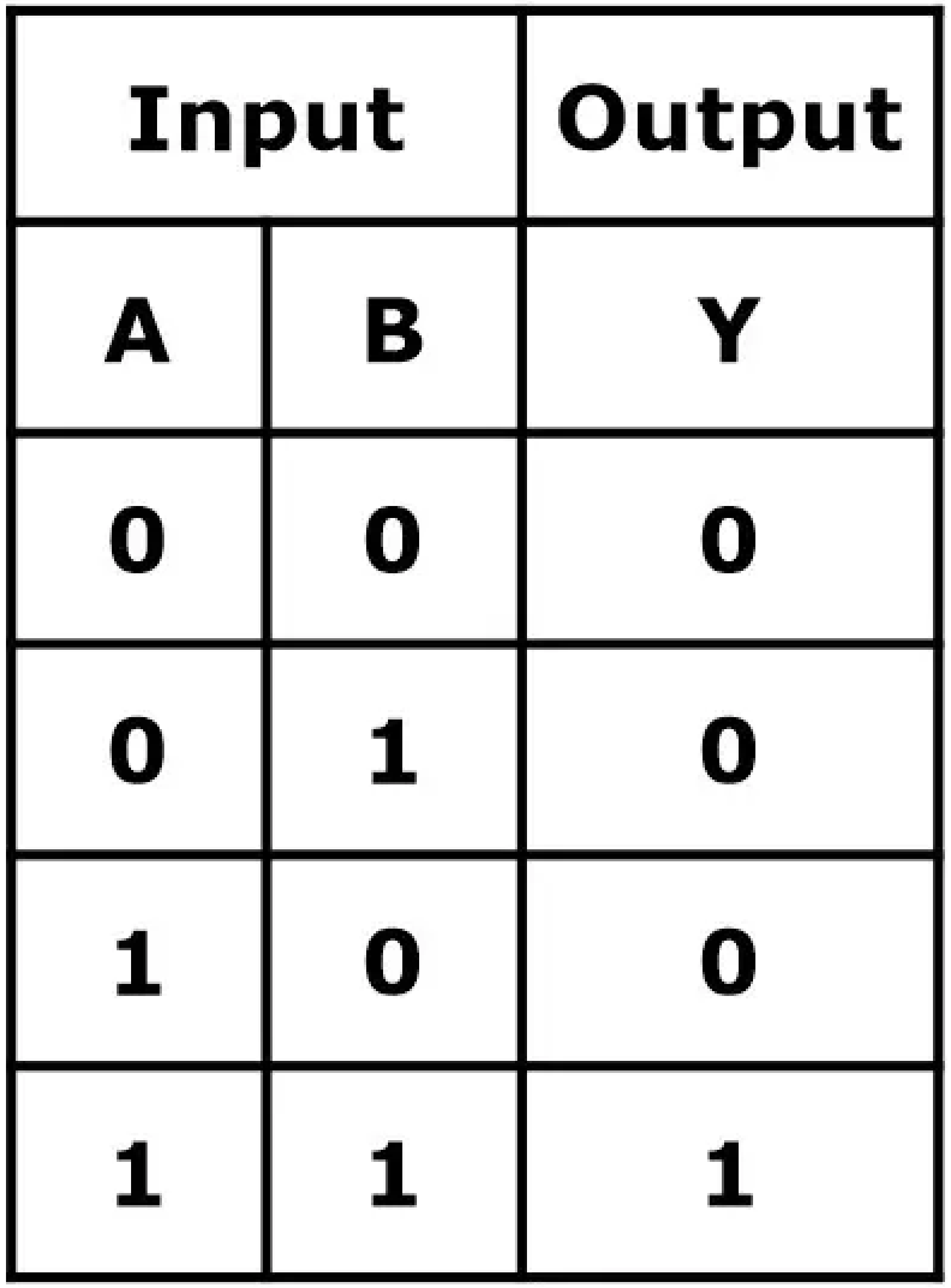
$$\begin{aligned} f(z) &= 1 \quad \text{if } z >= 0 \\ &= 0 \quad \text{if } z < 0 \end{aligned}$$

Training Algorithm

Step 4: Iteration

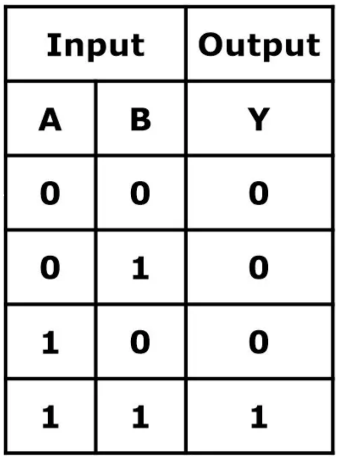


Building a Logical AND



Input		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Building a Logical AND



Input		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

We have:

w0 = -0.2, w1 = 0.3, w2 = -0.1, lr = 0.1

1st Epoch:

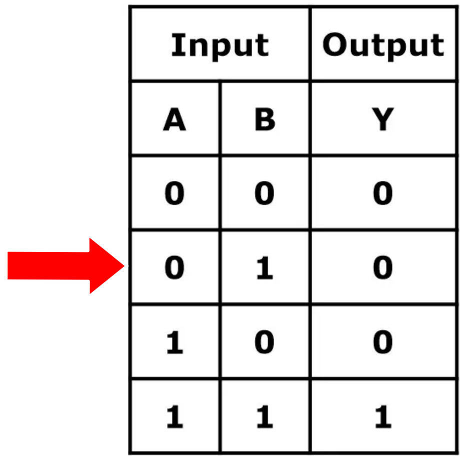
For x1 = 0, x2 = 0

z = w0 + w1*x1 + w2*x2 = -0.2 + 0.3 * 0 + (-0.1) * 0 = - 0.2

output = f(z) = f(-0.2) = 0

target = 0 (No need for weight update)

Building a Logical AND



Input		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

We have:

w0 = -0.2, w1 = 0.3, w2 = -0.1, lr = 0.1

1st Epoch:

For x1 = 0, x2 = 1

z = w0 + w1*x1 + w2*x2 = -0.2 + 0.3 * 0 + (-0.1) * 1 = - 0.3

output = f(z) = f(-0.3) = 0

target = 0 (No need for weight update)

Building a Logical AND

Input		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

We have:

w0 = -0.2, w1 = 0.3, w2 = -0.1, lr = 0.1

1st Epoch:

For x1 = 1, x2 = 0

z = w0 + w1*x1 + w2*x2 = -0.2 + 0.3 * 1 + (-0.1) * 0 = 0.1

output = f(z) = f(0.1) = 1

target = 0 (Weight Update Required)

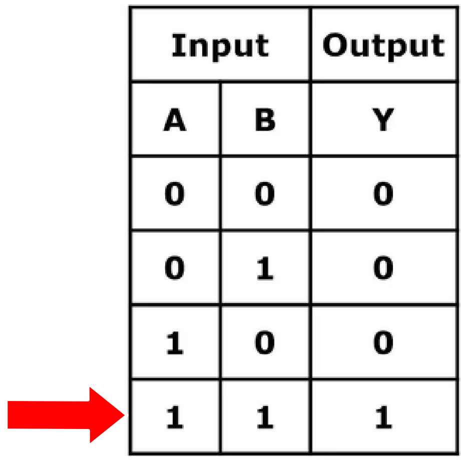
Weight Update:

w0 = -0.2 + 0.1*(0-1)*1= -0.3

w1 = 0.3 + 0.1*(0-1)*1 = 0.2

w2 = -0.1 + 0.1*(0-1)*0 = -0.1

Building a Logical AND



Input		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i = \eta(t - o)x_i$$

We have:

w0 = -0.3, w1 = 0.2, w2 = -0.1, lr = 0.1

1st Epoch:

For x1 = 1, x2 = 1

z = w0 + w1*x1 + w2*x2 = -0.3 + 0.2 * 1 + (-0.1) * 1 = -0.2

output = f(z) = f(-0.2) = 0

target = 1 (Weight Update Required)

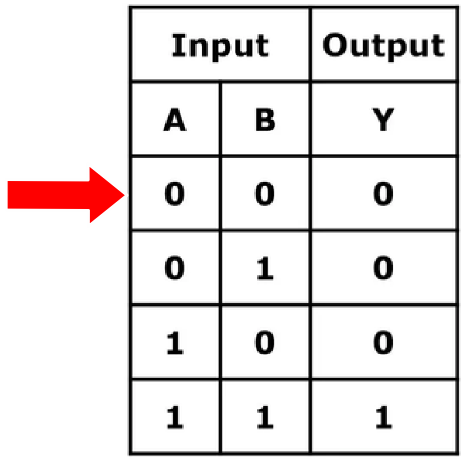
Weight Update:

w0 = -0.3 + 0.1*(1-0)*1= -0.2

w1 = 0.2 + 0.1*(1-0)*1 = 0.3

w2 = -0.1 + 0.1*(1-0)*1 = 0.0

Building a Logical AND



Input		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

We have:

w0 = -0.2, w1 = 0.3, w2 = 0.0, lr = 0.1

$$w_i \leftarrow w_i + \Delta w_i$$

2nd Epoch:

$$\Delta w_i = \eta(t - o)x_i$$

For x1 = 0, x2 = 0

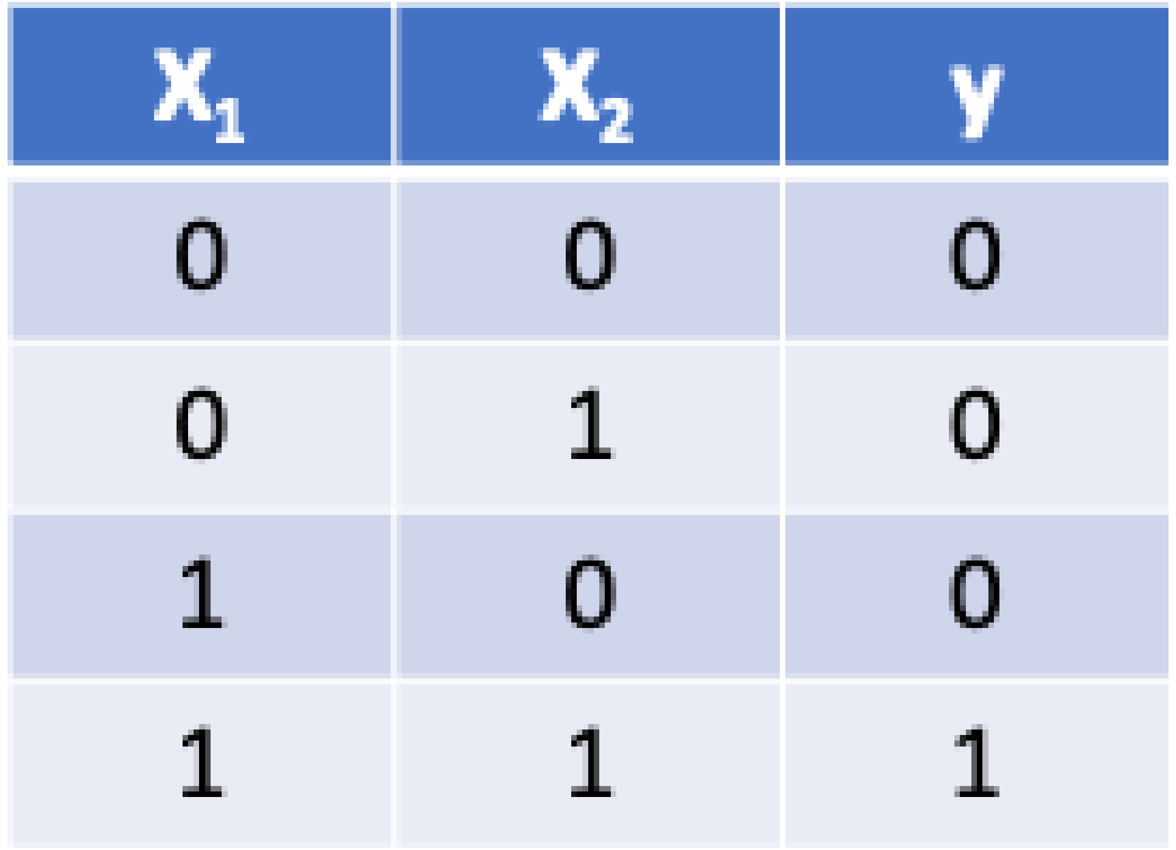
z = w0 + w1*x1 + w2*x2 = -0.2 + 0.3 * 0 + 0.0 * 0 = -0.2

output = f(z) = f(-0.2) = 0

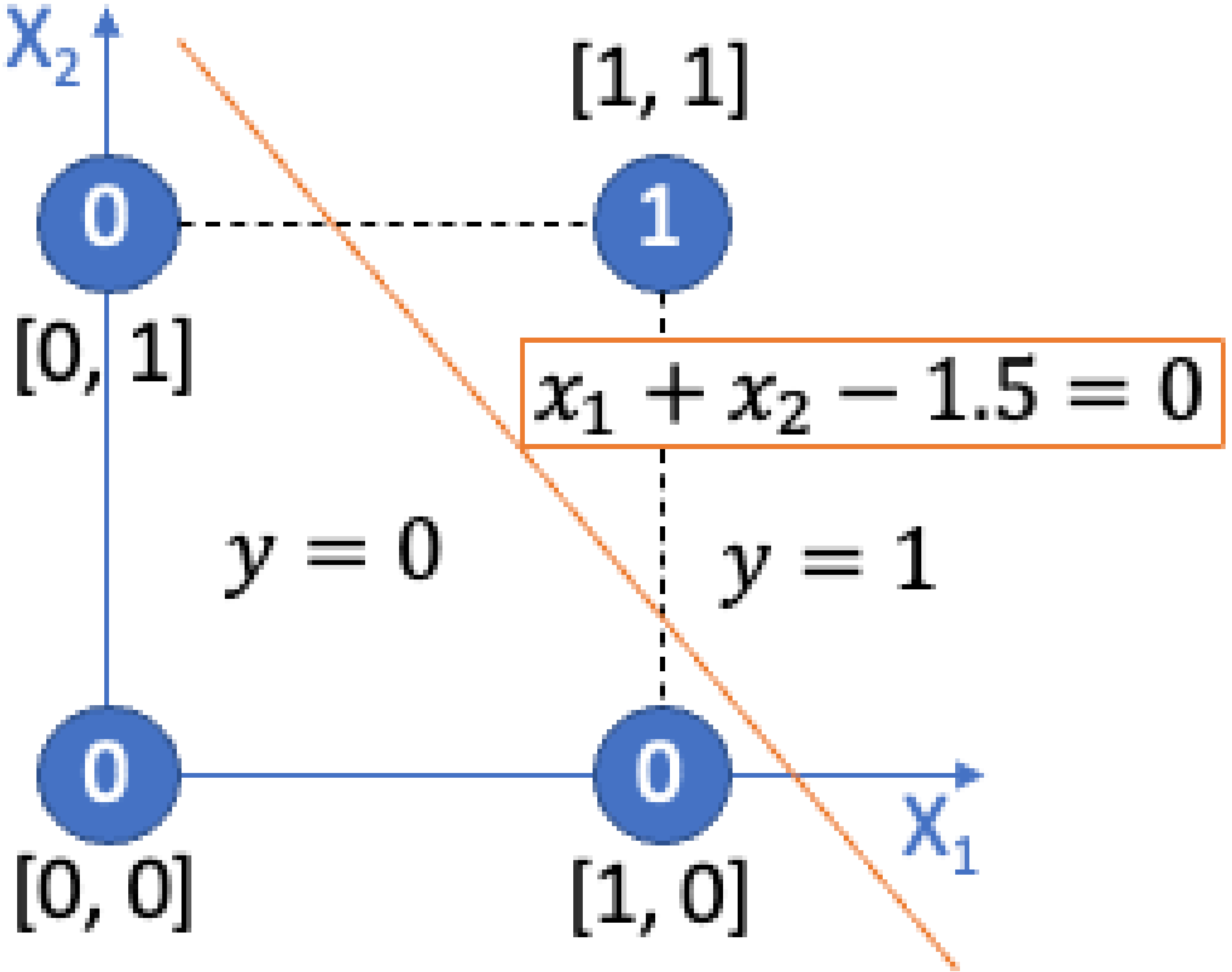
target = 0 (No weight update required)

Iterate Until Convergence or Minimum Error

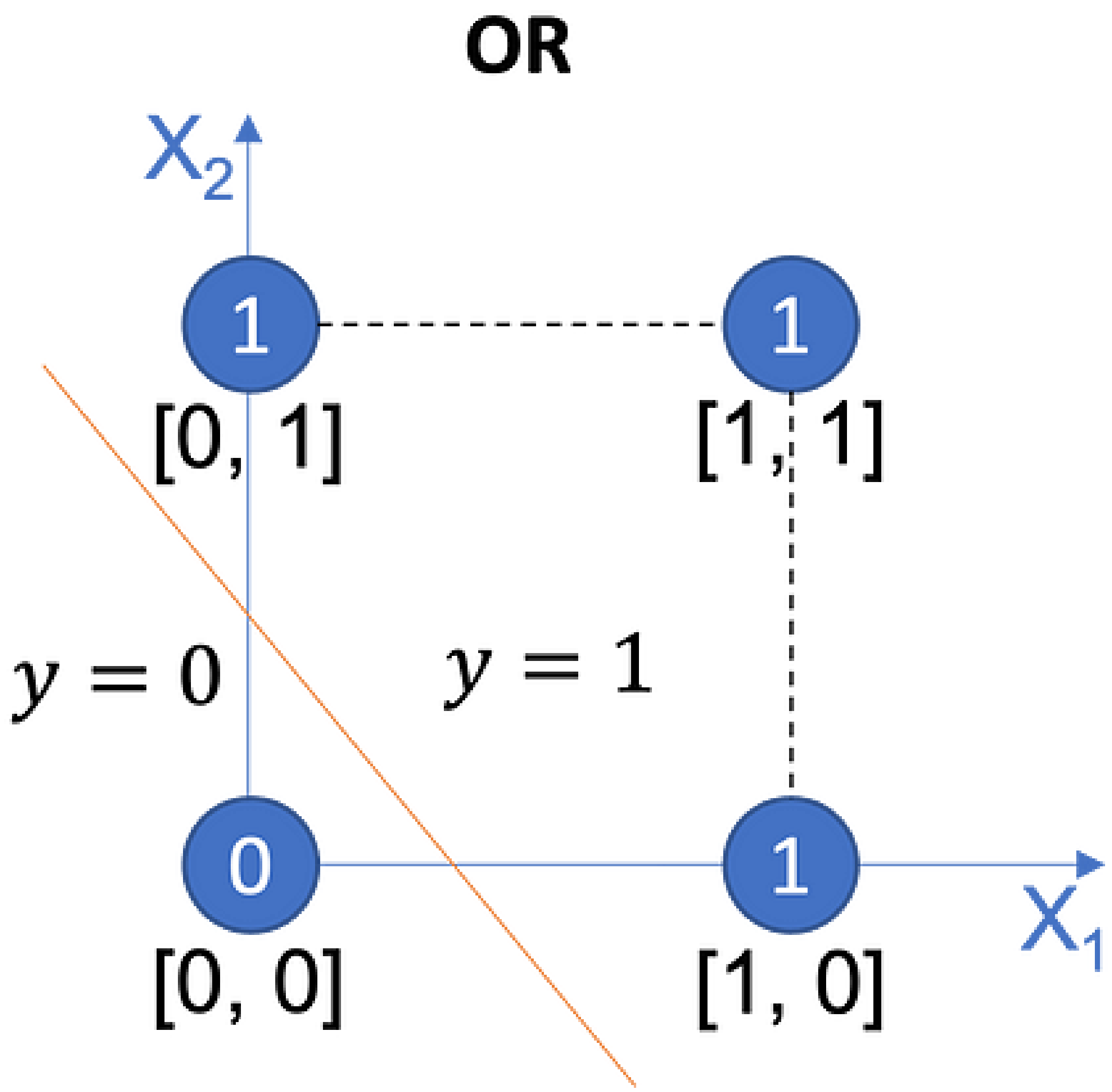
Building a Logical AND



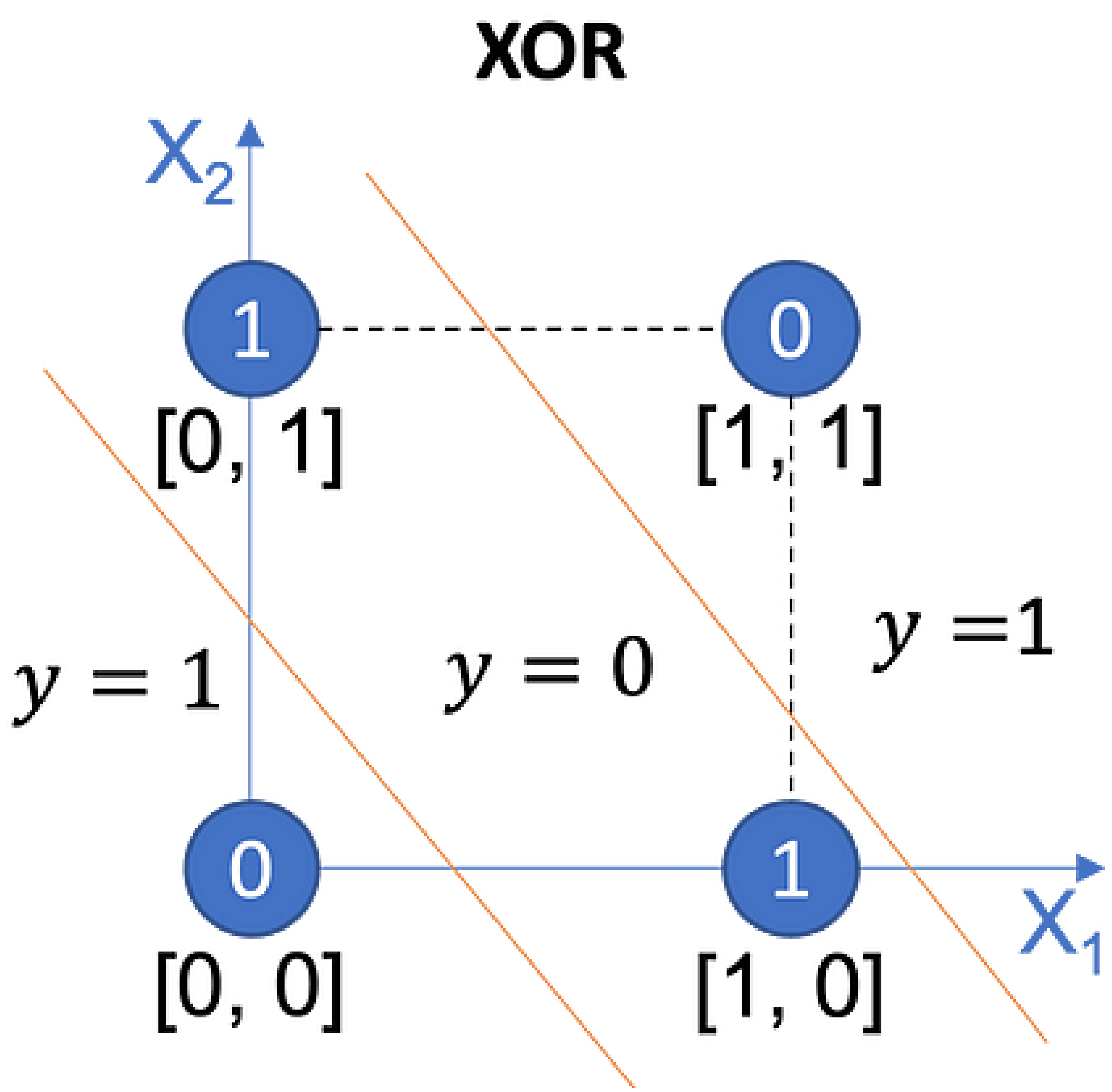
x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1



Logical OR

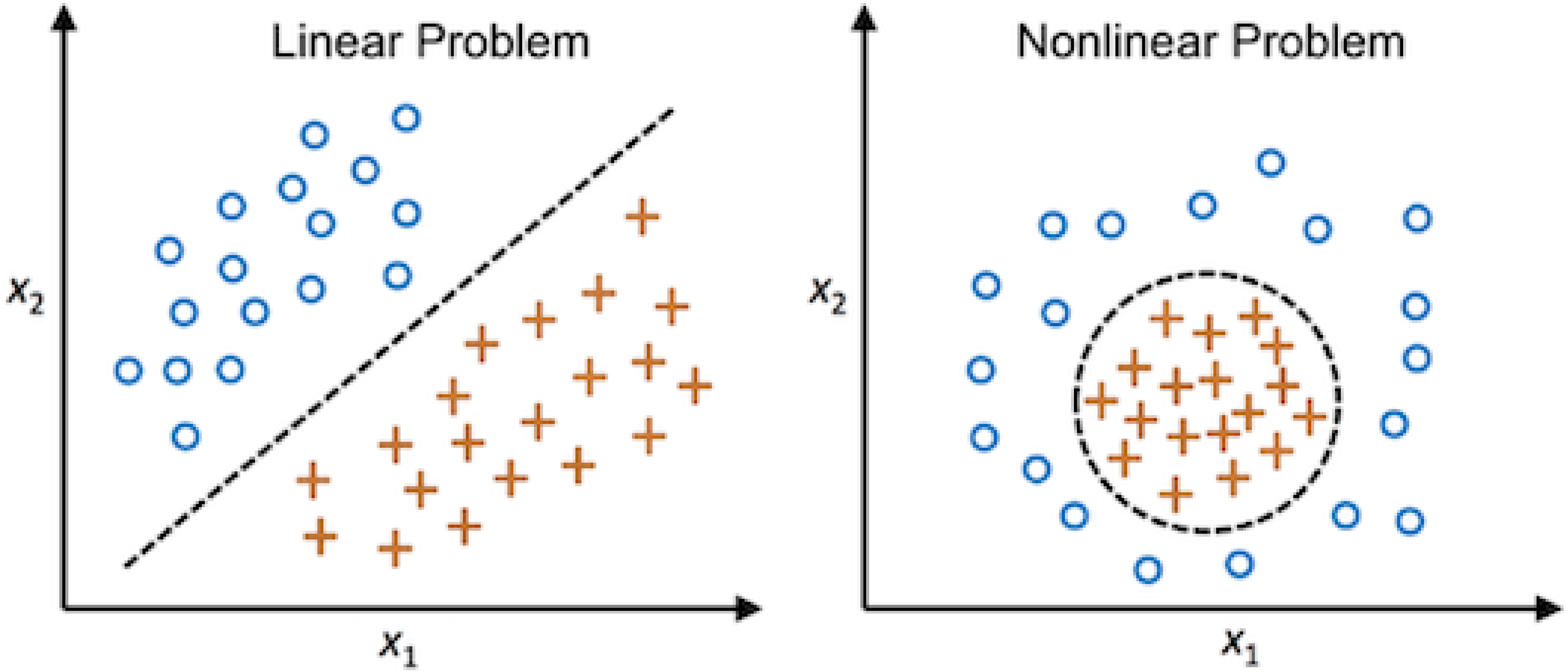


Logical XOR??

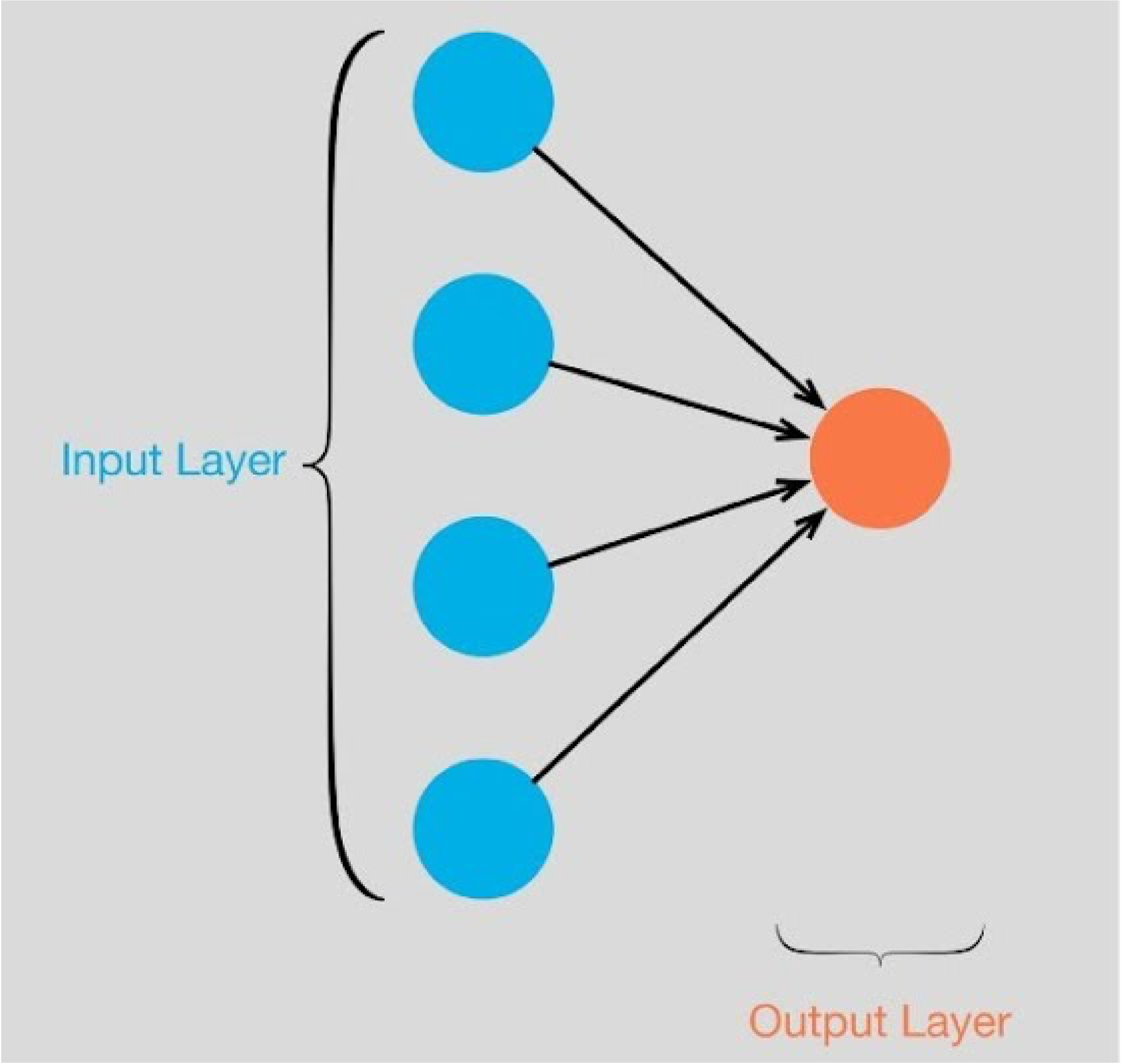


Not Linearly Separable!

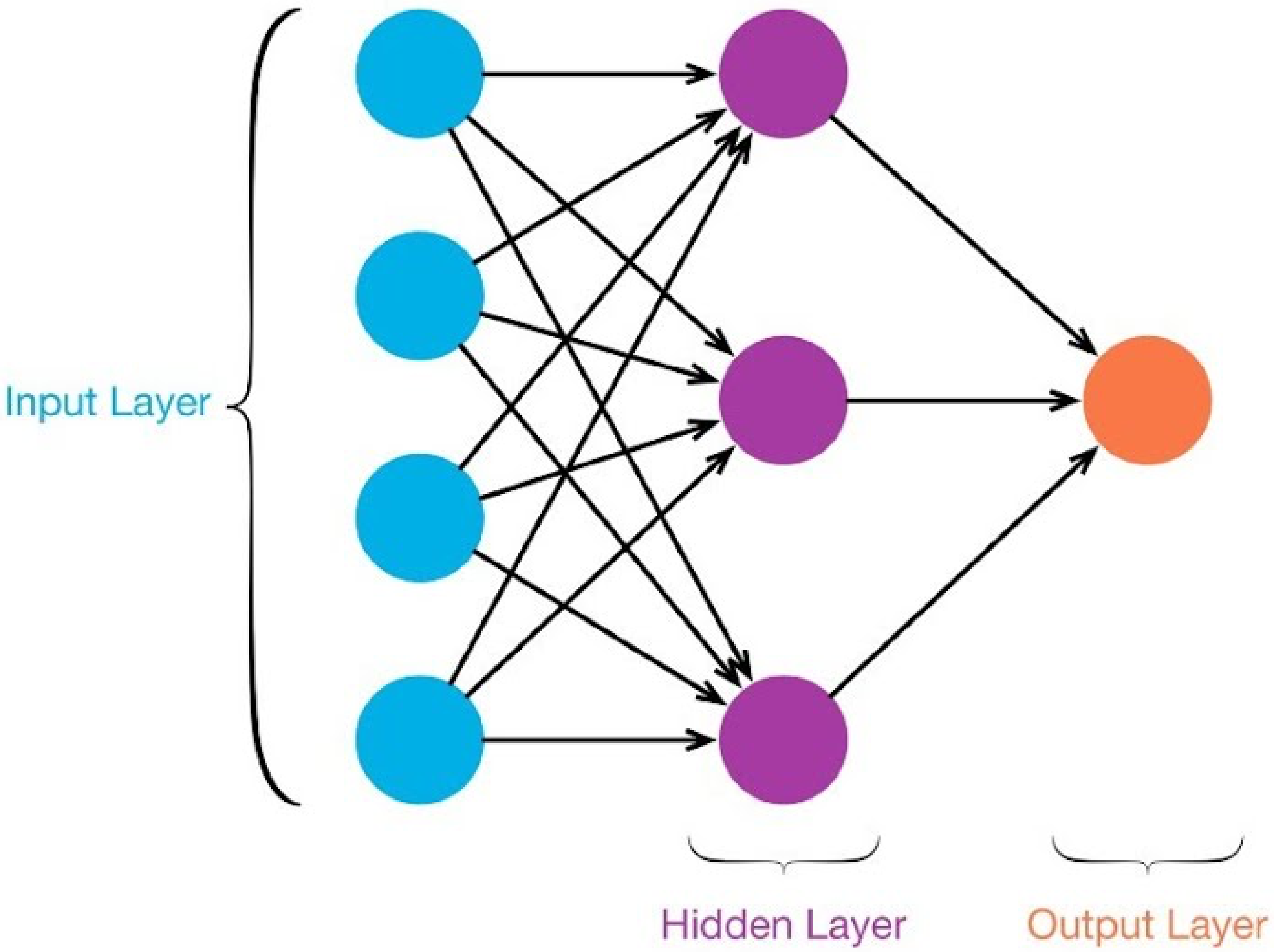
Linear vs Non-Linear Problem



Types of Perceptron



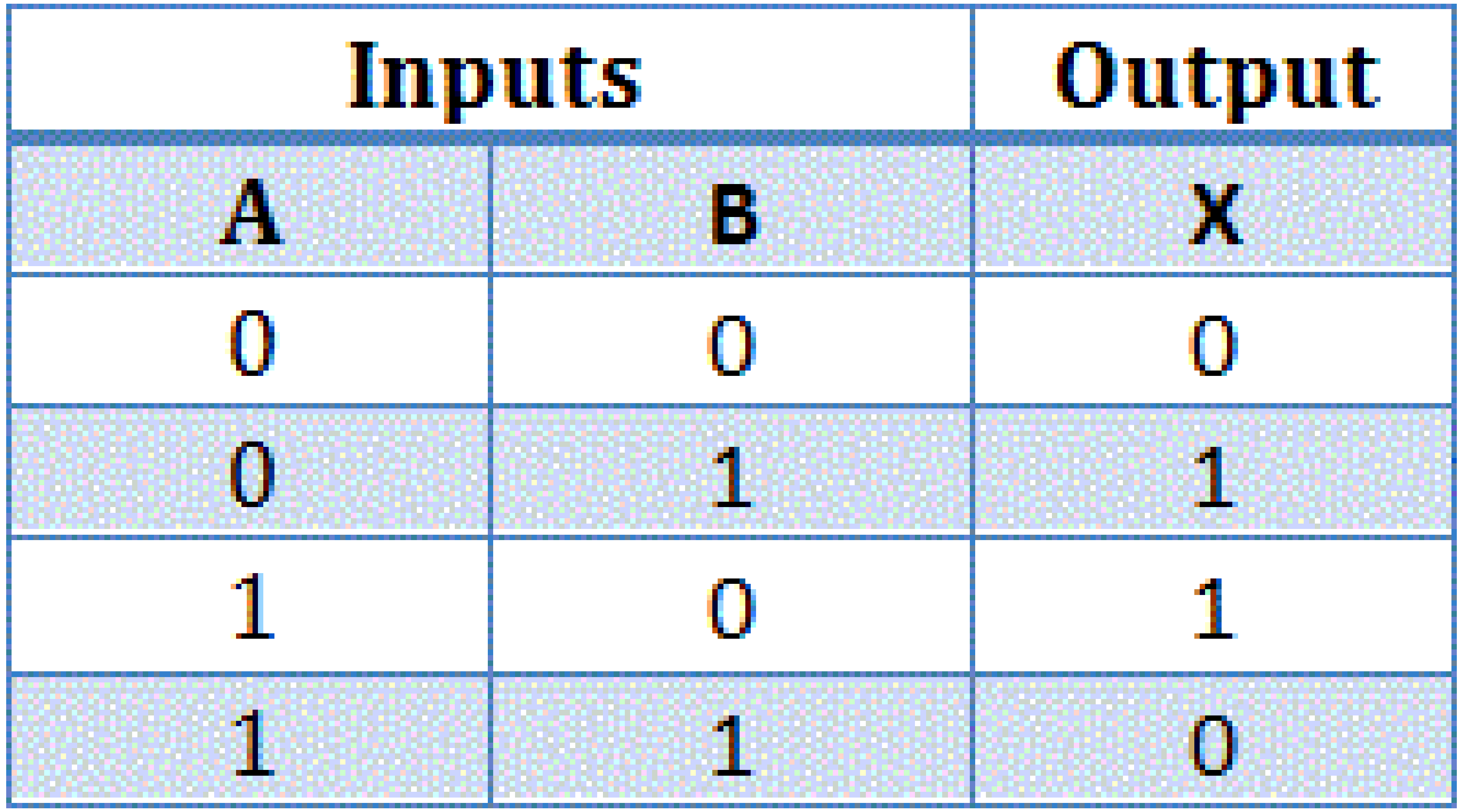
Single Layered Perceptron



Multi Layered Perceptron

Lets Solve XOR Problem using MultiLayer Perceptron Approach!

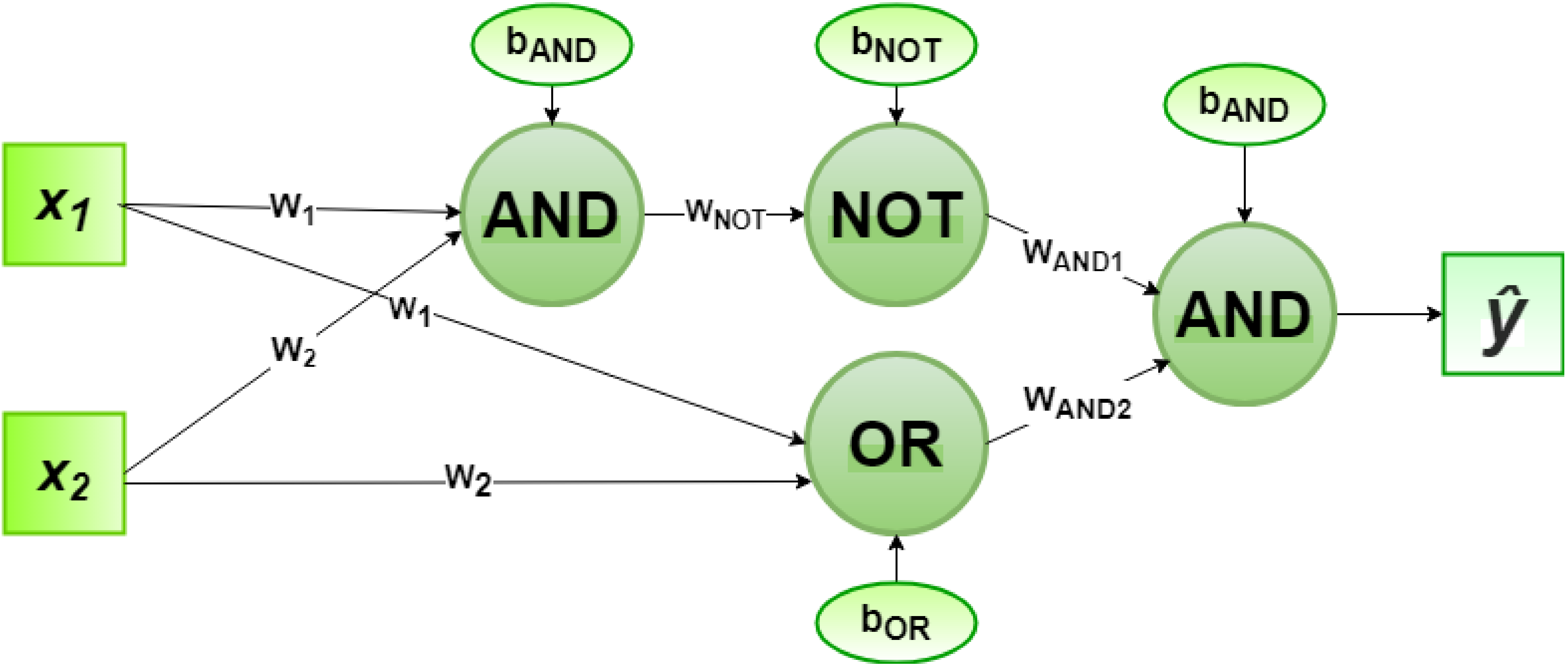
Building a Logical XOR



Inputs		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

$$\mathrm{XOR}(A,B)=(A\land\lnot B)\lor(\lnot A\land B)$$

Building a Logical XOR



$$XOR(A,B)=(A\land\neg B)\lor(\neg A\land B)$$

SOOOOO
ANY QUESTIONS???

Thank you!