

# XOR-Gate with Multilayer Perceptron



Mehedee Hassan · [Follow](#)

Published in Analytics Vidhya

4 min read · Aug 18, 2021



Listen



Share



More

*In this article, I will explain a simple mathematical calculation for perceptrons to represent a logical XOR-Gate.*

## Introduction

Though this is a simple concept, a beginner will find it as an interesting start of mathematical relation to the multilayer perceptron.

## Prerequisite

- Understanding perceptron.
- Understanding step function.
- Understanding weight,input,output.

## Explanation

The truth table for a two-input XOR-Gate is given below,

| X1 | X2 | output |
|----|----|--------|
| 0  | 0  | 0      |
| 1  | 0  | 1      |
| 0  | 1  | 1      |
| 1  | 1  | 0      |

Fig 1.1 : XOR-Gate Truth Table

We want to get outputs as shown in the above truth table. For this purpose, we have made an MLP (Multilayer Perceptron) architecture shown below.

Here, the circles are neurons(O1, N1, N2, X1, X2), and the orange and blue lines with numbers are the representation of input direction with weights. The numbers on the arrows represent weights. B1 and B2 represent biases.

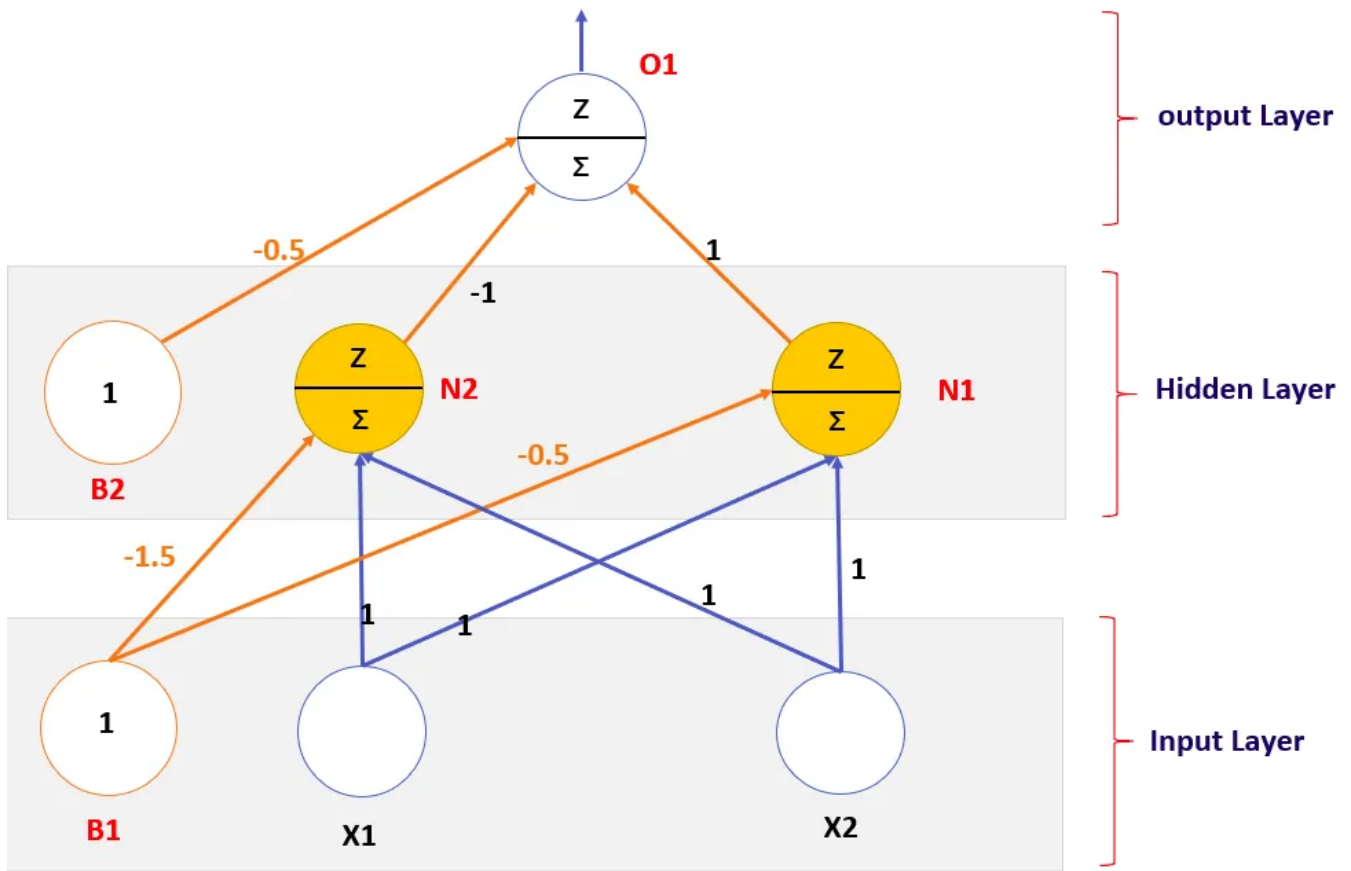


Fig 1.2:XOR-Gate representation using perceptrons.

### Step function:

The step function (z), triggers only if the weighted sum is 1 or greater than 1. That is to say ,

$$\begin{aligned} \text{if } X_1 * W_1 + X_2 * W_2 \geq 1, \text{ then, Step - Fuction}(X_1 * W_1 + X_2 * W_2) &= 1 \\ \text{if } X_1 * W_1 + X_2 * W_2 < 1, \text{ then, Step - Fuction}(X_1 * W_1 + X_2 * W_2) &= 0 \end{aligned}$$

Equation 1.1 :Defining step function

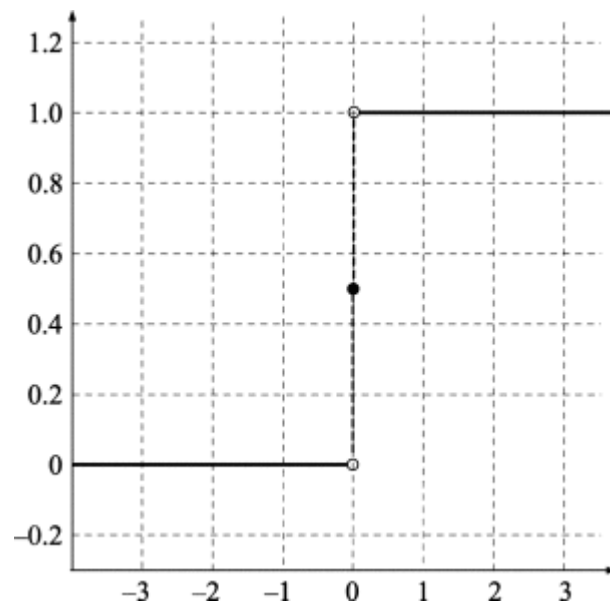


Fig 1.3 :Step function in graph

### Calculation of XOR gate output

Recall that if we get a value of 1 or greater than 1 for the weighted sum, we will get a value of 1 as an output of the step function otherwise we will get a value of 0 in the output.

#### Row 1, Truth table Fig(1.1),

The XOR gate truth table says, if  $X_1 = 0$  and  $X_2 = 0$ , the output should be 0 .

*For hidden layer neuron N1 (Fig1.2),*

$$\begin{aligned} N_1 &= X_2 * 1 + X_1 * 1 + B_1 * (-0.5) \\ &= 0 * 1 + 0 * 1 - 0.5 = -0.5 \end{aligned}$$

So ,  $\text{step\_function}(-0.5) = 0$  ,output of  $N_1 = 0$

*For the hidden layer neuron N2 (fig1.2),*

$$\begin{aligned} N_2 &= X_2 * 1 + X_1 * 1 + B_1 * (-1.5) \\ &= 0 * 1 + 0 * 1 - 1.5 = -1.5 \end{aligned}$$

So ,  $\text{step\_function}(-1.5) = 0$  ,output of  $N_2 = 0$

*For the output neuron O1 (fig1.2),*

$$\begin{aligned}
 O_1 &= N_2 * (-1) + N_1 * 1 + B_2 * (-0.5) \\
 &= (0) * (-1) + (0) * 1 + 1 * (-0.5) \\
 &= 0 - 0 - 0.5 = -0.5
 \end{aligned}$$

So, step\_function(-0.5) = 0 ,output of **O1 = 0**

Matched with the Fig 1.1 ,XOR truth table first row.

**Row 2,Truth table Fig(1.1),**

The XOR gate truth table says, if X1 = 1 and X2 =0 ,the output should be 1 .

*For the hidden layer neuron N1 (fig1.2),*

$$N_1 = X_2 * 1 + X_1 * 1 + B_1 * (-0.5) = 0 * 1 + 1 * 1 - 0.5 = 1 - 0.5 = 0.5$$

So, step\_function(0.5) = 1 ,output of **N1 = 1**

*For the hidden layer neuron N2 (fig1.2),*

$$N_2 = X_2 * 1 + X_1 * 1 + B_1 * (-1.5) = 0 * 1 + 1 * 1 - 1.5 = 1 - 1.5 = -0.5$$

So, step\_function(-0.5) = 0 ,output of **N2 = 0**

*For the output neuron O1 (fig1.2),*

$$\begin{aligned}
 O_1 &= N_2 * (-1) + N_1 * 1 + B_2 * (-0.5) \\
 &= (0) * (-1) + (1) * 1 + 1 * (-0.5) = 0 - 1 - 0.5 = 0.5
 \end{aligned}$$

So, step\_function(0.5) = 1 ,output of **O1 = 1**

Matched with the Fig 1.1 ,XOR truth table second row.

**Row 3,Truth table Fig(1.1),**

The XOR gate truth table says, if X1 = 0 and X2 =1 ,the output should be 1.

*For the hidden layer neuron N1 (fig1.2),*

$$N_1 = X_2 * 1 + X_1 * 1 + B_1 * (-0.5) = 1 * 1 + 0 * 1 - 0.5 = 1 - 0.5 = 0.5$$

So,  $\text{step\_function}(0.5) = 1$  ,output of **N1 = 1**

*For the hidden layer neuron N2 (fig1.2),*

$$N_2 = X_2 * 1 + X_1 * 1 + B_1 * (-1.5) = 1 * 1 + 0 * 1 - 1.5 = 1 - 1.5 = -0.5$$

So,  $\text{step\_function}(-0.5) = 0$  ,output of **N2 = 0**

*For the output neuron O1 (fig1.2),*

$$\begin{aligned} O_1 &= N_2 * (-1) + N_1 * 1 + B_2 * (-0.5) \\ &= (0) * (-1) + (1) * 1 + 1 * (-0.5) = 0 - 1 - 0.5 = -0.5 \end{aligned}$$

So,  $\text{step\_function}(0.5) = 1$  ,output of **O1 = 1**

Matched with the Fig 1.1 ,XOR truth table third row.

**Row 4,Truth table Fig(1.1),**

The XOR gate truth table says, if  $X_1 = 1$  and  $X_2 = 1$ , the output should be 0.

*For the hidden layer neuron N1 (fig1.2),*

$$N_1 = X_2 * 1 + X_1 * 1 + B_1 * (-0.5) = 1 * 1 + 1 * 1 - 0.5 = 1 + 1 - 0.5 = 1.5$$

So,  $\text{step\_function}(1.5) = 1$  ,output of **N1 = 1**

*For the hidden layer neuron N2 (fig1.2),*

$$N_2 = X_2 * 1 + X_1 * 1 + B_1 * (-1.5) = 1 * 1 + 1 * 1 - 1.5 = 1 + 1 - 1.5 = 0.5$$

So,  $\text{step\_function}(0.5) = 1$  ,output of **N2 = 1**

*For the output neuron O1 (fig1.2),*

$$\begin{aligned} O_1 &= N_2 * (-1) + N_1 * 1 + B_2 * (-0.5) \\ &= (1) * (-1) + (1) * 1 + 1 * (-0.5) = -1 + 1 - 0.5 = -0.5 \end{aligned}$$

So,  $\text{step\_function}(-0.5) = 1$ , output of **O1 = 0**

Matched with the Fig 1.1 ,XOR truth table fourth row.

## NOTE

The weights that I have used here are predetermined. In a real-world situation, we have to use a method called backpropagation to train this multilayer perceptron. After training, we will get the weights we have used here. But for this post, I have just shown the possibilities that we have if we want to solve such problems using perceptron. I will write about backpropagation to train a multilayer perceptron in the future.

## Conclusion

For each row of the XOR gate truth table, we found our multilayer perceptron structure (fig 1.2) has given the correct output.

Thanks for reading.

Neural Networks

Perceptron

Begginer

Mathematics

Step Functions



Follow

## Written by Mehedee Hassan

22 Followers · Writer for Analytics Vidhya

Lost in the coding space ..

---

More from Mehedee Hassan and Analytics Vidhya