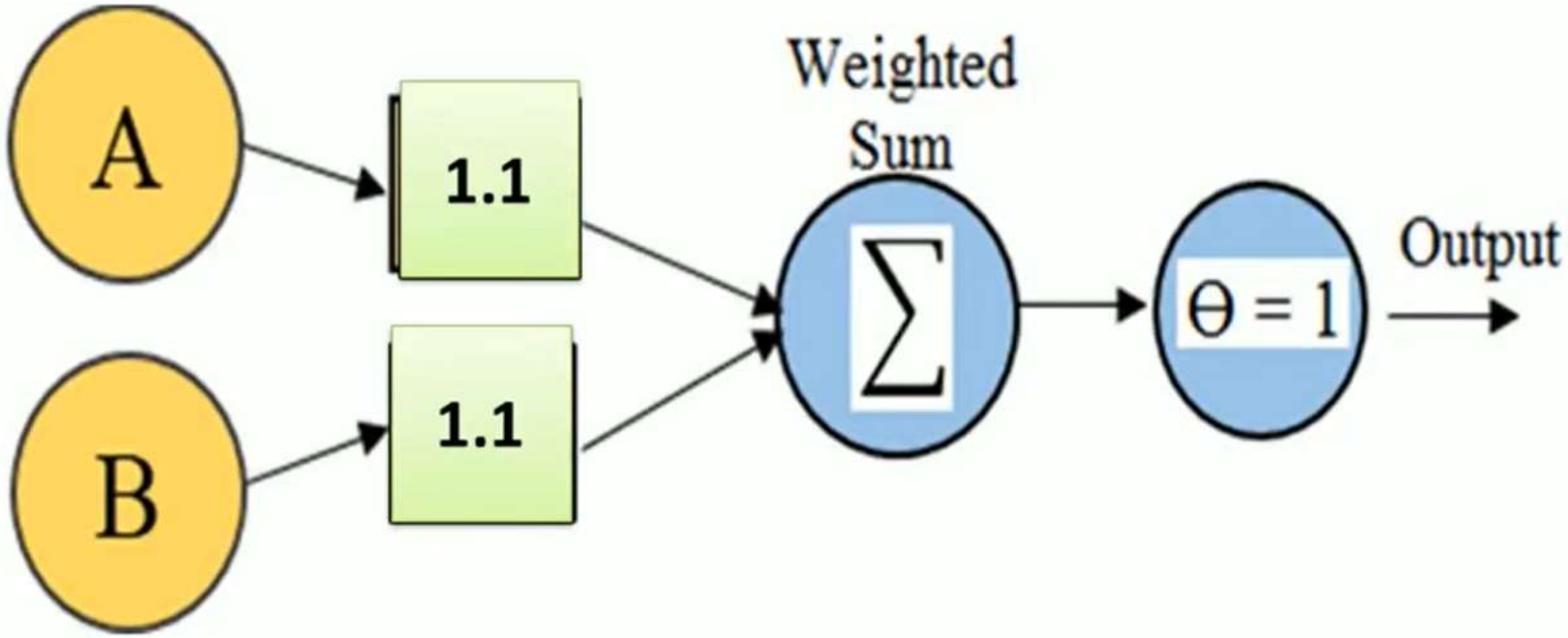
Artificial intelligence

Rajorshi Bhattacharya, Lecturer (CSE), EU

1) https://youtu.be/uxmNDNb0u9A
2) https://youtu.be/zmIzNBMsQYQ
3) https://youtu.be/uY8BS-nruAA

LOGICAL "OR" GATE

PERCEPTRON TRAINING RULE



Artificial Neural Networks

OR Gate - PERCEPTRON TRAINING RULE

$w1 = 0.6, w2 = 0.6$ Threshold $= 1$ and Learning Rate $n = 0.5$

Threshold is also called as “Activation Value”.

This will be given as a question.

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

- If (wi*xi) >= Threshold, then Output = 1
- else if (wi*xi) < Threshold, then Output = 0

OR Gate - PERCEPTRON TRAINING RULE

w1 = 0.6, w2 = 0.6 Threshold = 1 and Learning Rate n = 0.5

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

1. A=0, B=0 and Target = 0
   - $wi.xi = 0*0.6 + 0*0.6 = 0$
   - This is not greater than the threshold of 1, so the output = 0

OR Gate - PERCEPTRON TRAINING RULE

w1 = 0.6, w2 = 0.6 Threshold = 1 and Learning Rate n = 0.5

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

1. A=0, B=0 and Target = 0
   - $wi.xi = 0*0.6 + 0*0.6 = 0$
   - This is not greater than the threshold of 1, so the output = 0
2. A=0, B=1 and Target = 1
   - $wi.xi = 0*0.6 + 1*0.6 = 0.6$
   - This is not greater than the threshold of 1, so the output = 0

OR Gate - PERCEPTRON TRAINING RULE

w1 = 0.6, w2 = 0.6 Threshold = 1 and Learning Rate n = 0.5

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

2. A=0, B=1 and Target = 1

- $wi.xi = 0*0.6 + 1*0.6 = 0.6$
- This is not greater than the threshold of 1, so the output = 0

$$wi = wi + n(t - o)xi$$

$$w1 = 0.6 + 0.5(1 - 0)0 = 0.6$$

$$w2 = 0.6 + 0.5(1 - 0)1 = 1.1$$

OR Gate - PERCEPTRON TRAINING RULE

w1 = 0.6, w2 = 1.1 Threshold = 1 and Learning Rate n = 0.5

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

1. A=0, B=0 and Target = 0
   - *wi.xi = 0*0.6 + 0*1.1 = 0*
   - This is not greater than the threshold of 1, so the output = 0
2. A=0, B=1 and Target = 1
   - *wi.xi = 0*0.6 + 1*1.1 = 1.1*
   - This is greater than the threshold of 1, so the output = 1

OR Gate - PERCEPTRON TRAINING RULE

w1 = 0.6, w2 = 1.1 Threshold = 1 and Learning Rate n = 0.5

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

3. A=1, B=0 and Target = 1

- $wi.xi = 1*0.6 + 0*1.1 = 0.6$
- This is not greater than the threshold of 1, so the output = 0

$$wi = wi + n(t - o)xi$$

$$w1 = 0.6 + 0.5(1 - 0)1 = 1.1$$

$$w2 = 1.1 + 0.5(1 - 0)0 = 1.1$$

OR Gate - PERCEPTRON TRAINING RULE

w1 = 1.1, w2 = 1.1 Threshold = 1 and Learning Rate n = 0.5

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

1. A=0, B=0 and Target = 0
   - *wi.xi = 0*1.1 + 0*1.1 = 0*
   - This is not greater than the threshold of 1, so the output = 0
2. A=0, B=1 and Target = 1
   - *wi.xi = 0*1.1 + 1*1.1 = 1.1*
   - This is greater than the threshold of 1, so the output = 1

OR Gate - PERCEPTRON TRAINING RULE

w1 = 1.1, w2 = 1.1 Threshold = 1 and Learning Rate n = 0.5

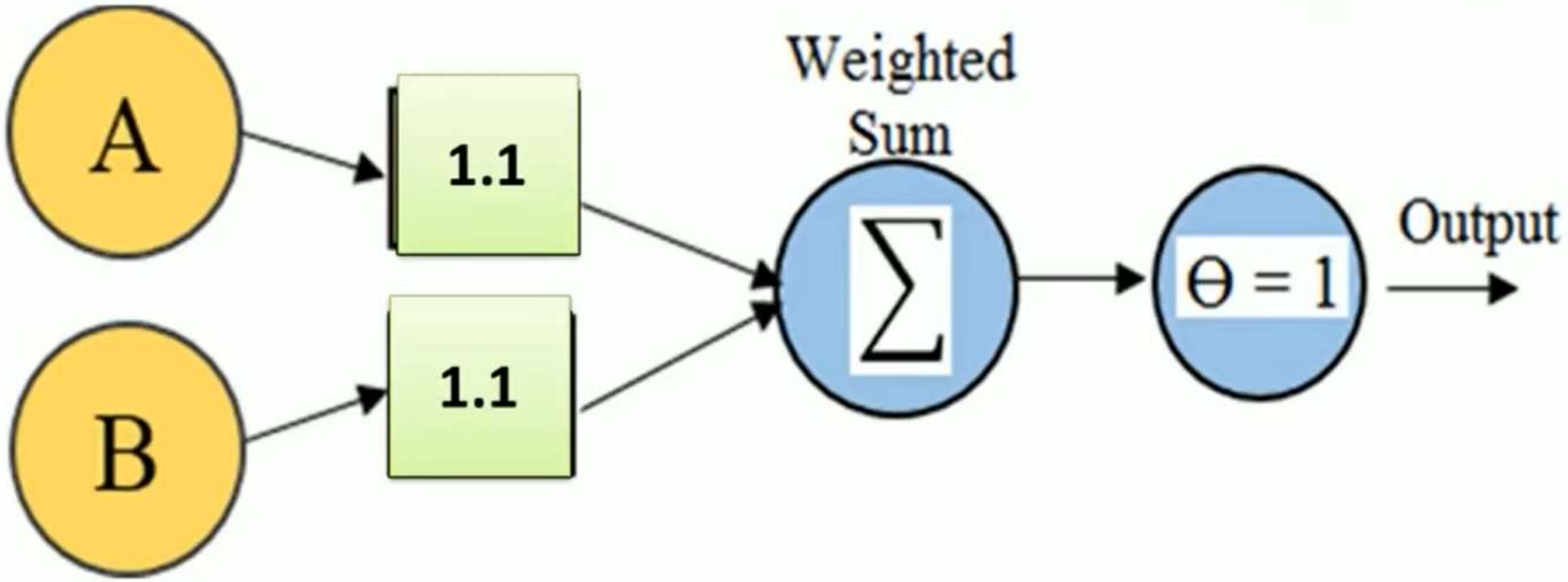
A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

3. A=1, B=0 and Target = 1
   - *wi.xi = 1*1.1 + 0*1.1 = 1.1*
   - This is greater than the threshold of 1, so the output = 1
4. A=1, B=1 and Target = 1
   - *wi.xi = 1*1.1 + 1*1.1 = 2.2*
   - This is greater than the threshold of 1, so the output = 1

OR Gate - PERCEPTRON TRAINING RULE

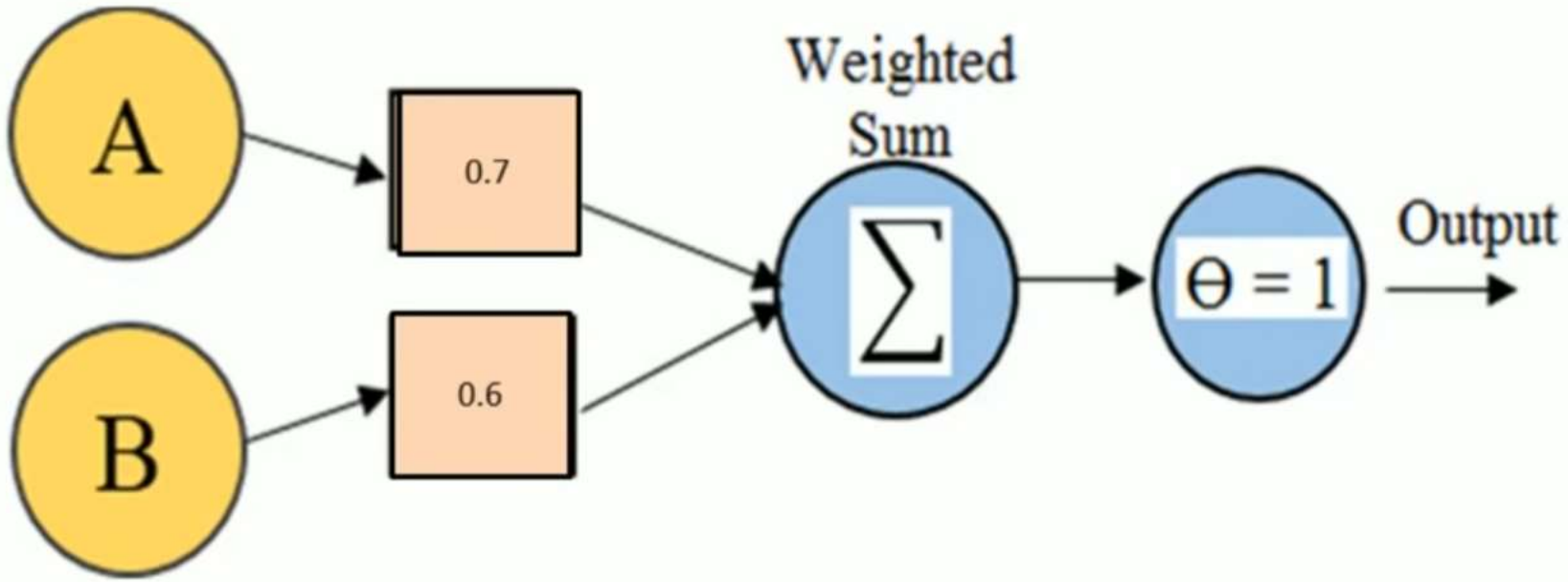
w1 = 1.1, w2 = 1.1 Threshold = 1 and Learning Rate n = 0.5

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1



LOGICAL "AND" GATE

PERCEPTRON TRAINING RULE



Artificial Neural Networks

AND Gate - PERCEPTRON TRAINING RULE

w1 = 1.2, w2 = 0.6 Threshold = 1 and Learning Rate n = 0.5

A	B	A ^ B
0	0	0
0	1	0
1	0	0
1	1	1

1. A=0, B=0 and Target = 0
   - *wi.xi = 0*1.2 + 0*0.6 = 0*
   - This is not greater than the threshold of 1, so the output = 0
2. A=0, B=1 and Target = 0
   - *wi.xi = 0*1.2 + 1*0.6 = 0.6*
   - This is not greater than the threshold of 1, so the output = 0

AND Gate - PERCEPTRON TRAINING RULE

w1 = 1.2, w2 = 0.6 Threshold = 1 and Learning Rate n = 0.5

A	B	A ^ B
0	0	0
0	1	0
1	0	0
1	1	1

3. A=1, B=0 and Target = 0

- *wi.xi = 1*1.2 + 0*0.6 = 1.2*
- This is greater than the threshold of 1, so the output = 1

$$wi = wi + n(t - o)xi$$

$$w1 = 1.2 + 0.5(0 - 1)1 = 0.7$$

$$w2 = 0.6 + 0.5(0 - 1)0 = 0.6$$

AND Gate - PERCEPTRON TRAINING RULE

w1 = 0.7, w2 = 0.6 Threshold = 1 and Learning Rate n = 0.5

A	B	A ^ B
0	0	0
0	1	0
1	0	0
1	1	1

1. A=0, B=0 and Target = 0
   - *wi.xi = 0*0.7 + 0*0.6 = 0*
   - This is not greater than the threshold of 1, so the output = 0
2. A=0, B=1 and Target = 0
   - *wi.xi = 0*0.7 + 1*0.6 = 0.6*
   - This is not greater than the threshold of 1, so the output = 0

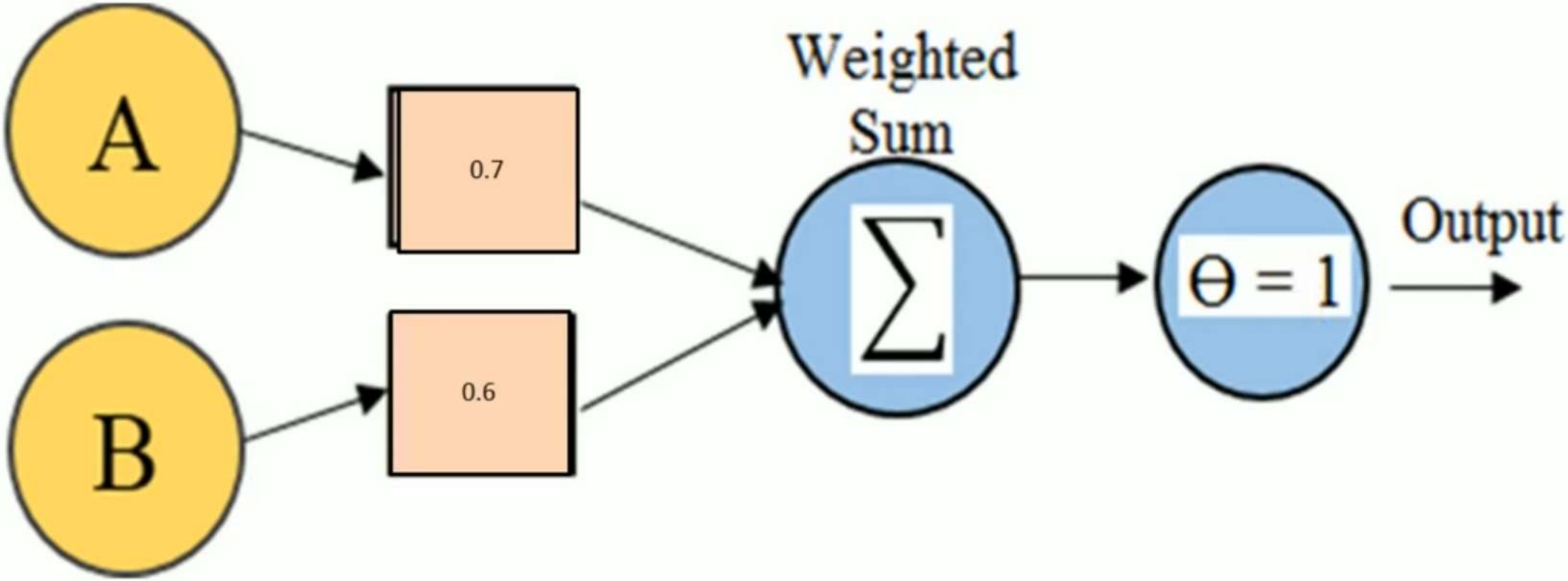
AND Gate - PERCEPTRON TRAINING RULE

w1 = 0.7, w2 = 0.6 Threshold = 1 and Learning Rate n = 0.5

A	B	A ^ B
0	0	0
0	1	0
1	0	0
1	1	1

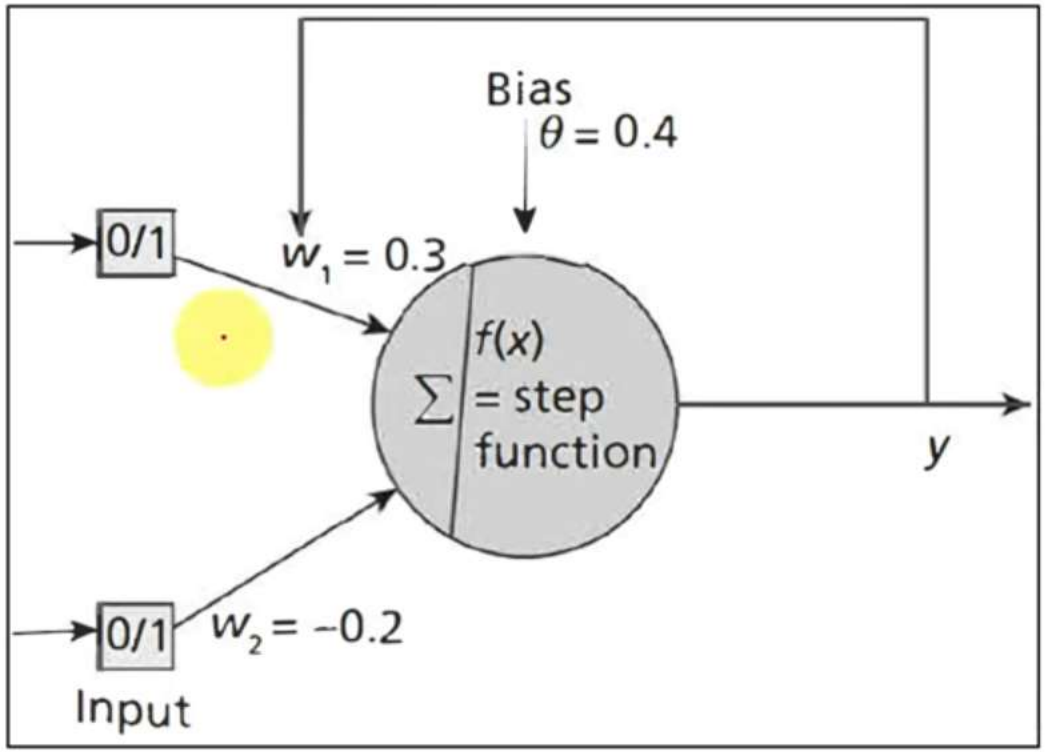
3. A=1, B=0 and Target = 0
   - *wi.xi = 1*0.7 + 0*0.6 = 0.7*
   - This is not greater than the threshold of 1, so the output = 0
4. A=1, B=1 and Target = 1
   - *wi.xi = 1*0.7 + 1*0.6 = 1.3*
   - This is greater than the threshold of 1, so the output = 1

AND Gate - PERCEPTRON TRAINING RULE



Perceptron Algorithm Boolean AND Function

- Consider a perceptron to represent the Boolean function AND with the initial weights $w1 = 0.3,\ w2 = -0.2$, learning rate $\alpha = 0.2$ and bias $\theta = 0.4$.
- The activation function used here is the Step function f(x) which gives the output value as binary, i.e., 0 or 1.
- If value of f(x) is greater than or equal to 0, it outputs 1 or else it outputs 0.
- Design a perceptron that performs the Boolean function AND and update the weights until the Boolean function gives the desired output.



Perceptron Algorithm Boolean AND Function

Epoch 1

Epoch	x_1	x_2	Y_{des}	Y_{est}	Error	w_1	w_2	Status
1	0	0	0	Step $((0 \times 0.3 + 0 \times -0.2) - 0.4) = 0$	0	0.3	–0.2	No change
	0	1	0	Step $((0 \times 0.3 + 1 \times -0.2) - 0.4) = 0$	0	0.3	–0.2	No change
	1	0	0	Step $((1 \times 0.3 + 0 \times -0.2) - 0.4) = 0$	0	0.3	–0.2	No change
	1	1	1	Step $((1 \times 0.3 + 1 \times -0.2) - 0.4) = 0$	1	0.5	0	Change

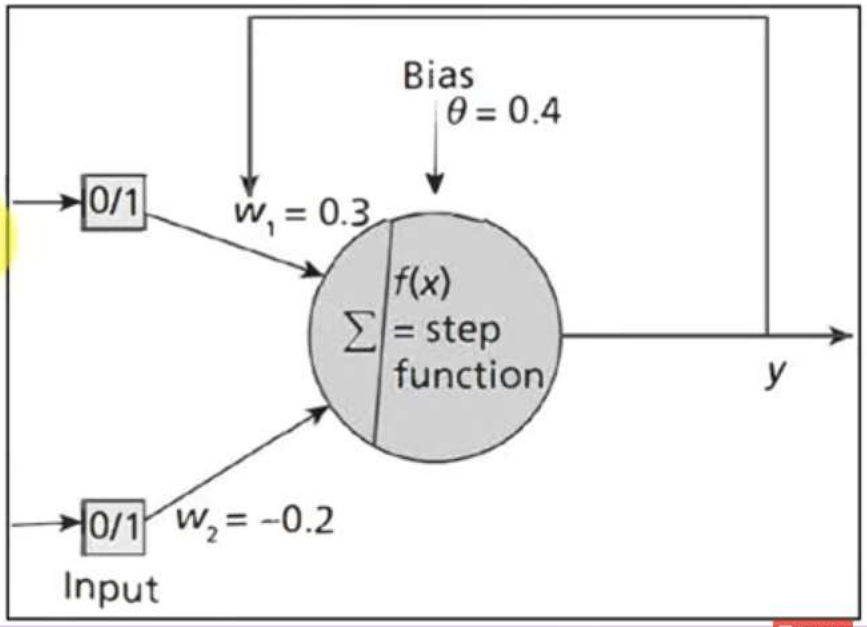
For input (1, 1) the weights are updated as follows:

$$\Delta w_1 = \propto \times e(t) \times x_1 = 0.2 \times 1 \times 1 = 0.2$$

$$w_1 = w_1 + \Delta w_1 = 0.3 + \Delta w_1 = 0.3 + 0.2 = 0.5$$

$$\Delta w_2 = \propto \times e(t) \times x_2 = 0.2 \times 1 \times 1 = 0.2$$

$$w_2 = w_2 + \Delta w_1 = -0.2 + \Delta w_2 = -0.2 + 0.2 = 0$$



Perceptron Algorithm Boolean AND Function

Epoch 2

Epoch	x_1	x_2	Y_{des}	Y_{est}	Error	w_1	w_2	Status
2	0	0	0	Step $((0 \times 0.5 + 0 \times 0) - 0.4) = 0$	0	0.5	0	No change
	0	1	0	Step $((0 \times 0.5 + 1 \times 0) - 0.4) = 0$	0	0.5	0	No change
	1	0	0	Step $((1 \times 0.5 + 0 \times 0) - 0.4) = 1$	–1	0.3	0	Change
	1	1	1	Step $((1 \times 0.3 + 1 \times 0) - 0.4) = 0$	1	0.5	0.2	Change

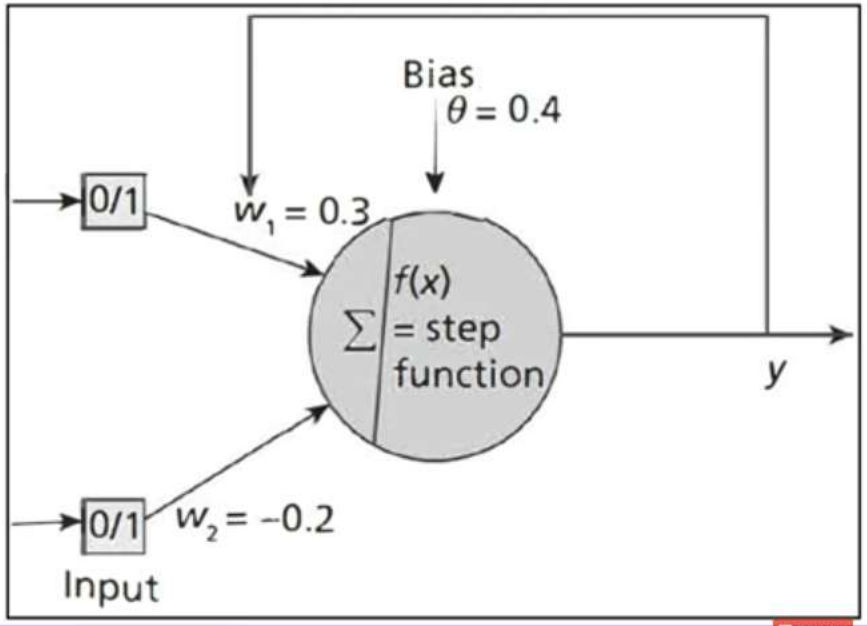
For input (1, 0) the weights are updated as follows:

$$\Delta w_1 = \propto \times e(t) \times x_1 = 0.2 \times -1 \times 1 = -0.2$$

$$w_1 = w_1 + \Delta w_1 = 0.5 + \Delta w_1 = 0.5 - 0.2 = 0.3$$

$$\Delta w_2 = \propto \times e(t) \times x_2 = 0.2 \times -1 \times 0 = 0$$

$$w_2 = w_2 + \Delta w_2 = 0 + \Delta w_2 = 0 + 0 = 0$$



Perceptron Algorithm Boolean AND Function

Epoch 3

Epoch	x_1	x_2	Y_{des}	Y_{est}	Error	w_1	w_2	Status
3	0	0	0	Step $((0 \times 0.5 + 0 \times 0.2) - 0.4) = 0$	0	0.5	0.2	No change
	0	1	0	Step $((0 \times 0.5 + 1 \times 0.2) - 0.4) = 0$	0	0.5	0.2	No change
	1	0	0	Step $((1 \times 0.5 + 0 \times 0.2) - 0.4) = 1$	–1	0.3	0.2	Change
	1	1	1	Step $((1 \times 0.3 + 1 \times 0.2) - 0.4) = 1$	0	0.3	0.2	No change

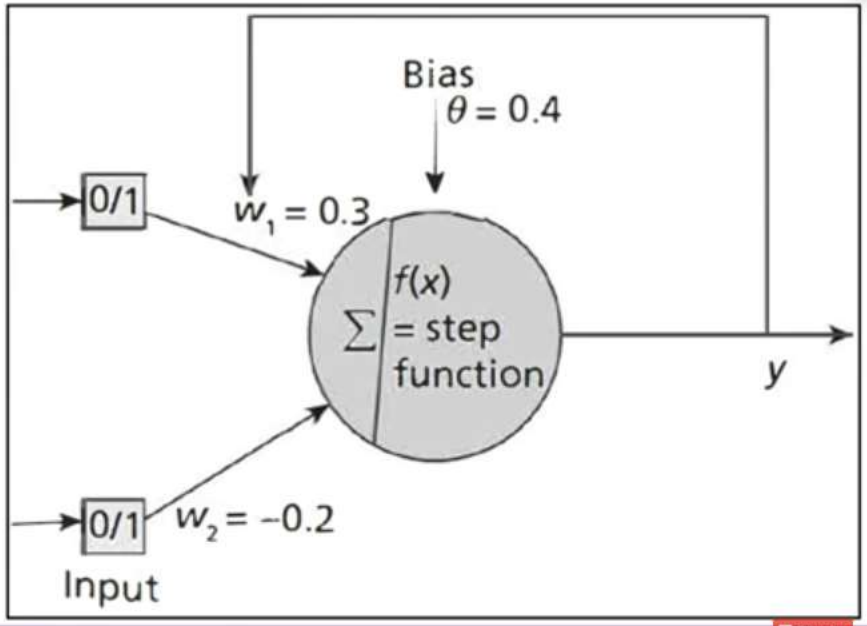
For input (1, 0) the weights are updated as follows:

$$\Delta w_1 = \propto \times e(t) \times x_1 = 0.2 \times -1 \times 1 = -0.2$$

$$w_1 = w_1 + \Delta w_1 = 0.5 + \Delta w_1 = 0.5 - 0.2 = 0.3$$

$$\Delta w_2 = \propto \times e(t) \times x_2 = 0.2 \times -1 \times 0 = 0$$

$$w_2 = w_2 + \Delta w_2 = 0 + \Delta w_2 = 0.2 + 0 = 0.2$$



Perceptron Algorithm Boolean AND Function

Epoch 4

Epoch	x_1	x_2	Y_{des}	Y_{est}	Error	w_1	w_2	Status
4	0	0	0	Step $((0 \times 0.3 + 0 \times 0.2) - 0.4) = 0$	0	0.3	0.2	No change
	0	1	0	Step $((0 \times 0.3 + 1 \times 0.2) - 0.4) = 0$	0	0.3	0.2	No change
	1	0	0	Step $((1 \times 0.3 + 0 \times 0.2) - 0.4) = 0$	0	0.3	0.2	No change
	1	1	1	Step $((1 \times 0.3 + 1 \times 0.2) - 0.4) = 1$	0	0.3	0.2	No change

- **It is observed that with 4 Epochs, the perceptron learns, and the weights are updated to 0.3 and 0.2 with which the perceptron gives the desired output of a Boolean AND function.**