International Journal of Electrical and Electronics Engineers
Vol. No.8 Issue 02, July-December 2016
ISSN (O) 2321-2055
ISSN (P) 2321 -2045

METHODOLOGY FOR DESIGNING LOGIC GATES & CIRCUITS USING McCULLOCH PITTS NEURON

P. Ramesh[1], N. Yadaiah[2]

[1]Electrical and Electronics Engineering Dept, VNR VJIET, Hyderabad, (India)

[2]Electrical and Electronics Engineering Dept, JNTU, Hyderabad, (India)

ABSTRACT

Mc Culloch–pitts neuron is a simplified artificial neuron model which successfully explains excitatory and inhibitory activities of the neurons. All the basic logic gates configuration can be designed using this Model. When the complexity of the logic circuit increases a single layer is insufficient in the design. If the number of variables are greater than the designing of the weights of the networks tends to be complex. Even for multilayered structure the selection of the weights and threshold values has been a ambiguous or challenging task with the increase in the complexity. This paper presents a systematic methodology on the selection of the weights and threshold values, starting with the basic gates (OR, AND, XOR) operation to any complex Boolean network. All the designed networks are verified with the truth tables. With the design rules, the implementation of logic circuits using McCulloch Pitts neurons for complex feed forward networks becomes simpler.

Key Words: McCulloch Pitts neuron, logic Circuits, Feed forward Network Design.

I. McCulloch Pitts Neuron

The first Computational Model (Fig.2) for an artificial Neuron was proposed by McCulloch and Pitts in 1943.

The inputs and outputs are binary (exclusively ones and zeros), the nodes produce only binary results. The activation strength is represented by the weights $W_{1,}$ $W_{2,}$ $W_{3,}$ ……………..W_n. Such a function is described mathematically in the following equations:

Net(Sum) = $\sum_{i=1}^{n} \mathrm{I_i W_i}$

Output = Sgn(net), i.e Output =1 for net ≥ 0, otherwise Output =0(Fig.1).

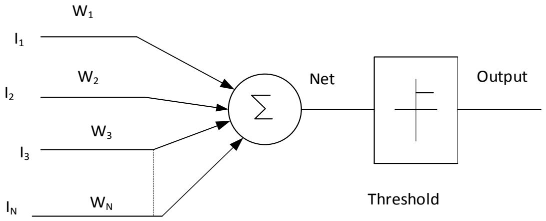


Fig.1. Mc Culloch –pitts neuron Representing Signal flow and the Operation

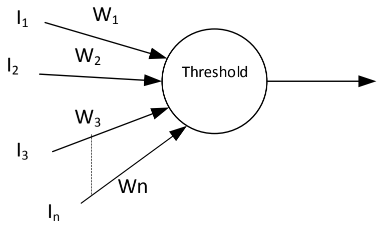


Fig.2. Mc Culloch –pitts neuron

Logic Gates

Logic gates form the essential components of electronic circuits. They have functional importance, which provides the logical platform for providing different levels of currents and voltages in any electronic circuits. They are often called combinational elements in digital circuits because of their capability to combine inputs to provide the output. These are the building blocks for designing a computer architecture[2]. These gates are the AND, OR, NOT, NAND, and EXOR. The goal of logic design or optimization is to find a network of logic gates which performs the desired combination of inputs and outputs.

The basic GATE operations are considered and the corresponding McCullah pitts Neuron architecture is implemented.

AND Gate

Truth Table for AND operation is as follows:

X	Y	X.Y
0	0	0
0	1	0
1	0	0
1	1	1

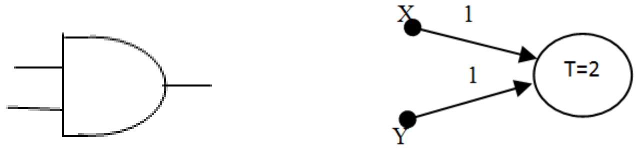


Fig.3 McCulloch Pitts Neuron for AND Gate Network Calculations

X	Y	Net	Output
0	0	0	0
0	1	1	0
1	0	1	0
1	1	2	1

OR Gate

Truth Table for OR Gate is as follows:

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

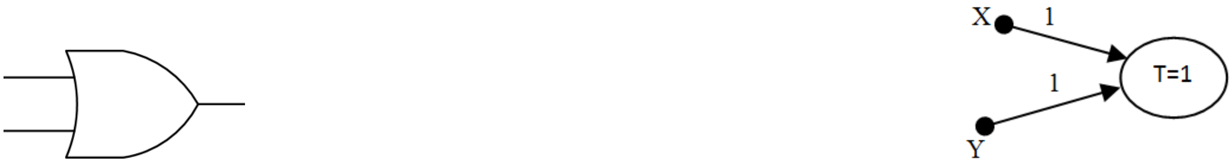


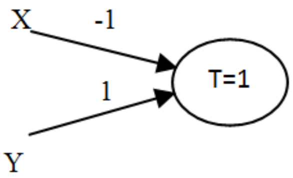
Fig.4 McCulloch Pitts Neuron for OR Gate

Network Calculations

X	Y	Net	Output
0	0	0	0
0	1	1	1
1	0	1	1
1	1	2	1

Logical Operation $\bar{X}$.Y

Truth Table for $\bar{X}$.Y



X	$\bar{X}$	Y	$\bar{X}$.Y
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0

Fig.3 McCulloch Pitts Neuron for $\bar{X}$.Y

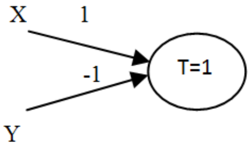
Network Calculations

X	Y	Net	Output
0	0	0	0
0	1	1	1
1	0	-1	0
1	1	0	0

The network Calculations are same as the truth table.

Logical Operation $X.\tilde{Y}$

Truth Table for $X.\tilde{Y}$



X	Y	$\bar{Y}$	$X.\bar{Y}$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

Fig.5 McCulloch Pitts Neuron for $\bar{Y}$.X

Network Calculations

X	Y	Net	Output
	0	0	0
0	1	-1	0
1	0	1	1
1	1	0	0

The network Calculations are same as the truth table.

XORGate

Boolean Expression fro XOR gate is represented by $\bar{X}$.Y+$X.\bar{Y}$.

The truth table is given as follows:

X	$\bar{X}$	Y	$\bar{Y}$	$\bar{X}$.Y+$X.\bar{Y}$
0	1	0	1	0
0	1	1	0	1
1	0	0	1	1
1	0	1	0	0

This is considered as a special case in the sequence of logical operations, where in the solutions is not possible by any combination of the weights of McCulloch Pitts Neuron Model. The explanation for this occurrence can be given in terms of linearly separable problems and linearly inseparable problems:

However, we can see the formulation of XOR Gate as OR Operation of $\overline{X}$.Y and X. $\overline{Y}$. Hence the following two layer network is constructed.

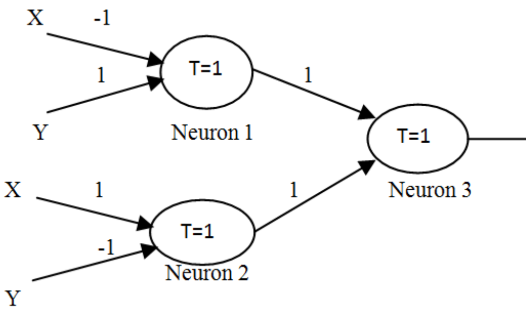


Fig 5. Two layered Structure for XOR Gate

Network Calculations

X	Y	Net1	Net2	Output1	Output2	Net	Output
0	0	0	0	0	0	0	0
0	1	1	-1	1	0	1	1
1	0	-1	1	0	1	1	1
1	1	0	0	0	0	0	0

The network represents the truth table for XOR.

Linearly Separable and Inseparable Problems

A dataset is called linearly separable if the different clusters can be reliably separated by a linear Hyperplane. For two input or two dimensional problem this would be a straight line. Here we examine the condition of linear separability and inseparability for the Gates OR, AND and XOR.

The input patterns are X and Y, the output is classified into two classes, represented by a dark circle for ‘1’ and empty circle for ‘0’. The two input patterns are the two axes for the hyperplane.

II PLOTTING THE DECISION BOUNDARIES

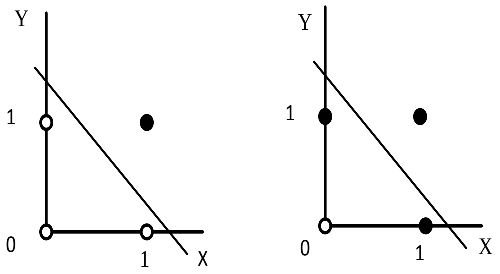


Fig 6.Decision Boundary for AND Gate OR Gate

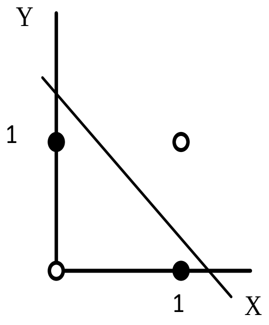


Fig 7. Decision boundary for XOR Gate

As indicated from the figure it always possible to separate the patterns for AND and OR gates by a straight line into two classes. This separation is not possible for XOR gate by a single line. Hence XOR Gate falls in the category of linearly inseparable problem. Therefore this problem can only be solved by two layer network.

Design of logic Circuits with three Inputs

The Boolean expression for three input network is given by $\bar{X}$.Y.Z+ $X.\bar{Y}.Z$

The truth table is given as follows:

X	Y	Z	$\bar{X}$.Y.Z	$X.\bar{Y}.Z$	$\bar{X}$.Y.Z +$X.\bar{Y}.Z$
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	0	0

The expression is seen as the OR operation of two expressions (1) $\bar{X}$.Y.Z and (2) + $X.\bar{Y}.Z.$ Each of the expression is implemented separately and their OR operation is implemented. Therefore, the resulting network is as follows:

X	Y	Z	$\bar{X}.Y.\bar{z}$	$X.\bar{Y}.\bar{z}$	$\bar{X}.Y.Z$	$X.\bar{Y}.Z$	$\bar{X}.Y.\bar{z}+X.\bar{Y}.\bar{z}$ $+\bar{X}.Y.Z+$ $X.\bar{Y}.Z$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	0	0	0
1	0	0	0	1	0	1	1
1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

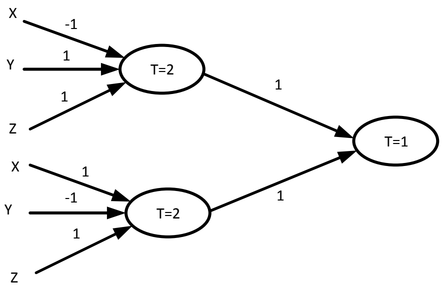


Fig 8. Two layered Structure for Boolean Expression

Network Calculations are as follows:

X	Y	Z	Net1	Net2	Output1	Output2	Net3	Output
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	1	1	1	0	1	0	1	1
1	0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	1	1
1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

It is verified that the network represents the truth table:

III METHODOLOGY FOR CONSTRUCTION OF NEURAL NETWORKS

The Methodology is described for the Boolean expressions in Sum of products form. This assumption is not at the cost of loosing generality because the expressions in Sum of Products and Product of Sum are mutually Interchangeable.

In Product of Sum form, AND operation precedes OR operation.

Considering a Boolean expression with N variables, out of which M variables are having negation operation. The Threshold value is given by N – M

All the weights associated with negation variables are given a negative sign, while the remaining variables are given positive. Next for the OR operation all the weights are 1, and the threshold value is 1.

Testing and Validation of the methodology

We consider a complex three input Boolean expression represented by

$\bar{X}$.Y.$\bar{Z}$+ $X.\bar{Y}.\bar{Z}$+$\bar{X}$.Y.Z+ $X.\bar{Y}.Z$

The truth table is given as follows:

It is verified that the given expression is in Sum of Products form. It is OR operation of four expressions given by:

(1) $\bar{X}$.Y.$\bar{Z}$ (2) $X.\bar{Y}.\bar{Z}$

(3)$\bar{X}$.Y.Z (4) $X.\bar{Y}.Z$.

Considering the first expression $\bar{X}$.Y.$\bar{Z}$. The number of variables are 3, out of which two, $\bar{X}$ and $\bar{Z}$are having negation. Therefore the weights associated with X and Z variables are given negative sign with unity weights.

Threshold (T) = N – M = 3 -2 = 1.

Considering the second expression $X.\bar{Y}.\bar{Z}$

The number of variables are 3, out of which two, $\bar{Y}$ and $\bar{Z}$are having negation. Therefore the weights associated with Y and Z variables are given negative sign with unity weights.

Threshold (T) = N – M = 3 -2 = 1.

Considering the third expression $\bar{X}$.Y.Z The number of variables are 3, out of which one variable, $\bar{X}$ is having negation. Therefore the weights associated with $\bar{X}$ is given negative sign with unity weight.

Threshold (T) = N – M = 3 -1 = 2.

Considering the fourth expression $X.\bar{Y}.Z$. The number of variables are 3, out of which one variable, $\bar{Y}$ is having negation. Therefore the weights associated with $\bar{Y}$ is given negative sign with unity weight.

Threshold (T) = N – M = 3 -1 = 2.

In the layer which corresponds to OR operations all the weights are given unity weights with threshold value to be 1. Hence the network is represented in the following figure:

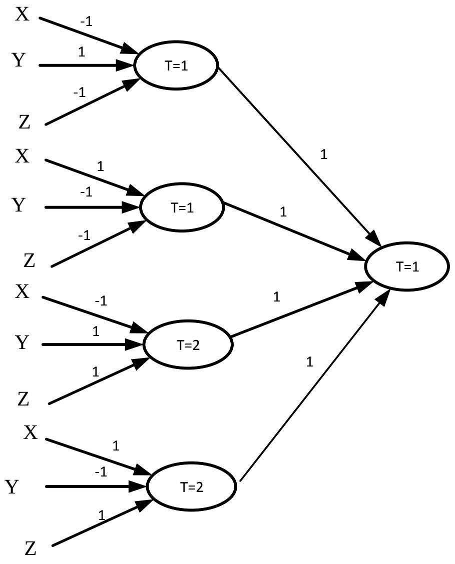


Fig 9. Two layered Structure for Boolean Expression

Network Calculations are as follows:

X	Y	Z	Net1	Net2	Net3	Net4	Output1	Ouput2	Output3	Output4	Net5	Output
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1	0	1	0	1	1
0	1	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	1	1	1
1	0	1	0	0	0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0

The network is same as the truth table.

The principle of homogeneity is applicable for this methodology. If all the weights are scaled by a factor, then the threshold at the respective units should be scaled by the same factor. Thus we can arrive at infinite sets of network representing a given Boolean expression.

IV CONCLUSION

Assignment of the weights for the network is the most crucial in ANN. This work arrives at a systematic methodology for the designing any Boolean expression, starting from the basic gate configuration. Logical operations being the fundamentals units for any electronic system, their design methodology is of utmost importance. This is a generalized method, which can be applied to any Boolean expression with any number of variables. The illustrated methodology is verified at all the neurons, layers and finally on the output, the results are validated with truth tables at every stage.

V FUTURE SCOPE

The design methodology can be extended to the multivariable classification problems. After converting the complex patterns into binary data, multilayer neural networks can be implemented to pattern classification task. This work can also give the estimation on the number of layers and selection on number of neurons in each layer suitable for each prototype problem, further it leads to design of optimized neural network.

REFERENCES

[1] Warren S. Mcculloch And Walter Pitts , A Logical Calculus Of The Ideas Immanent In Nervous Activity , *Bulletin Of Mathematical Biophysics Volume 5, 1943* Pg: 115-133

[2] Cambridge International General Certificate of Secondary Education Computer study Guide 2014.

[3] Adi Ben-Israel And Yuri Levin The Geometry Of Linear Separability In Data Sets Lin. Algeb. And Appl. *Haifa Conference On Matrix Theory*, January 3–5, 2005.

[4] S. N. Sivanandam, S.N. Deepa, Introduction to Neural Networks Using Matlab 6.0, Tata McGraw-Hill Education, 2006.