Artificial Neuron Network Implementation of Boolean Logic Gates by Perceptron and Threshold Element as Neuron Output Function

Ele, Sylvester I.[1], Adesola, W. A.[2]

[1]Department of Computer Science, University of Calabar, Calabar, Cross River State, Nigeria

[2]Department of Computer Science, Cross River University of Technology, Calabar, Cross River State, Nigeria

Abstract: ***Threshold functions and Artificial Neural Networks (ANNs) are known for many years and have been thoroughly analyzed. The primary interest of these paper is to implement the basic logic gates of AND and EXOR by Artificial Neuron Network using Perceptron, and Threshold elements as Neuron output functions. The McCulloch-Pitts neural model was applied as linear threshold gate. The linear threshold gate was used to classify the set of inputs (χ_1, χ_2) into two classes, which yielded a binary output, X. The weighted values $Ш_1$,$Ш_2$, were normalized in the ranges of either (0,1) or (-1,1) and associated with each input line (χ_1, χ_2), sum is the weighted sum, and T is a threshold constant. With the binary inputs of χ_1, $\chi_2 = 0$ or 1, the weights $Ш_1$,$Ш_2 = 1$, and an offset -1.5, weighted summation as propagation; the output of the binary AND function unit was defined as $X = ƒ(-1.5 + \chi_1 + \chi_2)$, with $ƒ(\chi) = 0\ \forall\chi < 0$ and $ƒ(\chi) = 1\ \forall\chi \geq 0$.***

Keywords: Neuron, Perceptron, Threshold, Logic gates

1. Background and Introduction of the Study

Computers are great at solving algorithmic and mathematical problems, but often the world can't easily be defined with a mathematical algorithm. Facial recognition and language processing is a couple of examples of problems that can't easily be quantified into an algorithm; however these tasks are insignificant to humans. The key to Artificial Neural Networks is that their design enables them to process information in a similar way to our own biological brains, by drawing inspiration from how our own nervous system functions [1]. The building blocks of feed-forward networks are computation units (or neurons).

Artificial neural networks refer to computing systems whose central theme is borrowed from the analogy of biological neural networks (Mehrota, et al, 1996). Neural networks are being investigated as a computational paradigm in many fields of artificial intelligence. The pioneers of cybernetics were clearly inspired by neurology and the current knowledge of the human brain to develop the architectures of modern computing machines [3]. The first, very simplified model, mathematical model of a neuron operating in an all or none fashion: the Threshold Logic Gate (TLG). It did not take very long for a hardware implementation to be developed.

Digital logic gates or Boolean logics are electronic devices that make logical decisions based on the different combinations of digital signals present on its inputs. Digital logic gates may have more than one input but generally only have one digital output. Individual logic gates can be connected together to form combinational or sequential circuits or larger logic gate functions. Digital logic gate is the basic building block from which all digital electronic circuits and microprocessor based systems are constructed from (www.electronic-tutorial.ws). The great interest to threshold elements and threshold logics, lasting decades, is caused, by wider functional threshold elements' capabilities in comparison with the traditionally based ones' (that is, AND, NAND, OR, NOR, EXOR etc.), and base on the fact, that threshold elements may be used as a functional basis for artificial neural networks (Varshavsky, et al).

One of the primary functions of the brain is associative memory. We associate the faces with names, letters with sounds, or we can recognize the people even if they have sunglasses or if they are somehow elder now. Associative memories can be implemented either by using feed forward or recurrent neural networks (Ankara).

McCulloch & Pitts (McCulloch, 1943) are generally recognized as being the designers of the first neural network. They recognized that combining many simple processing units together could lead to an overall increase in computational power. Many of their suggested ideas are still in use today. For instance, the idea that a neuron has a threshold level and once that level is reached the neuron fires is still the fundamental way in which artificial neural networks operate.

Threshold functions and Artificial Neural Networks (ANNs) are known for many years and have been thoroughly analyzed. However, most works focused on theoretical capabilities of such elements rather than constructing such elements in hardware (Nikodem, 2010). When the threshold function is used as the neuron output function, and binary input values 0 and 1 are assumed, the basic Boolean functions AND, OR, EXOR or NOT of two variables can be implemented by choosing appropriate weights and threshold values. The first two neurons receives two binary inputs U_1, U_2 and produces $y(U_1, U_2)$ for the Boolean functions AND and OR respectively. The last neuron implements the NOT function (Hertz et al, 91).

Computational model of the biological neuron was first introduced by McCulloch and Pitts (McCulloch and Pitts, 1943) in the 1940s. McCulloch and Pitts combined mathematical logic and neurophysiology to put forward a binary threshold unit as a computational model for an artificial neuron operating in discrete time.

Rosenblatt, an American psychologist proposed a computational model of neurons that he called The Perceptron in 1958 (Rosemblatt, 1958). The essential innovation was the introduction of numerical interconnection weights. A neuron is an information processing unit that is fundamental to the operation of a neural network (Haykin, 1998). The perceptron model of the neuron has three basic elements.

1) Synapses that are characterized by the strength of the weights
2) An adder for summing the input signals, weighted by the respective synapses and the threshold of the neuron;
3) An activation function for limiting the amplitude of the output (in this case completely unclocked with the firing event).

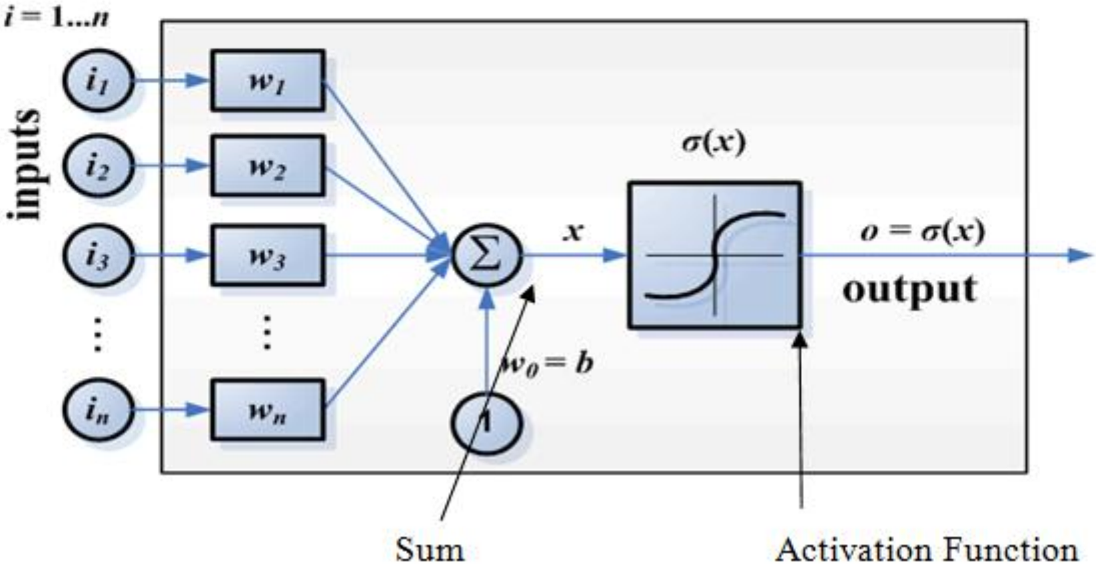

Figure 1: Structure of a typical Perceptron

As shown in the diagram above a typical perceptron will have many inputs and these inputs are all individually weighted. The perceptron weights can either amplify or deamplify the original input signal. For example, if the input is 1 and the input's weight is 0.2 the input will be decreased to 0.2 (Jacobson, 2013). A step function will typically output a 1 if the input is higher than a certain threshold, otherwise its output will be 0.

2. Biological Neuron Concept

The human central nervous system is comprised of about $1.3x10^{10}$ neurons and that about $1x10^{10}$ of them takes place in the brain. At any time, some of these neurons are firing and the power dissipation due this electrical activity is estimated to be in the order of 10 watts. According to him, monitoring the activity in the brain has shown that, even when asleep, $5x10^{7}$ nerve impulses per second are being relayed back and forth between the brain and other parts of the body. A neuron operates by receiving signals from other neurons through connections, called synapses. The combination of these signals, in excess of a certain threshold or activation level, will result in the neuron firing, which is sending a signal on to other neurons connected to it. If a neuron fires, an electrical impulse is generated. This impulse starts at the base, called the hillock, of a long cellular extension, called the axon, and proceeds down the axon to its ends (Cheshire Engineering Corporation, 2003).

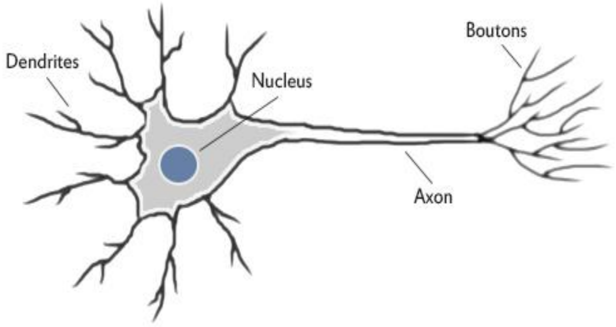

Figure 2: Structure of a Typical Neuron
(Source: Jacobson www.theprojectpot.com)

An axon, from the figure above, having a length varying from a fraction of a millimeter to a meter in human body, prolongs from the cell body at the point called **axon hillock**. At the other end, the axon is separated into several branches, at the very end of which the axon enlarges and forms terminal **buttons**. Terminal buttons are placed in special structures called the synapses which are the junctions transmitting signals from one neuron to another. A neuron naturally drives 10^{3} to 10^{4} synaptic junctions. The **synaptic vesicles** holding several thousands of molecules of chemical transmitters, take place in terminal buttons.

At time t=0 a neuron is excited; at time T, normally it may be of the order of 50 milliseconds, the neuron fires a train of impulses along its axon. Sometime later, say around t=T+τ, the neuron may fire another train of impulses, as a result of the same excitation, though the second train of impulses will usually contain a smaller number (Helici,). The schematic representation of biophysical neuron shown below explains the concept.

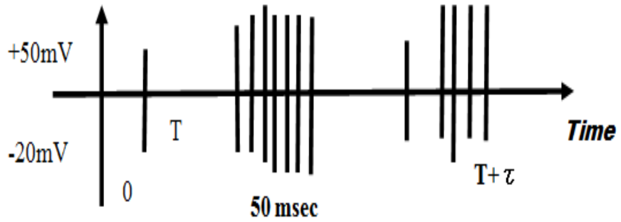

Figure 3: Representation of biophysical neuron output signal after excitation at tine t=0

3. Artificial Neuron Model

The transmission of a signal from one neuron to another through synapses is a complex chemical process in which specific transmitter substances are released from the sending side of the junction. Artificial neuron models are at their core simplified models based on biological neurons. This allows them to capture the essence of how a biological neuron functions. We usually refer to these artificial neurons as 'perceptrons' (Jacobson,). Artificial neurons have the same basic components as biological neurons. The simplest ANNs consist of a set of McCulloch-Pitts neurons labelled by indices k, i, j and activation flows between them via synapses with strengths w_{ki}, w_{ij} (Bullinaria, 2004) as shown in the figure below.

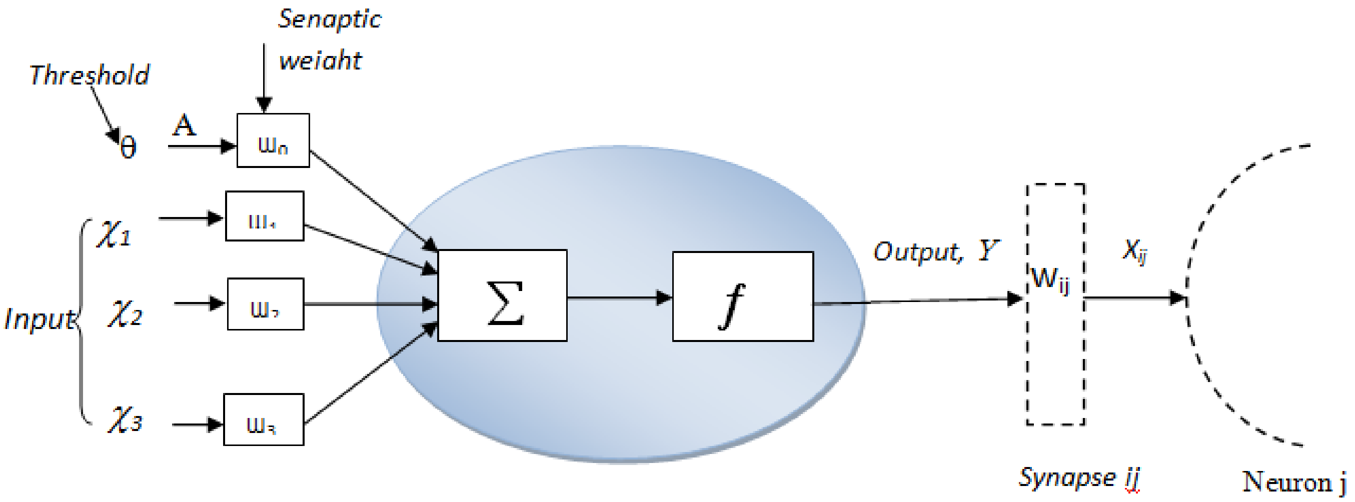

Figure 4: A Simple Artificial Neural Network

4. Implementation of Logic Gates Using Artificial Neuron

When the threshold function is used as the neuron output function, and binary input values 0 and 1 are assumed, the basic Boolean functions AND and EXOR of two or three variables can be implemented by choosing appropriate weights and threshold values. The primary interest of these paper is to implement the basic logic gates of AND and EXOR using Boolean functions (Krose and Smagt, 1996). McCulloch & Pitts, as we earlier noted, are generally recognized as being the designers of the first neural network. Originally the neuron output function f(χ) in McCulloch Pitts model proposed as threshold function, however linear, ramp and sigmoid and functions are also widely used output functions:

For Linear: $f(\chi) = \mathbf{k}\chi$ For Threshold: $f(\chi) = \begin{cases} 0 \ if \ \chi < 0 \\ 1 \ if \ k = \chi \end{cases}$

$$\text{Ramp: } f(\chi) = \begin{cases} 0 & \chi \le 0 \\ \chi / k & 0 < k \le \chi \\ 1 & k < \chi \end{cases}$$

$$\text{sigmoid: } f(\chi) = \frac{1}{1 + \chi^{-k\chi}}$$

For the implementation of the logic gates, we have inputs χ_1 and output out = sgn($\mathbf{w_1}\,\chi_1 + \mathbf{w_2}\,\chi_2 - \theta$) and need to solve for $\mathbf{w_1}$ and θ **(** Bishop, 2007**)**.

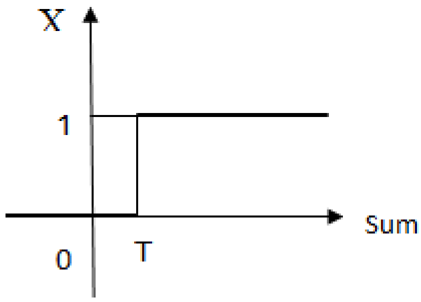

Figure 5: Linear Representation of Threshold Functions

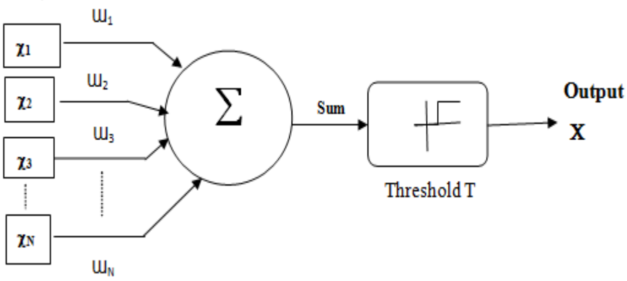

Figure 6: Symbolic Illustration of Linear Threshold Gate

1. AND

Boolean Function: $\chi_1 \wedge \chi_2 = X$

Logic Table

χ_1	χ_2	Output(X)
0	0	0
0	1	0
1	0	0
1	1	1

From the table, we can now solve for Ш_1 andθ:

$\text{Ш}_1\ 0 + \text{Ш}_2\ 0 - \theta < 0$

$\text{Ш}_1\ 0 + \text{Ш}_2\ 1 - \theta < 0$

$\text{Ш}_1\ 1 + \text{Ш}_2\ 0 - \theta < 0$

$\text{Ш}_1\ 1 + \text{Ш}_2\ 1 - \theta > 0$

$\theta > 0\ ,\ \text{Ш}_1, \text{Ш}_2 < \theta\ ,\ \text{Ш}_1 + \text{Ш}_2 > \theta$

Logic Gate:

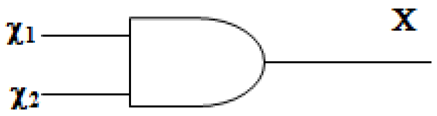

Figure 7: Logic Gate of the Binary AND Function

Artificial Neuron:

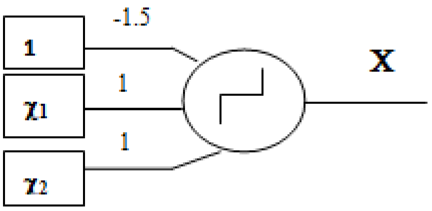

Figure 8: Threshold Function implementation of the Binary AND Function

Perceptron implementation of Binary AND function:

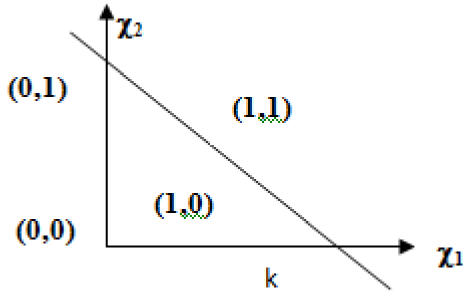

Figure 9: Perceptron implementation of Binary AND function

Given the binary inputs $\chi_{1,}\ \chi_2 = 0$ or 1, the weights $\text{Ш}_1,\text{Ш}_2 = 1$, an offset -1.5, weighted summation as propagation; the output of the unit is defined as:

$X = f\ (-1.5 + \chi_1 + \chi_2)$, with $f\ (\chi) = 0\ \forall\chi < 0$ and $f\ (\chi) = 1\ \forall\chi \geq 0$.

2. EXOR

Boolean Function: $\chi_1 \wedge \chi_2 = X$

Logic Table

χ_1	χ_2	Output(X)
0	0	0
0	1	1
1	0	1
1	1	0

From the table, we can now solve for w_1 andθ:

$\text{Ш}_1\ 0 + \text{Ш}_2\ 0 - \theta < 0$

$\text{Ш}_1\ 0 + \text{Ш}_2\ 1 - \theta < 0$

$\text{Ш}_1 1 + \text{Ш}_2\ 0 - \theta < 0$

$\text{Ш}_1\ 1 + \text{Ш}_2\ 1 - \theta > 0$

$\theta < 0, \text{Ш}_1, \text{Ш}_2 > \theta, \text{Ш}_1 + \text{Ш}_2 > \theta$

Logic Gate:

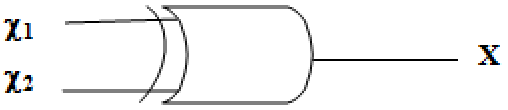

Figure 10: Logic Gate of the Binary EXOR Function

Artificial Neuron:

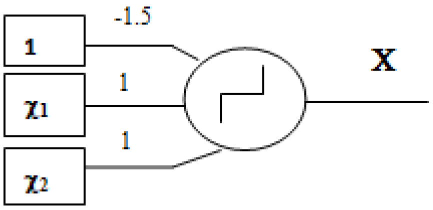

Figure 11: Threshold Function implementation of the Binary EXOR Function.

Perceptron implementation of logical EXOR function:

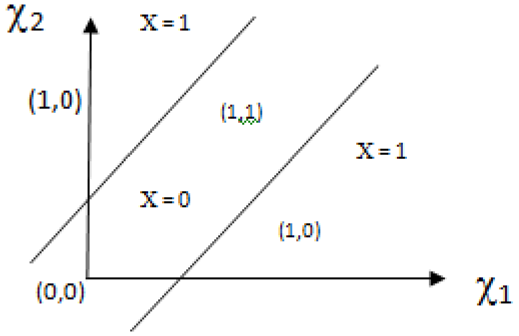

Figure 12: Perceptron implementation of Binary EXOR function

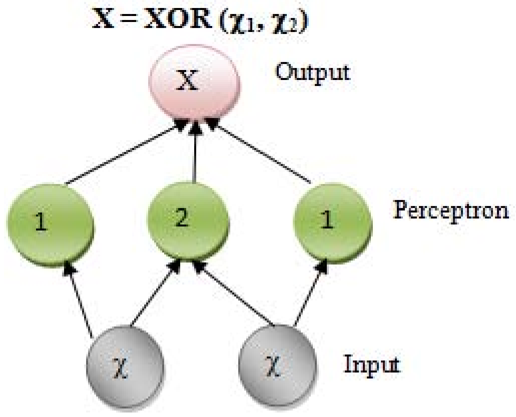

Figure 13: A two-layer Neural Network capable of calculating Binary EXOR function

5. Conclusion

This article implemented the binary AND and EXOR gates using Artificial Neural Network. This implementation was achieved by using perceptron and threshold elements as neuron output functions. The McCulloch-Pitts neural model was applied as linear threshold gate. The linear threshold gate was used to classify the set of the inputs, **(χ_1, χ_2)** into two classes, which yielded a binary output, **X**. In each binary functions, AND and EXOR, the threshold function, the perceptron, and the layered neural Network capable of calculating the binary function were realized.

References

[1] Jacobson, L. (2013). Introduction to Artificial Neural Network. www.theprojectpot.com/tutorials

[2] Mehrota K. et al(1996). Elements of Artificial Neural Networks. MIT Press. https:\\www.mitpress.mit.edu/books/elements-artificial-neural-networks.

[3] Alan P. O. et al (). Adaptive Boolean Logic Using Ferroelectrics Capacitors as Basic Units of Artificial Neurons. www.electronic-tutorials.ws.

[4] Nikodem N, (2010), Hardware Implementation of Artificial Neural Networks for Boolean Functions with Generalized Threshold Gate ircuits. Lecture notes in Computer Science volume 6438, 2010.

[5] Varshavsky, V. Marakhovsky, V. & Levin, I. Artificial Neurons Based on CMOS β-DrivenThreshold Elements with Functional Inputs.

[6] McCulloch, W. S.; Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophysiol., vol. 5, pp. 115-133.

[7] Rosemblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psych. Revue, vol. 65, pp. 386-408.

[8] Haykin, S. (1998). Neural Networks: A Comprehensive Foundation. Prentice Hall, 2nd Edition

[9] Ugur H.(2009). Artificial Neural Networks. EE543 Lecture Notes Metu EEE . ANKARA.

[10] Basic Concepts for Neural Networks (2003). Cheshire Engineering Corporation, October 2003. www.chechireeng.com

[11] Introduction to Neural Networks : Revision Lectures © John A. Bullinaria, 2004

[12] Krose, B. and van der Smagt, P.(1996) "An Introduction to Neural Networks," The University of Amsterdam, November 1996.

[13] Bishop, C.M.(2007). Pattern recognition and machine learning. Springer, October 2007.