A Comprehensive Review on McCulloch-Pitts Neuron Model

J. Vijaychandra, B.Sesha Sai, B.Sateesh Babu, P.Jagannadh

***Abstract*: This Paper presents the concepts of basic models of Biological Neuron and the Artificial Neurons. The main aim of writing this paper is to provide the concepts of basic model of Artificial Neural Networks i.e., McCulloch-Pitts Neuron model and implementation of logical gates viz., AND, OR, NOT, AND NOT and XOR with detailed calculations. In addition, this Paper also addresses on the Historical developments of the Artificial Neural Networks.**

***Index Terms*: Artificial Neural Networks, Biological Neuron McCulloch-Pitts, Neuron model.**

I. INTRODUCTION

. This Paper will provide the basic knowledge of Artificial Neural Networks. The main aim of this Paper is to make the students understand the analysis of Artificial Neurons and its basic model. The Organisation of the Paper is as Follows. It begins with the concept of Biological Neuron and followed by the study of Artificial Neuron Model, Historical developments and the basic ANN model i.e., Mcculloch-Pitts Neuron Model and ends with the mathematical analysis of different logic gates.

II. LITERATURE SURVEY

The detailed analysis and concepts of of Mcculloch Pitts Neuron Model was explained clearly in the text book of Introduction to Artificial Neural Systems-Jacek M.Zurada, Jaico Publishing House,1997[1].

In Fundamentals of Neural Networks Architectures, Algorithms and Applications-by LaurenceFausett, Pearson [2], the importance of Artificial neural network with different characteristics has been proposed.

In the book, Neural Networks, Algorithms, Applications and programming Techniques by James A.Freeman, David M.Skapura [3], different models of artificial neural networks were explained and one among them is the basic model of ANNs i.e., Mcculloch Pitts Neuron model.

In the book, Introduction to Neural Networks using MATLAB 6.0 by S N Sivanandam, S Sumathi, S N Deepa TMGH[4], the historical developments of ANNs were clearly explained and gives us a proper understandingchara of the development of ANNs.

In the book, Neural Networks- Simon Hakins, Pearson Education [5], the characteristics of the ANNs have been ckearly explained.

In the books which were shown in the references [6],[7],[8],[9], the concepts of Mcculloch pitts neuron model was explained with the implementation of different logic gates. These References have also described the historical developments and characteristics of ANNS in detail.

Herve Debar, Monique Becker and Didier Siboniin "A Neural Network Component for an Intrusion Detection System", Les UhsCedex France, 1992.[10] has been discussed some of the applications of ANNS in Detection systems.

Ajith Abraham, in his paper "Artificial Neural Networks" Stillwater OK, USA, 2005. " [11] has given more details about the origin of ANNs and the applications in various disciplines.

Carlos Gershenson, in "Artificial Neural Networks for Beginners", United Kingdom. [12] has defined the concepts of Biological Neuron model and the ANNs with different characteristics.

In the paper by Anil K Jain, Jianchang Mao and K.M Mohiuddin, "Artificial Neural Networks: A Tutorial", Michigan State University, 1996 [13], they described the concepts of implementation of different logical gates using basic model of ANN i.e., Mcculloch Pitts Neuron Model.

In UgurHALlCI, "Artificial Neural Networks", Chapter I, ANKARA [14], a complete analysis of ANNs was presented .

In Eldon Y. Li, "Artificial Neural Networks and their Business Applications", Taiwan, 1994 [15], there is a discussion on the applications of ANNs in the Business.

The references [16] Christos Stergiou and DimitriosSiganos, "Neural Networks". Describes the models of ANNs in detail.

The Limitations and Disadvantages of Artificial Neural Network were explained in detail in the reference [17].

In Image of a Neuron form website http://transductions.neti 20 I 0102/04/3 I 3/neuronsl [18], the image of Biological and Artificial Neurons has been presented.

Revised Manuscript Received on April 07, 2019.

J. Vijaychandra, Department, of EEE, Sri SIvani College of Engineering, Srikakulam, INDIA.

B.Sesha sai, Department, of EEE, Sri SIvani College of Engineering, Srikakulam, INDIA.

B. Sateesh babu, Department, of EEE, Sri SIvani College of Engineering, Srikakulam, INDIA.

P. Jagannadh, Department, of EEE, Sri SIvani College of Engineering, Srikakulam, INDIA.

References [19] to [30] describes all the concepts of ANNs in detail and the applications of ANNs in different disciplines have been provided. In addition, these were explained different algorithms of ANNs, Characteristics, advantages and disadvantages of ANNs.

III. BIOLOGICAL NEURON

In general, our human brain consists of mainly three parts i.e., cerebrum, Cerebellum and Medulla & Oblongata. The Human brain is a collection of about Ten Billion interconnected neurons. Here each neuron can be considered as a cell that uses Biochemical reactions to receive, process and transmit the information.[1][2][3]

A Biological Neuron mainly consists of Five parts. They are Dendrites, Soma, Axon Hillock, Axon and the Synapse. Dendrites forms a tree like structure and makes the Input nodes to reach Soma. Soma, also called Cell body contains cell's nucleus and other vital components.

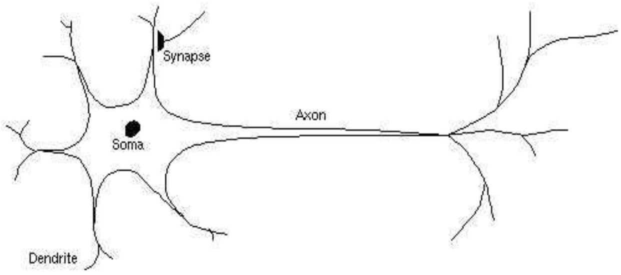


Fig1: Biological Neuron Model

The Axon is a tubular like structure at which it collects the output information which was processed before in the Soma. Axon Hillock is the connection link between the Soma and the Axon. The output information from the axon can be connected to the input nodes of the another neuron with the help of Synapse. Hence Synapse, also called Synaptic weights are called Neuro Transmitters.

IV. ARTIFICIAL NEURAL NETWORK

Use either SI (MKS) or CGS as primary units. (SI units are strongly encouraged.) English units may be used as secondary units (in parentheses). ANNs are said to be Massively, Parallel, Adaptive network consisting of some simple non-linear computing elements called Neurons which are intended to perform some computational tasks similar to a Biological neuron.[19]to[30]

It can also be defined as, "An Information processing paradigm which is highly inspired by the way of human biological neuronal systems".[11][12][13][14][15]

The Artificial Neural networks resembles the brain of a human in Two different aspects i.e., ANNs adopt a learning mechanism which can be used to acquire the knowledge and storing the knowledge can be done with the help pf Synaptic weights. In general, An Artificial Neural Network is characterised by its Architecture, Training and Activation Function[3][5][6][10][15][16].

The basic model of an Artificial Neural Network has been shown below.

It consists of Input nodes denoted by x_1, x_2…. x_n, Weights are represented by w_1, w_2…w_{1n} and weights can be considered as the information used by the neural net to solve a computational task. The Aggregation block or the Summation block in the model represents the information processing block where the net input can be calculated and the Activation function is used to calculate the output response of a neuron. The sum of the weighted input signal is applied with the activation function to obtain the response.Finally, the output will be obtained at the output terminal or node[4].

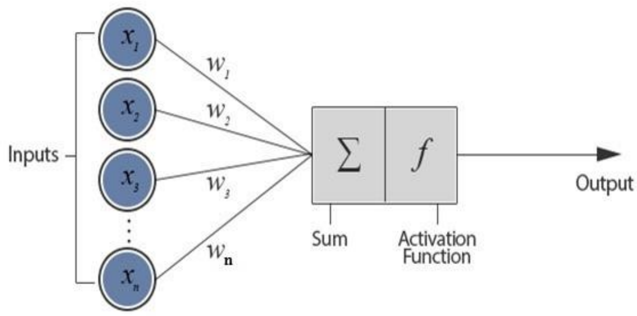


Fig2: Artificial Neural Network Model

V. HISTORICAL DEVELOPMENTS

[4][6][7][8][9][17][18]

1.) 1943-MCCULLOCH-PITTS MODEL:

Warren Mcculloch and Walter Pitts have proposed a model in the year 1943 and this model has been considered as the Beginning of the Modern era of ANNs. This model forms a logical calculus of ANN. This neuron is a binary activated neuron in which it allows only the binary states 0 and 1. Here the neuron is associated with a Threshold value and the activation of the neuron depends on the Threshold value. The neuron will activate only when the net input to the neuron is greater than the Threshold value[1],[3].

2.) 1949-Hebb's Book: The Organisation of Behaviour":

The concept of Synaptic Modification was introduced for the first time in the year 1949. According to the Hebb, if two neurons are found to be active simultaneously, then the strength of the connection between the two neurons should be increased.

3.) 1958- Perceptron Network:

In this year, Rossenblatt introduced a network called Perceptron network and in this network, the weights on the connection paths can be adjusted.

4.) 1960-ADALINE Network:

In the year 1960, Widraw and Hoff introduced ADALINE network. It uses a learning rule called "Mean square rule" or "Delta rule".

5.) 1960-Hopfield's Network:

John Hopfield developed a network in which the nets are widely used as the nets of Associative Memories. This net provides a good solution for the Traveling salesman problem.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

6.) 1972-Kohonen's SOM:

The nets developed here make the use of Data interpretation using Topographic maps which are very commonly seen in the Nervous system.

7.) 1985-Parker,1986-Lecum:

The Back propagation network was introduced in this year in which this method propagates the information of error at the output units back to the Hidden units with the help of a Generalized Delta Rule.

8.) 1988-Grossberg:

He developed a learning rule similar to that of Kohonen and this net is used in the counter propagation net. Here the learning used is Outstar Learning and this occurs for all the units in a particular layer.

9.) 1987,1990-Adaptive Resonance Theory(ART):

It was formulated by Carpenter and Grossberg. The proposed theory was designed for both Binary inputs and the Continuous Valued Inputs. Here in these type of nets, the Input patterns can be presented in any order.

10.) 1990-Vanpik:

He developed Support Vector Machine(SVM)

VI. CHARACTERISTICS OF ANN

The ANNs exhibits following characteristics like Processing of information in Parallel, Non linearity, Input-output Mapping, Ability of learning, Fault tolerant, Adaptability and Response.[4][6][12][16]

Mc Culloch – Pitts Neuron Model: -

Mc Culloch – Pitts Neuron Model was formulated in the year by Warren Mc Culloch and Walter Pitts in the year 1943. It is characterized by its formalism, elegant and precise mathematical definition. The neuron allows only the binary states i.e., '0's and '1's. so it is called as a binary activated neuron. These neurons are connected by direct weighted path. The connected path can be excitatory or inhibitory.[4][5][6]

Excitatory connections have positive weights and inhibitory connections have negative weights.There will be same weight for the excitatory connections entering into a particular neuron. The neuron is associated with the Threshold value.[19][20][21] The neuron fires if the net input to the neuron is greater than the Threshold. The threshold is set, so that the inhibition is absolute, because the non-zero inhibitory output will prevent the neuron from firing[5-7].

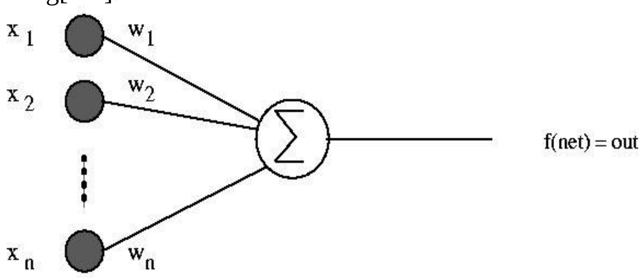


FIG3: Mc Culloch – Pitts Neuron Model

Y = Mc Culloch – Pitts Neuron which can receive signal from any other neurons.

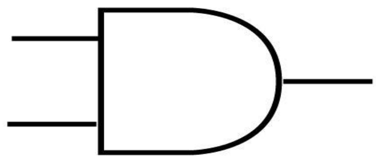
W = Weights of the neuron. Weights are Excitatory when positive and Inhibitory when negative.

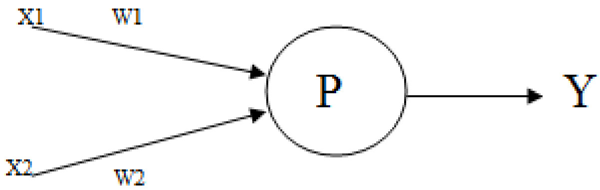
The Mc Culloch – Pitts Neuronhas an activation function

$f(Y) = 1$, if $Y_{in} >= \Theta$

$= 0$, if $Y_{in} < \Theta$.

AND GATE IMPLEMENTATION:





Truth table:

INPUT X1	INPUT X2	OUTPUT Y= X1 * X2
0	0	0
0	1	0
1	0	0
1	1	1

Net input P = ∑ (Inputs * Weights) = [(X1*W1) +(X2*W2)]

Assuming the weights are excitatory, W1 = 1 and W2 = 1

∴ P= (X1*1) +(X2*1) = X1+X2

Here, the Threshold $\Theta = 2$

INPUT X1	INPUT X2	NET INPUT P	OUTPUT Y= X1 * X2
0	0	0	0
0	1	1	0
1	0	1	0
1	1	2	1

$Y(P) = 1$, if $P >= \Theta$

$= 0$, if $P < \Theta$.

Case 1: X1 = 0, X2 = 0, then P = (0*1)+(0*1) = 0 <2, => Y= 0

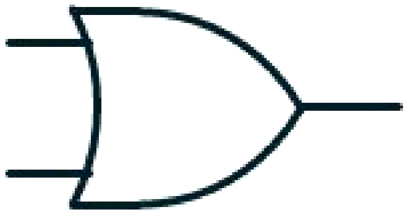
Case 2: X1 = 0, X2 = 1, then P = (0*1)+(1*1) = 1 <2, => Y= 0

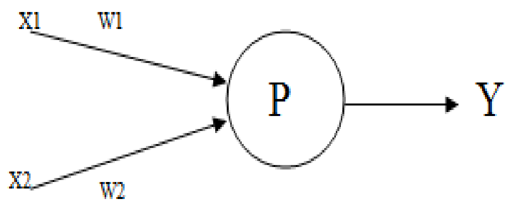
Case 3: X1 = 1, X2 = 0, then P = (1*1)+(0*1) = 1 <2, => Y= 0

Case 4: X1 = 1, X2 = 1, then P = (1*1)+(1*1) = 2 >=2, => Y= 1

OR GATE IMPLEMENTATION:

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication





Truth table:

INPUT X1	INPUT X2	OUTPUT Y= X1 +X2
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**1**

Net input P = ∑ (Inputs * Weights) = [(X1*W1) +(X2*W2)]
Assuming the weights are excitatory, W1 = 1 and W2 = 1
∴ P= (X1*1) +(X2*1) = X1+X2
Here the Threshold ϴ = 1

INPUT X1	INPUT X2	NET INPUT P	OUTPUT Y= X1 +X2
0	**0**	**0**	**0**
0	**1**	**1**	**1**
1	**0**	**1**	**1**
1	**1**	**2**	**1**

Y(P) = 1, if P >= ϴ
= 0, if P <ϴ.

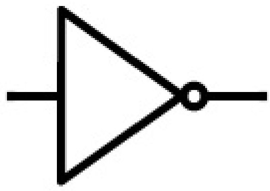
Case 1:X1 = 0, X2 = 0, then P = (0*1)+(0*1) = 0 <1, => Y= 0

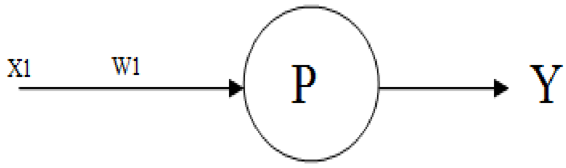
Case 2:X1 = 0, X2 = 1, then P = (0*1)+(1*1) = 1>=1, => Y= 1

Case 3:X1 = 1, X2 = 0, then P = (1*1)+(0*1) = 1 >=1, => Y= 1

Case 4:X1 = 1, X2 = 1, then P = (1*1)+(1*1) = 2 >=1, => Y= 1

NOT GATE IMPLEMENTATION:





Truth table:

INPUT X1	OUTPUT Y=~X1
0	**1**
1	**0**

Net input P = ∑ (Inputs * Weights) = [(X1*W1)]
Assuming the weights are excitatory, W1 = 1
∴ P= (X1*1) =X1
Here, the Threshold ϴ = 0

INPUT X1	NET INPUT P	OUTPUT Y=~X1
0	**0**	**1**
1	**1**	**0**

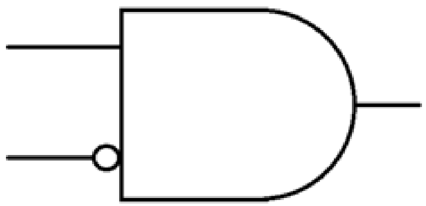
Y(P) = 1, if P<=ϴ
= 0, if P >ϴ.

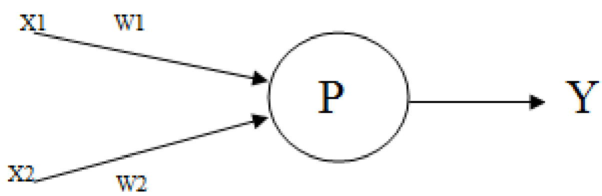
Case 1:X1 = 0, W1 =1, then P = (0*1) = 0 <=1, => Y= 1

Case 2:X1 = 1, W1 =1, then P = (1*1) = 1 > 0, => Y= 0
Note: If the inhibitory weights are used, then the threshold "ϴ" should satisfy the relation

$\theta >= (n\omega - p)$

This is the condition for absolute inhibition.
AND NOT GATE IMPLEMENTATION:





In an AND NOT Gate the output is high only when the first input is HIGH and the second input is LOW.

Truth table:

INPUT X1	INPUT X2	OUTPUT Y= X1
0	**0**	**0**
0	**1**	**0**
1	**0**	**1**
1	**1**	**0**

Assuming the weights are excitatory, W1 = 1 and W2 = 1. We should always consider the case shown in the highlighted portion because the output neuron lies in this case only.
Net input P = ∑ (Inputs * Weights) = [(X1*W1) +(X2*W2)]
Assuming the weights are excitatory, W1 = 1 and W2 = 1
∴ P= (X1*1) +(X2*1) = X1+X2
Here, the Threshold ϴ = 1

Case 1:X1 = 0, X2 = 0, then P = (0*1)+(0*1) = 0 <1, => Y= 0

Case 2:X1 = 0, X2 = 1, then P = (0*1)+(1*1) = 1 >=1, => Y= 1

Case 3:X1 = 1, X2 = 0, then P = (1*1)+(0*1) = 1 >=1, => Y= 1

Case 4:X1 = 1, X2 = 1, then P = (1*1)+(1*1) = 2 >=1, => Y= 1
Y(P) = 1, if P >=ϴ
= 0, if P < ϴ.
We should satisfy the condition shown in the highlighted portion only because the output, Y is high (or) should be high only when the inputs X1 = 1, X2 = 0.
The output for the Inputs X1=0, X2 = 1 and X1 = 1, X2 = 1 are also >= ϴ but we should not consider these conditions, because, we always should satisfy the condition for the Input X1= 1, X2 = 0.For achieving the required output, we must change the weights.
Now assuming the weights, W1 = 1 and W2 = -1.

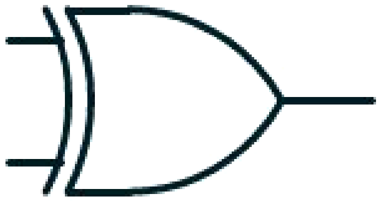
Case 1:X1 = 0, X2 = 0, then P = (0*1)+(0*-1) = 0 <ϴ, => Y= 0

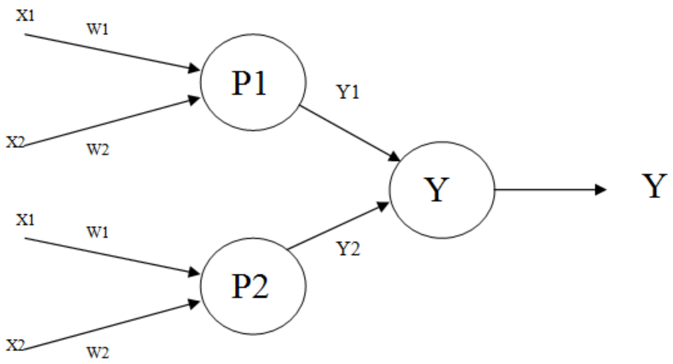
Case 2:X1 = 0, X2 = 1, then P = (0*1)+(1*-1) = -1 <ϴ, => Y= 0

Case 3:X1 = 1, X2 = 0, then P = (1*-1)+(0*-1) = 1 >=ϴ, => Y= 1

Case 4:X1 = 1, X2 = 1, then P = (1*1)+(1*-1) = 0 <ϴ, => Y= 0
{Note: Threshold value "ϴ" can be calculated by,
$\theta >= (n\omega - p)$
where n = number of inputs
w = excitatory input weights
p = inhibitory input weights
$\theta >= (2*1-1)$
$=> \theta >= (1)$ **}**

XOR GATE IMPLEMENTATION:





In an XOR gate, the output is high, when both the inputs are different and it is low, when the inputs are equal.
Truth table:

INPUT X1	INPUT X2	OUTPUT Y= X1' X2 +X1 X2'
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**

The formulation of "XOR" gate is done by considering the "OR" operation of **X1' *X2** and **X1 *X2'**
So Y = Y1+Y2 = (X1' * X2)+(X1 *X2')
<u>For Y1 = X1' * X2</u>

1. Assuming the weights are excitatory, W1 = 1 and W2 = 1

Net input P1 = ∑ (Inputs * Weights) = [(X1*W1) +(X2*W2)]
∴ P1= (X1*1) +(X2*1) = X1+X2
Here, the Threshold ϴ = 1

Case 1:X1 = 0, X2 = 0, then P1 = (0*1)+(0*1) = 0 <ϴ, => Y1= 0

Case 2:X1 = 0, X2 = 1, then P1 = (0*1)+(1*1) = 1 >=ϴ, => Y1= 1

Case 3:X1 = 1, X2 = 0, then P1 = (1*1)+(0*1) = 1 >=ϴ, => Y1= 1

Case 4:X1 = 1, X2 = 1, then P1 = (1*1)+(1*1) = 2>=ϴ, => Y1= 1
Truth table for Y1:

INPUT X1	INPUT X1'	INPUT X2	OUTPUT Y1= X1'* X2
0	**1**	**0**	**0**
0	**1**	**1**	**1**
1	**0**	**0**	**0**
1	**0**	**1**	**0**

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The obtained outputs are not matching the desired output conditions of the truth table.

2. Now assuming the weights, W1 = 1 and W2 = -1.

Net input P1 = ∑ (Inputs * Weights) = [(X1*W1) +(X2*W2)]

∴ P1= (X1*1) +(X2*-1) = X1-X2

Here, the Threshold Θ = 1.

Case 1:X1 = 0, X2 = 0, then P1 = (0*1)+(0*-1) = 0 <Θ, => Y1= 0

Case 2:X1 = 0, X2 = 1, then P1 = (0*1)+(1*-1) = -1 <Θ, => Y1= 0

Case 3:X1 = 1, X2 = 0, then P1 = (1*1)+(0*-1) = 1 >=Θ, => Y1= 1

Case 4:X1 = 1, X2 = 1, then P1 = (1*1)+(1*-1) = 0 <Θ, => Y1= 0

The obtained outputs are again not matching the desired output conditions of the truth table.

3. Now assuming the weights, W1 = -1 and W2 = 1.

Net input P1 = ∑ (Inputs * Weights) = [(X1*W1) +(X2*W2)]

∴ P1= (X1*-1) +(X2*1) = -X1+X2

Case 1:X1 = 0, X2 = 0, then P1 = (0*-1)+(0*1) = 0 <Θ, => Y1= 0

Case 2:X1 = 0, X2 = 1, then P1 = (0*-1)+(1*1) = 1 <Θ, => Y1= 1

Case 3:X1 = 1, X2 = 0, then P1 = (1*-1)+(0*1) = -1 >=Θ, => Y1= 0

Case 4:X1 = 1, X2 = 1, then P1 = (1*-1)+(1*1) = 0 <Θ, => Y1= 0

Now, the obtained outputs are matching the desired output conditions of the truth table.

<u>For Y2 = X1 * X2'</u>

1. Assuming the weights are excitatory, W1 = 1 and W2 = 1

Net input P2 = ∑ (Inputs * Weights) = [(X1*W1) +(X2*W2)]

∴ P2= (X1*1) +(X2*1) = X1+X2

Here, the Threshold Θ = 1.

Case 1:X1 = 0, X2 = 0, then P2 = (0*1)+(0*1) = 0 <Θ, => Y2= 0

Case 2:X1 = 0, X2 = 1, then P2 = (0*1)+(1*1) = 1 >=Θ, => Y2= 1

Case 3:X1 = 1, X2 = 0, then P2 = (1*1)+(0*1) = 1 >=Θ, => Y2= 1

Case 4:X1 = 1, X2 = 1, then P2 = (1*1)+(1*1) = 2 >=Θ, => Y2= 1

Truth table for Y2:

INPUT X1	INPUT X2	INPUT X2'	OUTPUT Y1= X1'* X2
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

The obtained outputs are not matching the desired output conditions of the truth table.

2. Now assuming the weights, W1 = 1 and W2 = -1.

Net input P2 = ∑ (Inputs * Weights) = [(X1*W1) +(X2*W2)]

∴ P2= (X1*1) +(X2*-1) = X1-X2

Here, the Threshold Θ = 1.

Case 1:X1 = 0, X2 = 0, then P1 = (0*1)+(0*-1) = 0 <Θ, => Y2= 0

Case 2:X1 = 0, X2 = 1, then P1 = (0*1)+(1*-1) = -1 <Θ, => Y2= 0

Case 3:X1 = 1, X2 = 0, then P1 = (1*1)+(0*-1) = 1 >=Θ, => Y2= 1

Case 4:X1 = 1, X2 = 1, then P1 = (1*1)+(1*-1) = 0 <Θ, => Y2= 0

The obtained outputs are now matching the desired output conditions of the truth table.

<u>For net Output Y = Y1+Y2 = (X1' * X2)+(X1 *X2')</u>

Assuming the weights are excitatory, W1 = 1 and W2 = 1

Net OutputY = ∑ (Outputs * Weights) = [(Y1*W1) +(Y2*W2)]

∴ Y= (Y1*1) +(Y2*1) = Y1+Y2

Here, the Threshold Θ = 1.

Case 1:Y1 = 0, Y2 = 0, then Y = 0+0 = 0 =>Y = 0

Case 2:Y1 = 0, Y2 = 1, then Y = 1+0 = 1 =>Y = 1

Case 3:Y1 = 1, Y2 = 0, then Y = 0+1 = 1 =>Y = 1

Case 4:Y1 = 1, Y2 = 1, then Y = 0+0 = 0 =>Y = 0

VII. CONCLUSION

In this Paper, we have provided the comparison between the Biological Neuron and an Artificial Neuron. We have discussed the historical developments, characteristics of artificial neural network and the basic model of Artificial neuron model i.e., Mcculloch-Pitts Neuron model and analysed the implementation of various logical gates viz., AND, OR, NOT, ANDNOT and XOR.

REFERENCES

1. Introduction to Artificial Neural Systems-Jacek M.Zurada, Jaico Publishing House,1997.
2. Fundementals of Neural Networks Architectures, Algorithms and Applications-by LaurenceFausett, Pearson.
3. Neural Networks, Algorithms, Applications and programming Techniques by James A.Freeman, David M.Skapura.
4. Introduction to Neural Networks using MATLAB 6.0 by S N Sivanandam, S Sumathi, S N Deepa TMGH
5. Neural Networks- Simon Hakins, Pearson Education.
6. Neural Engineering by C.Eliasmith and CH.Anderson, PHI.
7. Neural Networks and Fuzzy Logic System by Bart Kosko, PHI Publications.
8. I. Aleksander, H. Morto, An Introduction to Neural Computing, London: Chapman and Hall, 1990.
9. M.Caudill, C. Butler, Understanding Neural Netwoks, MA, Cambridge: MIT, Press, 1992.
10. Herve Debar, Monique Becker and Didier Siboni "A Neural Network Component for an Intrusion Detection System", Les UhsCedex France, 1992.

11. Ajith Abraham, "Artificial Neural Networks" Stillwater OK, USA, 2005. "
12. Carlos Gershenson, "Artificial Neural Networks for Beginners", United Kingdom.
13. Anil K Jain, Jianchang Mao and K.M Mohiuddin, "Artificial Neural Networks: A Tutorial", Michigan State University, 1996.
14. UgurHALlCI, "Artificial Neural Networks", Chapter I, ANKARA.
15. Eldon Y. Li, "Artificial Neural Networks and their Business Applications", Taiwan, 1994.
16. Christos Stergiou and DimitriosSiganos, "Neural Networks".
17. Limitations and Disadvantages of Artificial Neural Network from website http://www.ncbi.nlm.nih.gov/pubmed/8892489
18. Image of a Neuron form website http://transductions.neti 20 I 0102/04/3 I 3/neuronsl
19. About Artificial Neural Network from website http:// en. wikipedia.orglwikil Artificial_ neural_network
20. RC Chakraborty, "Fundamentals of Neural Networks" myreaders.info/html/artificiaUntelligence.html, June 01, 2010. '
21. Prof. Leslie Smith, "An Introduction to Neural Networks" University of Stirling., 1996,98,2001,2003. '
22. Prof.Dr. Eduardo Gasca A., "Artificial Neural Networks" Toluca ,
23. KishanMehrotra, Chilukuri K Mohan and Sanjay Ranka "Elements of artificial neural network" 1996
24. Weyiu Yi 339229, "Artificial Neu;al Networks", 2005.
25. Vincent Cheung and Kevin Cannons, "An Introduction of Neural Networks", Manitoba, Canada, May 27,2002.
26. Howard Demuth and Mark Beale, "Neural Network Toolbox", with the help of met lab, user guide version 4
27. Girish Kumar Jha, "Artificial Neural Network and its Applications", IARI New Delhi.
28. About Neural Network from website http://en.wikipedia.org I wikiINeural network.
29. About Feed Back Network from website http://www.idsia.ch/-juergen/rnn.html.
30. Sucharita Gopal, "Artificial Neural Networks for Spatial Data Analysis", Boston, 1988.

AUTHORS PROFILE

J. Vijay Chandra has completed his Masters degree in Electrical and Electronics Engineering in 2017. He is currently working as an Assistant Professor at Sri Sivani College of Engineering. Srikakulam, INDIA.

B.Sesha Sai has completed his Masters degree in Electrical and Electronics Engineering in 2014. He is currently working as an Assistant Professor at Sri Sivani College of Engineering. Srikakulam, INDIA.

B.Sateesh Babu has completed his Masters degree in Electrical and Electronics Engineering in 2017. He is currently working as an Assistant Professor at Sri Sivani College of Engineering. Srikakulam, INDIA.

P.Jagannadh has completed his Masters degree in Electrical and Electronics Engineering in 2016. He is currently working as an Assistant Professor at Sri Sivani College of Engineering. Srikakulam, INDIA.