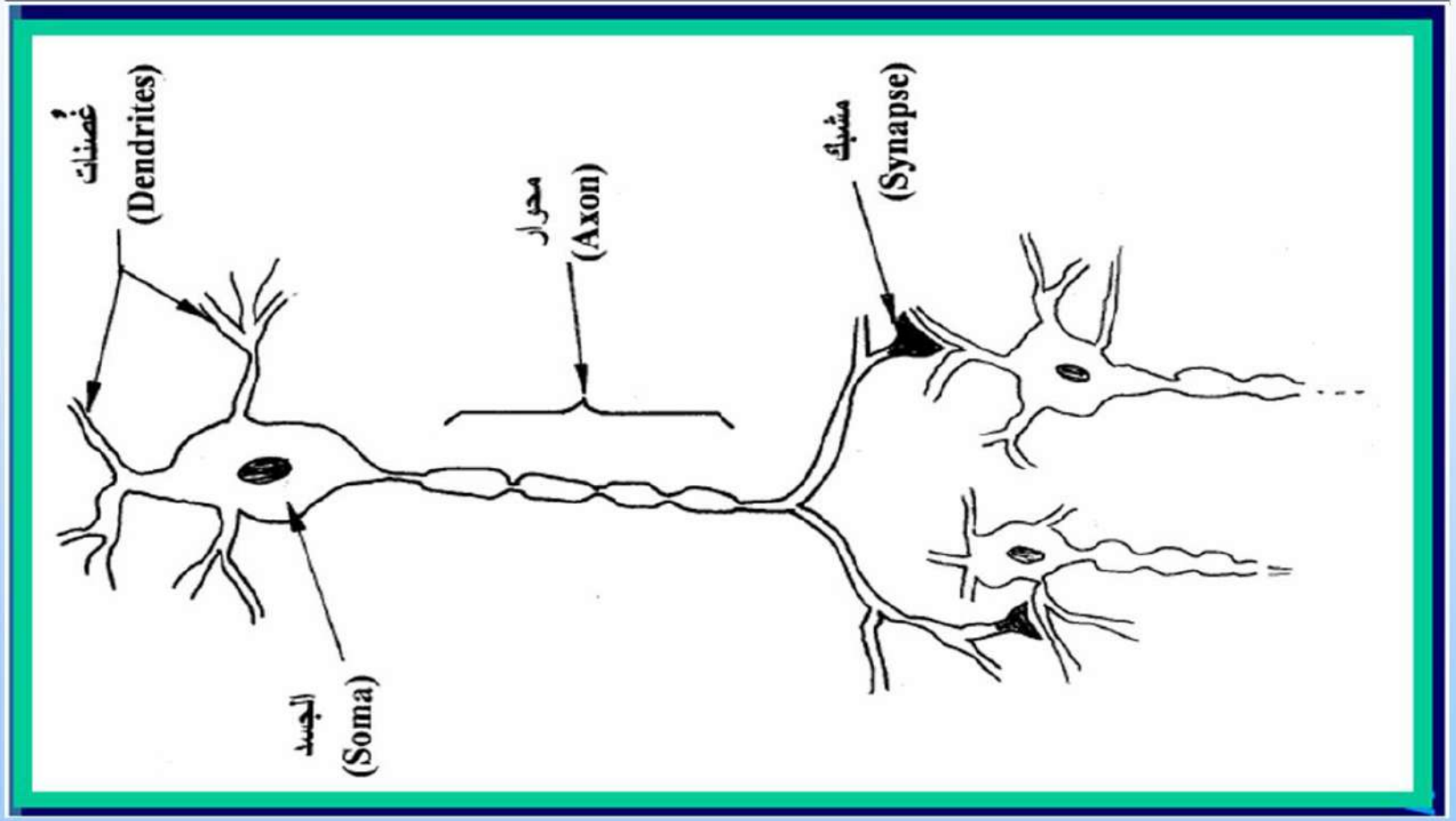
Chapter: 5

Neural Networks

How to the human learn?

The human body consists of millions of neurons. These cells transmit sensations and reactions to and from the human brain via the spinal cord. Through these neurons, knowledge about the outside world is stored in the human mind, by adjusting the weights within these cells

How to the human learn?



How to the human learn?

If we take a simple example, a child learns to identify pictures of animals. If we show this child a picture of a cow, a cat, and a chicken, mentioning the name of each animal in front of it. And we repeated these pictures several times. After that comes the testing stage and we showed the previous pictures with pictures of other animals and let it be a picture of a bird so that it is asked to know the name of the animal that appears in the picture. Chicken picture. This is because the image of the bird is similar in many external characteristics to the image of the chicken which was stored in his mind. But with the diversification of the pictures and their repetition, the child will learn more each time.

How to the human learn?

Scientists thought of a way in which they could simulate this process that occurs in the human mind, and they came up with the science of Neural Networks, which falls under the sciences of artificial intelligence, so that they make computers smart devices that can acquire knowledge in the same way that a person acquires knowledge. It is a method of adjusting weights while learning

A Neural Networks

A neural network consists of a device that contains several processors with several memories connected to each other in parallel, making it perform tasks with the same idea as the neurons in the human brain

A Neural Networks

It is a device designed to simulate the way the human mind performs a particular task. It is a massive processor distributed in parallel, and composed of simple processing units, which stores practical knowledge to make it available to the user by adjusting the weights. Where it can be compared with the neural networks in the human body according to the following:

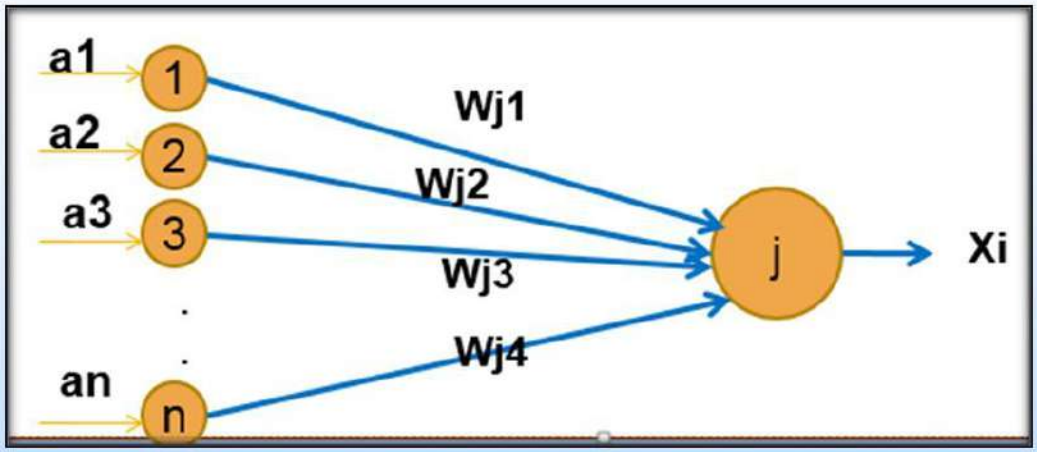
A Neural Networks

A nerve cell consists of nerve endings, nuclei, and axons, where the nerve pathways in the nervous system receive and process electrochemical signals, where these signals are transmitted through the synapses, then the nucleus works to collect those incoming signals and if they are greater than a certain value sufficient to excite the cell, the signals are transmitted Through the sheath the nerve endings then spread to other nerve cells connected to it.

An Artificial Neural Network

The artificial neural network consists of a neuron corresponding to the body of the biological cell (the nucleus), and the nerve endings in the biological cell correspond to the input of the artificial cell, while the output in the artificial cell corresponds to the axon that connects the vital cell with other cells

An Artificial Neural Network



Inputs $(a_1, a_2, \ldots\ldots\ldots\ldots\ldots, a_n)$

Weights $(W_1, W_2, W_3, \ldots\ldots\ldots\ldots W_n)$

Summ Function ($\sum$)

Continue activation or thresholdoutput

The Components of neural Networks

The neural networks consists of input units and processing units in which arithmetic operations are carried out by which the weights are set and through which the appropriate reaction is obtained for each of the inputs to the network

The Components of neural Networks

The input units represent the input layer, and the processing units represent the processing layer, which is the output of the network. And between each of these layers there is a layer of interfaces that connect each layer to the next layer, in which the weights of each interface are adjusted, and the network contains only one layer of input units, but it may contain more than one layer of processing layers.

Applications for neural networks

- Pattern recognition and image recognition.
- The ability to recognize distorted images.
- Classification operations into a number of categories.
- Recognize sounds.

Learning of AI Neural Network

The network learns by giving it a set of examples, which is called a training class. The methods of teaching neural network are divided into two parts:

- Supervised Learning.
- Unsupervised Learning.

Learning of AI Neural Network

- **Supervised Learning.**

In this method, the training class consists of two pairs of vectors, the input vector which is the input values of the network, and the output vector which is the values the network outputs. The weights are modified according to a specific algorithm.

Learning of AI Neural Network

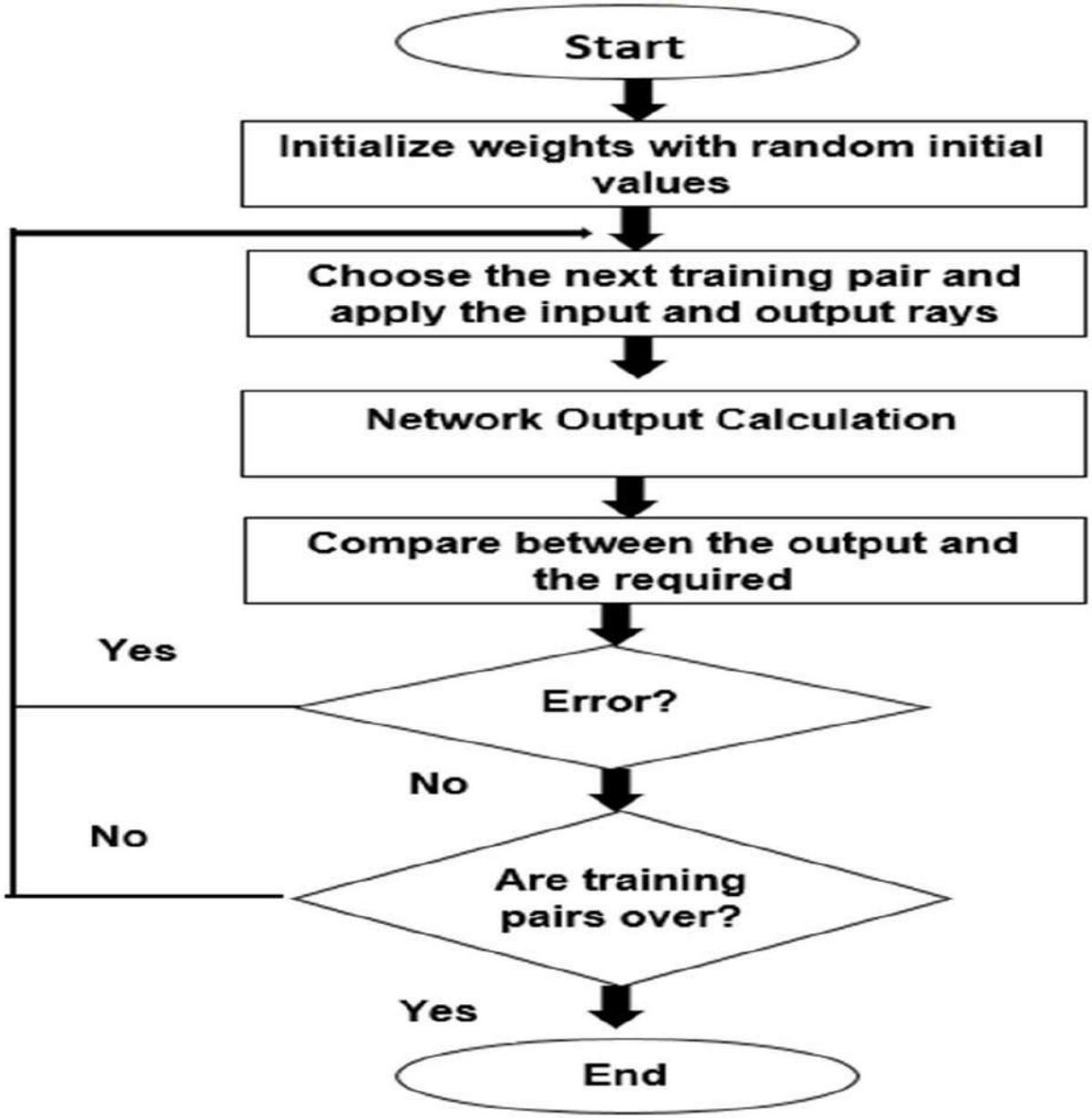
- **Unsupervised Learning**

In this method the training class is an input vector only without displaying the output to the network where the network can modify the weights of its correlations from the stored examples

Learning of AI Neural Network

Therefore, it can be said that training the network is intended to adjust the weights so that applying a set of input values should lead to the production of a set of required output values so that each input ray is linked to an output ray to form a single pair called the training pair.

The training process



How to the neural network work?

There are different types of neural networks and each type of network has its own way of working. In order to explain how the neural network works, we will discuss the idea of one of these networks, the Perceptron network, which is used in applications that need to classify patterns into two categories only

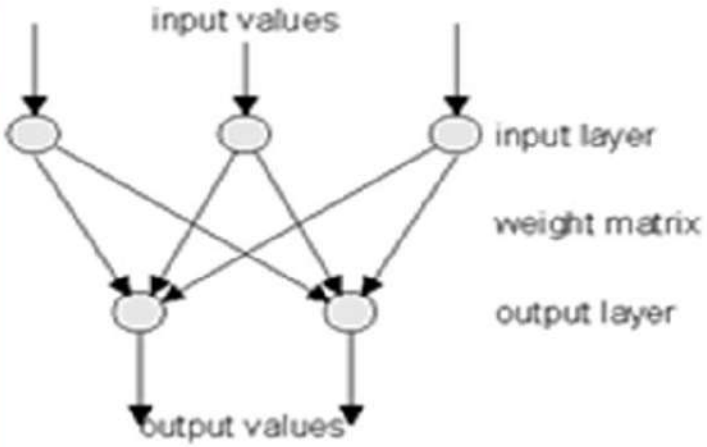
The Perceptron Network

There are two main types of perceptron network:

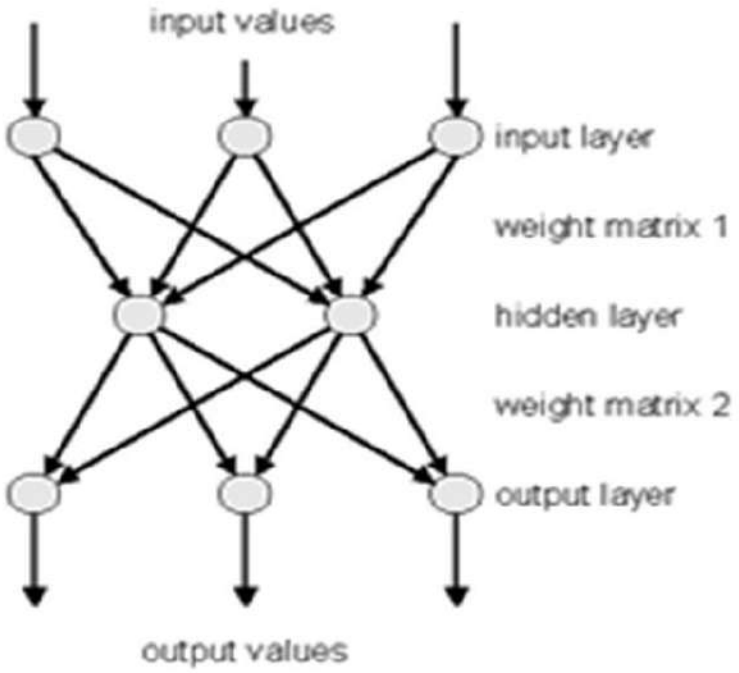
- one layer
- multi-layers

The Perceptron network

Typically, there are two types of perceptrons:



One Layer Perceptron



Multi Layer Perceptron

The Perceptron Network

It is clear that the single layer consists of the input and output layers in addition to the weights matrix that connects these two layers, while the multi-layers have a hidden layer between the input and output layers in addition to two weights matrixes to link this layer with the input and output layers. The basic rule for the perceptron is given as follows:

The Perceptron Network

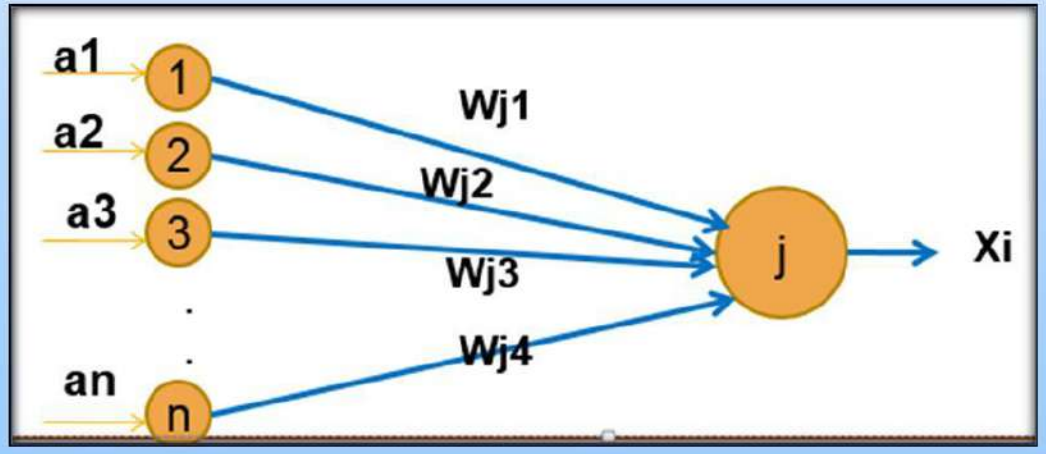
This network consists of the input layer and one or two layers of the processing layer so that the weights are adjusted for only one layer of the interfaces that connect the previous layers, and the idea of this network is summarized in two stages:

- Learning Stage
- Test Stage

Learning Stage

1. Weighted Sum

It is the stage of adjusting the weights until we reach weights capable of giving correct answers.



Learning Stage

It is the process of adding the product of the weights to the inputs of the preceding layer:

$$S_j = \sum a_i * W_{ji}$$

Learning Stage

2. **Transformation**

In which the result of the addition process is converted to one of the values that are supposed to be among the desired network outputs.

For example, if the network were to learn how to classify numbers into odd and even numbers, giving each odd number the value 0 and each even number the value 1.

Learning Stage

The value of S_j is the result of the addition process. It will not always give the value 0 or 1. So it is necessary to convert this output to one of the two values, through the conversion rule that the programmer sets. For example, the rule would be:

If Sj > 0 then Xj = 1

If Sj <= 0 then Xj = 0

Learning Stage

3. weights adjustment

In this stage, the network output is compared to the correct output by subtracting the target and the output from the network output. The network needs to adjust its weights using the following rule learning:

$$W_{ji}\ new = W_{ji}\ old + C(T_j - X_j)^* a_i$$

where C stands for learning rate.

Example of the Perceptron Network

Suppose we have a network of type Perceptron and we want to learn it how to implement the logical AND function, as shown in the following table:

a1	a2	Y
0	0	0
0	1	0
1	0	0
1	1	1

where a1,a2 stands for input, and y stands for output.

Example of the Perceptron Network

And the transformation Rule is:

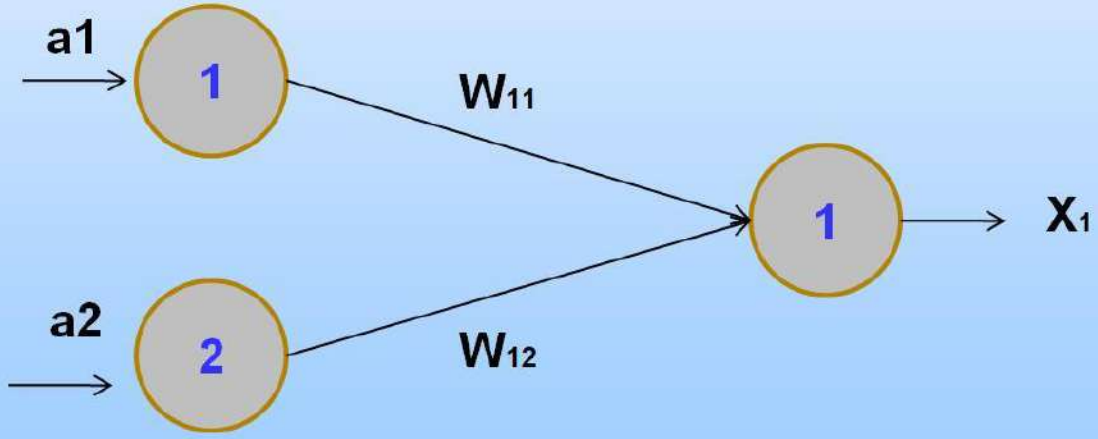
$$\text{If } S_j > 0 \text{ then } X_j = 1$$

$$\text{If } S_j <= 0 \text{ then } X_j = 0$$

We will assume the learning rate at C = 0.5.

Example of the Perceptron Network

To represent this network, we need two input units because there are two inputs, and one processing unit because there is one output.



Example of the Perceptron Network

From the diagram, we find that there is one layer of interfaces and we will impose different weights on them in the range from -0.5 to 0.5.

$$W_{11} = 0.1$$

$$W_{12} = 0.3$$

Example of the Perceptron Network

1. Weighted Sum

$$S_j = \sum a_i w_{ji}$$

$$S_j = 0.0 * 0.1 + 0.0 * 0.3$$

$$S_j = 0.0$$

2. Transformation

$$S_j = 0.0 \rightarrow S <= 0$$

$$X_i = 0 \quad \text{أذا}$$

Example of the Perceptron Network

3. weights adjustment

The output of the first input is 0 and the output of the network is 0 and the product of these two outputs is 0, which indicates that the weights of the network do not need to be adjusted.

Example of the Perceptron Network

Let's move to the second entry, using the same weights as the previous ones, because they were not modified because there is no need for that

1. **Weighted Sum**

$$Sj = \sum ai*wji$$

$$S = 0.0 * 0.1 + 1.0 * 0.3$$

$$S = 0.3$$

2. **Transformation**

$$S = 0.3 \rightarrow S > 0$$

$$X = 1 \quad \text{أذا}$$

Example of the Perceptron Network

3. Weights Adjustment

Where the target product of the second input is 0 also, the net result is 1, by subtracting the net result from the target product gives the value -1 which is not equal to zero so it is necessary to adjust all the weights of the network.

Example of the Perceptron Network

$$W_{ji}new = w_{ji}old + C(t_j - X_j)a_i$$

$$W_{11}new = 0.1 + 0.5 *(-1) * 0.0 \rightarrow W_{11}new = 0.1$$

$$W_{12}new = 0.3 + 0.5 *(-1) * 1.0 \rightarrow W_{12}new = -0.2$$

Example of the Perceptron Network

Thus, we have finished the second entry, let's start with the third, using the weights after adjusting it, following the same previous steps:

1. **Weighted Sum**

$$Sj = \sum ai*wji$$

$$S = 1.0 * 0.1 + 0.0 * (-0.2)$$

$$S = 0.1$$

2. **Transformation**

$$S = 0.1 \rightarrow S > 0$$

$$X = 1 \quad \text{أذا}$$

Example of the Perceptron Network

3. Weights Adjustment

Where the target product of the third input is 0 also, the net result is 1, by subtracting the net result from the target product gives the value -1 which is not equal to zero so it is necessary to adjust all the weights of the network.

Example of the Perceptron Network

$W_{jinew} = w_{jiold} + C(t_j - X_j)a_i$

$W11new = 0.1 + 0.5 *(-1) * 1.0 \rightarrow W11new = -0.4$

$W12new = -0.2 + 0.5 *(-1) * 0.0 \rightarrow W12new = -0.2$

Example of the Perceptron Network

We move to the fourth and final input using weights after modification:

1. **Weighted Sum**

$$Sj = \sum aiwji$$

$$S = 1.0 * -0.4 + 1.0 * (-0.2)$$

$$S = -0.6$$

2. **Transformation**

$$S = -0.6 \rightarrow S <= 0$$

$$X = 0 \quad \text{أذا}$$

Example of the Perceptron Network

3. Weights Adjustment

Where the target product of the fourth input is 0 also, the net result is 1, by subtracting the net result from the target product gives the value -1 which is not equal to zero so it is necessary to adjust all the weights of the network.

Example of the Perceptron Network

$$W_{jinew} = w_{jiold} + C(t_j - X_j)a_i$$

$$W11new = -0.4 + 0.5 * (1) * 1.0 \rightarrow W11new = 0.1$$

$$W12new = -0.2 + 0.5 * (1) * 1.0 \rightarrow W12new = 0.3$$

Example of the Perceptron Network

With this, we have finished showing all the examples on the network called an attempt, in this attempt the network gave only one correct result out of 4 results, meaning that the network has not learned yet and needs other attempts in the same way, so that it can give correct answers to all the examples, then we say that the network learned, and remains only tested

Reasons for not learning the network:

- The quality of the network is not suitable for the application.
- The number of processing units are not suitable.
- The weights with which the network starts are not suitable.
- The learning rate is inappropriate.
- The training category has not been carefully selected.
- The Transformation rule is not suitable

Advantages and Disadvantages of the Perceptron Network

Advantages

- Ease of program design.

Disadvantages:

- It cannot classify patterns into more than two sets .
- The number of processing layers is limited (one or two layers only)