Artificial Intelligence

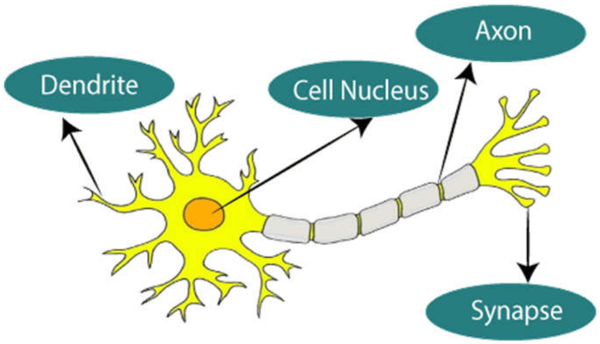
Dr. Mubashir Ahmad (Ph.D.)

Artificial Neural Network

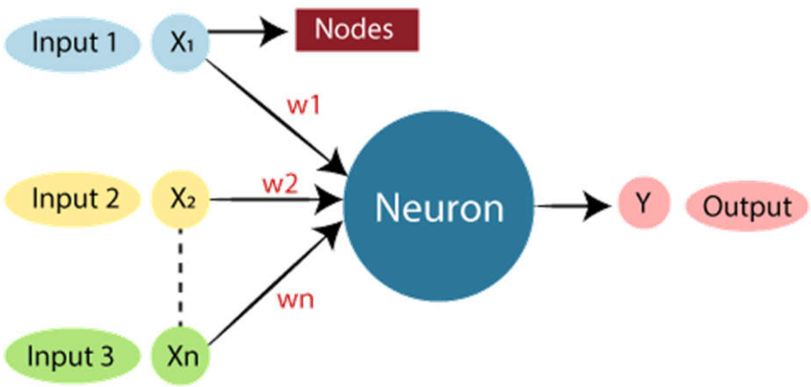
- An artificial neural network (ANN) may be defined as an information-processing model that is inspired by the way biological nervous systems, such as the brain, process information.
- This model tries to replicate only the most basic functions of the brain.
- An ANN is composed of a large number of highly interconnected processing units (neurons) working in unison to solve specific problems.
- Like human being, Artificial neural networks learn by example.
- An ANN is configured for a specific application, such as spam classification, Face Recognition, pattern recognition through a learning process.

Artificial Neural Network

- The term "Artificial Neural Network" is derived from Biological neural networks that develop the structure of a human brain.
- Similar to the human brain which has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks.
- These neurons are known as nodes.

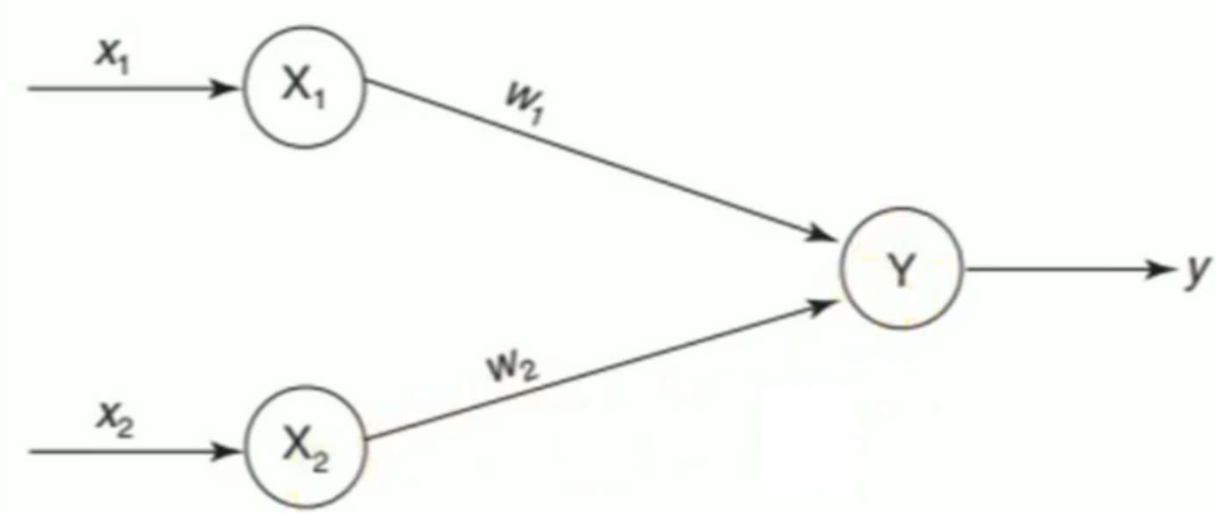


Biological Neural Network.



The typical Artificial Neural Network

Artificial Neural Network



$$y_{in} = x_1 w_1 + x_2 w_2$$

$$y = f(y_{in})$$

Activation function

- Input neurons X1 and X2 are connected to the output neuron Y, over a weighted interconnection links (W1 and W2).
- For the above simple neuron net architecture, the net input has to be calculated in the following way:

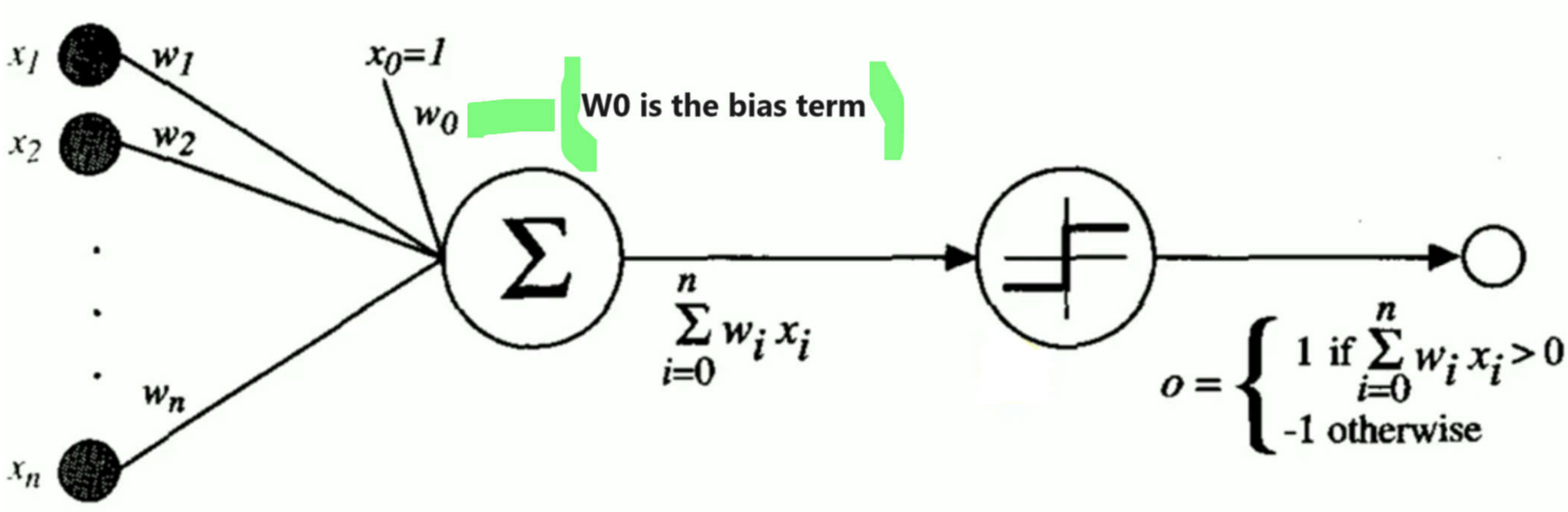
Perceptron in Artificial Neural Network

- A perceptron unit is used to build the ANN system.
- A perceptron takes a vector of real-valued inputs, calculates a linear combination of these inputs, then outputs a 1 if the result is greater than some threshold and -1 otherwise.
- More precisely, given inputs ***x1*** through ***xn,*** the output ***o(x1, . . . , xn)*** computed by the perceptron is

$$o(x_1, \ldots, x_n) = \begin{cases} 1 \text{ if } w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n > 0 \\ -1 \text{ otherwise} \end{cases}$$

- where each ***wi*** is a real-valued constant, or weight, that determines the contribution of input ***xi*** to the perceptron output

Perceptron in Artificial Neural Network



How to update the weights in perceptron?

- One way to learn an acceptable weight vector is to begin with random weights, then iteratively apply the perceptron to each training example, modifying the perceptron weights whenever it misclassifies an example.
- This process is repeated, iterating through the training examples as many times as needed until the perceptron classifies all training examples correctly.
- Weights are modified at each step according to the ***perceptron training rule,*** which revises the weight ***wi*** associated with input ***xi*** according to the rule

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

where

LR

Actual output

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

Target

How to update the weights in perceptron?

Perceptron_training_rule *(X, η)*

initialize **w** *(wi ← an initial (small) random value)*

repeat

  for each training instance (**x**, **tx**) *∈ X*

input /label

    compute the real output **ox** *= Activation(Summation(***w.x***))*

    if (**tx** *≠* **ox**)

      for each wi

        $wi \leftarrow wi + \Delta wi$

        $\Delta wi \leftarrow \eta\,(tx - ox)xi$

      end for

    end if

  end for

until all the training instances in X are correctly classified

return **w** → learned parameters

Perceptron Training rule OR Gate Example 1....

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

Perceptron Training rule OR Gate Example 1.... Update weights

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

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

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

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

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

Perceptron Training rule OR Gate Example 1…. Again from start

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

Perceptron Training rule OR Gate Example 1…. Weights updating….

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
   - $wi.xi = 1*0.6 + 0*1.1 = 0.6$

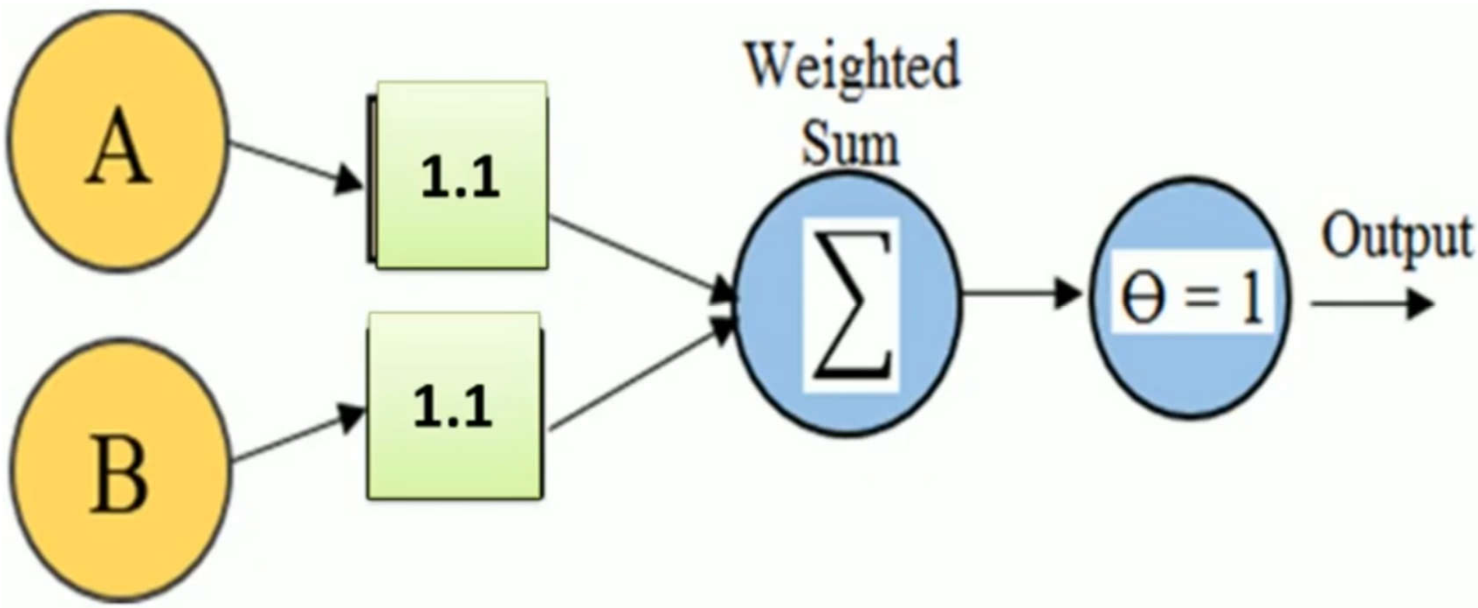
This is not greater than the threshold of 1, so the output = 0

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

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

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

Perceptron Training rule OR Gate Example 1....



Perceptron OR Gate Example 1…. Weights updating apply again from start….

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

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 ✓

Perceptron OR Gate Example 1…. Weights updating apply again from start….

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

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

Perceptron Training Rule AND Gate Example 2….

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

Perceptron Training Rule AND Gate Example 2....

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

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

update weights

Perceptron Training Rule AND Gate Example 2.... Weights update

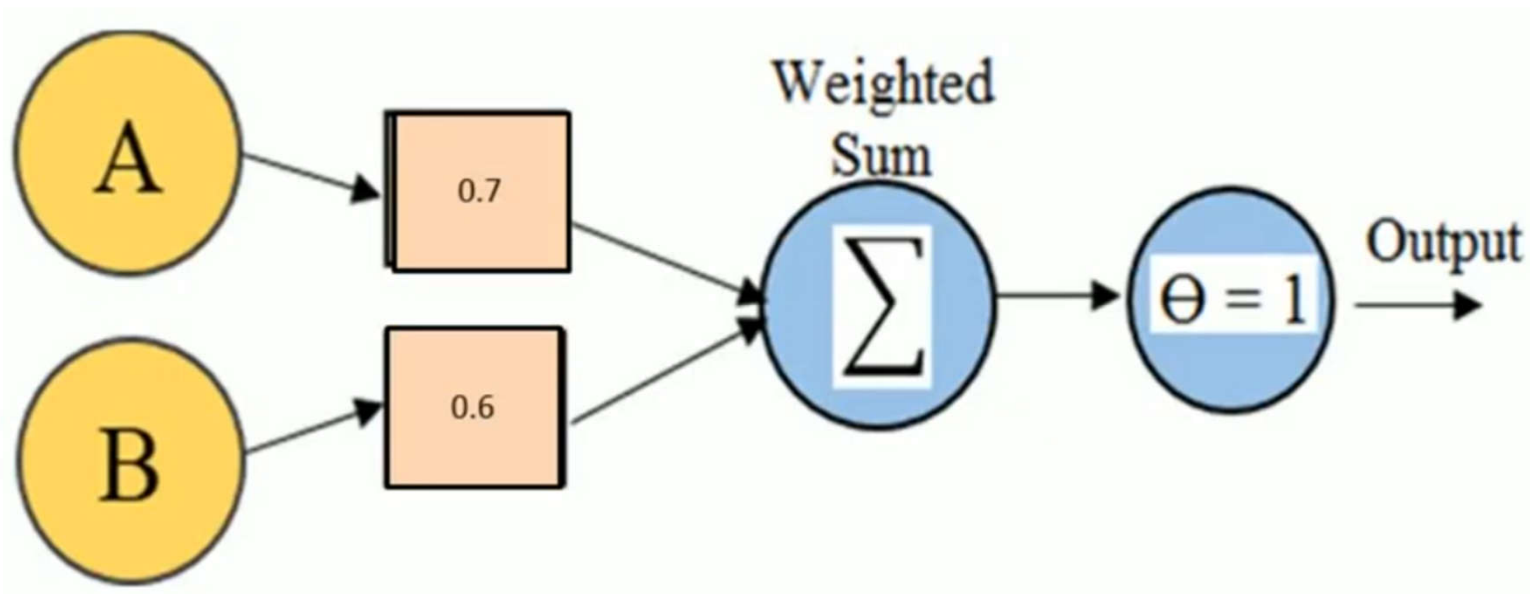
$$wi = wi + n(t - o)xi$$

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

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

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

Perceptron Training Rule AND Gate Example 2….



Perceptron Training Rule AND Gate Example 2.... Updated Weights

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* $< 1 = 0$

Perceptron Training Rule AND Gate Example 2…. Updated Weights

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

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

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