ISSN 2319 - 6629
Volume 8, No.3, April - May 2019
International Journal of Wireless Communications and Networking Technologies
Available Online at http://warse.org/IJWCNT/static/pdf/file/ijwcnt14832019.pdf
https://doi.org/10.30534/ijwcnt/2019/14832019

Design and Implementation MAXNET Neural Network with MATLAB

Liqaa S. Mezher
Department of Computer Engineering, Al-Mustansiriyah University, Baghdad, Iraq
Email: liqaa14@gmail.com

ABSTRACT

In this paper a software realization of a Neural Network (NN), the present work describes the implementation of the Max - Net algorithm. The performance of a neural network is characterized by the neural network's structure, transfer function, and learning algorithm. The designed architecture is described using MAX –Net neural network classification based on single Layer. The neural network's structure depends on the complexity of the relationship between the input and the output. This no need training part of the neural network, the weight matrix is fixed based on absolute value. This paper discusses the issues involved in implementation of a multi-input neuron with linear excitation functions (poslin) using MATLAB program and MATLAB Simulink neural network tools software implementation is presented.

Key words: Artificial neural network, Max –Net Artificial neural network, MATLAB, Simulink MATLAB of Neural Network.

1. INTRODUCTION

Artificial neural network simulates the movement process of neurons cells of human brain. Specifically, artificial neural network is composed of a large number of neurons linking with each other and automatically change network structure by adjusting the weights of neurons in order to simulate samples data [1].

Artificial neural network or simply neural network is a computational model inspired by the biological nervous system. Neural network is a nonlinear model, which is very simple in computation and has the capability to solve complex real problems including prediction and classification [2].

Max – Net is the type of neural network using feedback, weight matrix depended on absolute value, and not need training.

The original Max-Net yields max-min fairness for a network of homogeneous sources, or general weighted max-min fairness for heterogeneous sources [3].

The paper is organized as in the following. In section two, presents a briefly description the Max – Net neural network about parameters of patterns, weight matrix and the transfer function using positive line. In section three, presents the algorithm of Max – Net neural network. In section 4, presents the design of artificial neural network using Max – Net neural network with three layers. In section 5, presents the case study of Max – Net applied by MATLAB program and Simulink MATLAB using neural network tools, Last section 6, some of the important conclusion are given and future work.

2. MAX-NET NEURAL NETWORK

Max-Net is a neural network for finding the neurons in each row that receives the maximum input would be the winner neuron [4].

The many nodes of inputs layer of Max-Net are completely connected for any input vector, the MAXNET gradually suppresses all but the neuron with the largest initial input. Each neuron is connected to every other neuron in the hidden layer. This type of network is no need for training the network, since the weights are symmetrical, fixed and is given by abs, as shown in Figure 1.

The abs. has to be positive and smaller than 1. The Max-Net based on Winner Take All (WTA) competition policy [5], Winner take-all learning only the weight corresponding to the winner neuron is updated.

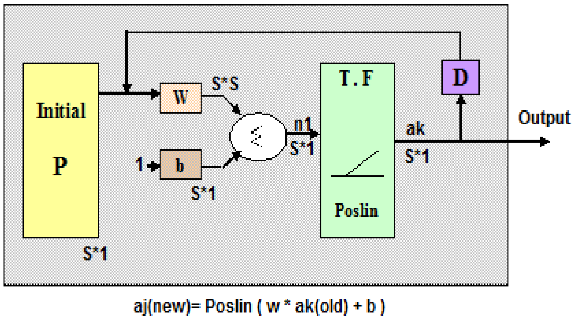

Figure 1: Block Diagram of Max-Net Neural Network

The Max-Net operates as a feedback (a recurrent) recall network that operates in the auxiliary mode. The transfer function used by the neurons is the positive-linear function, as given in (1).

$$f_{Poslin}(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

After a number of recurrences, the only non- zero node will be the one with the largest initializing entry from I/P vector. And the most active neuron is selected by MAXNET, and is used as the final output, as shown in Fig. 2.

3. ALGORITHM OF MAX-NET NEURAL NETWORK

Step 1: Set activation, bias values and choose an abs for the weight matrix (set 0 <abs < 1/M), as given in (2).

$$\text{weights} \ : w_{ji} = \begin{cases} 1 & \text{if } i = j \\ -\varepsilon & \text{otherwise} \end{cases} \quad (2)$$

Where: ei(j) is the i'th component of the j'th exemplar vector.
N: Number of input nodes (input vector components).
M: Number of exemplar vector.

Step 2: For K=1,2,… repeat steps (3 to 6), while stopping condition is false.

Step 3: For each neuron, i=1, 2, …,N,j=1,2,…,M, compute the net signal, it receives for the next step, as given in (3).

$$a_{in,j}(new) = a_j(old) - \varepsilon \ sum_{K\neq j}(a \ (old)) \quad (3)$$

Step 4: Update the activations for i=1,2, …,N, as given in (4).

$$a_j(new) = f_{Poslin}(a_{in,j}(new)) \quad (4)$$

Step 5: Save the activation for use in the next iteration, for i= 1, 2, …, m, as given in (5).

$$a_j(old) = a_j(new) \quad (5)$$

Step 6: Test stopping condition. If more than one node has a nonzero output then go to step, otherwise, stop.

4. DESIGN MAX-NET NEURAL NETWORK

In this paper, the proposed design consists of the neural network used in experiment are given below:

a. **Type**: Max-Net Neural Network.
b. **Number of layers**: Three (input layer, hidden layer and output layer).
c. **Number of input neurons:** 3
d. **Number of neurons in each layer:** 3
e. **Number of absolute value**:0.25
f. **Transfer function**: of the i'th layer: Poslin(x).
g. No need training function.
h. **Number of output neurons**:1
i. **The learning rate** is set to 0.1 (default).
j. **The number of iteration**: 8
k. **The learning algorithm**: Max-Net algorithm.

5. SIMULATION AND RESULT

Input signal transmits from input layer to hidden layer to output layer after a series of function transformation. If analytical results from output layer have large deviations with expected results, modeling using ANN MATLAB R2008a was used to learning the Max -Net ANN developed in this study.

A three-layer Max – Net architecture was used for the ANN model. The poslin(x) transfer function was used for the hidden layer to the output layer, by equation (1).
The ANN architecture consists of three inputs (p1, p2, p3), hidden layers with the optimum number of neurons and a single output variable.

The ANN architecture for Max- Net Neural network is shown in Fig. 2.

In the table.1, run this network by using MATLAB program with the same parameters and eight iterations.

Table 1: Input and Output of Max- Net

No. of iteration	Input A(t)	Output A(t+1)
1	[4;3;4]	[4.25;3;4.25]
2	[4.25;3;4.25]	[4.4375;2.675;4.4375]
3	[4.4375;2.675;4.4375]	[4.6094;2.6563;4.6094]
4	[4.6094;2.6563;4.6094]	[4.793;2.3516;4.793]
5	[4.793;2.3516;4.793]	[5.0068;2;5.0069]
6	[5.0068;2;5.0069]	[5.2551;2;5.2551]
7	[5.2551;2;5.2551]	[5.4413;2;5.4413]
8	[5.4413;2;5.4413]	[5.581;2;5.581]

In Figure 3, represents the output of Max- Net neural network using M file in MATLAB.

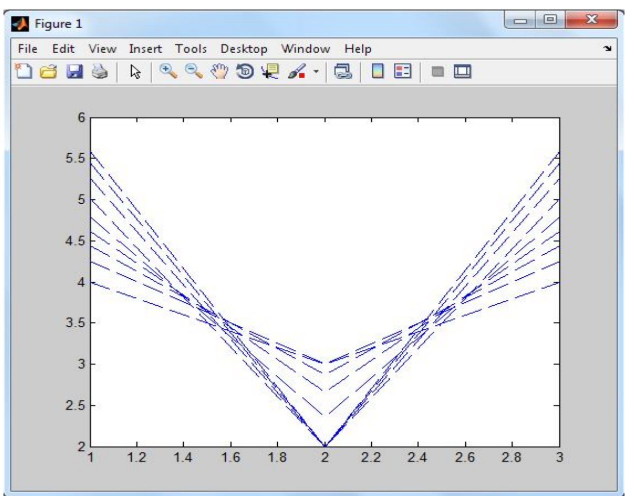

Figure 3: Result of Max-Net Neural Network in MATLAB using (M- file)

In Figure 4, represents the Simulink Max – Net, the general Simulink Max – Net, the set activation input vector of patterns applying in Max- Net neural network and display the output.

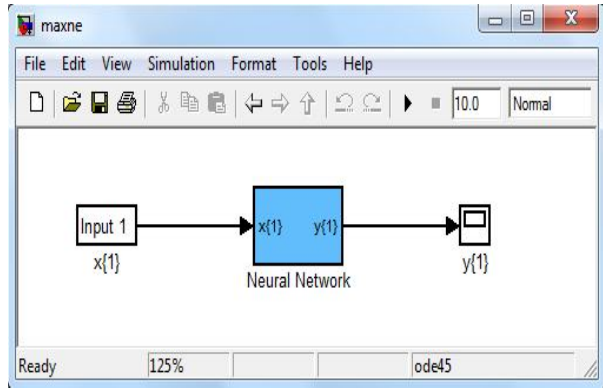

Figure 4: Block Diagram of Simulink Max-Net Neural Network

In Figure 5, represents the Simulink Max – Net, the two layers applied in Max- Net neural network and used feedback.

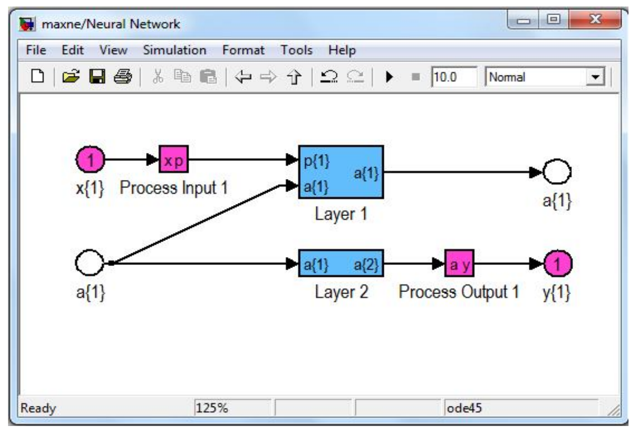

Figure 5: Layers of Simulink Max-Net Neural Network

In Figure 6, represents the Simulink Max – Net, the process between the input values and the input layer in Max- Net neural network and determined the minimum and maximum of the input values.

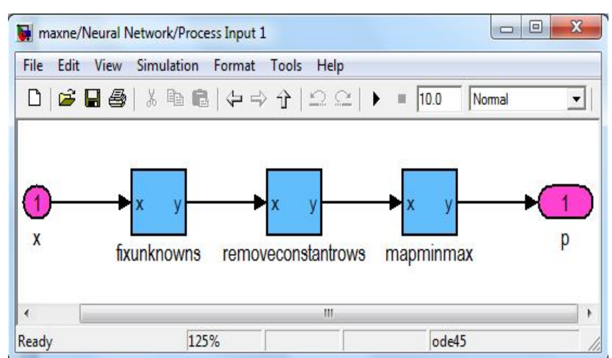

Figure 6: Block Diagram of Process input in Simulink Max-Net Neural Network

In Figure 7, represents the Simulink Max – Net, the multiply from the input values with the weight matrix (matrix weight on the basis of the absolute value of the scope of the absolute value of (0 <absolute <1 / M, where M, where M is the number of exemplar vector). And then combining the result with the bias b{1}, each neuron calculating net signal, it gets to activate the update (transfer function used poslin (x) = x, and the output of the transfer function by using feedback input to another repeat.

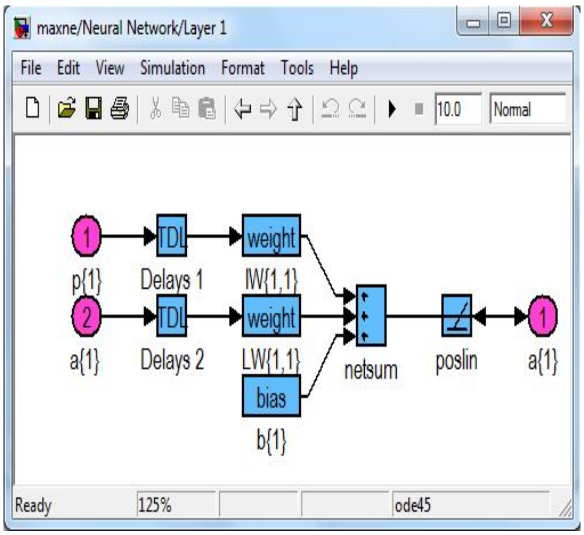

Figure 7: Block Diagram of Layer 1 in Simulink Max-Net Neural Network

In Figure 8, represents the Simulink Max – Net, the delay of layer one in Max- Net neural network =1, but the initial value of input without delay.

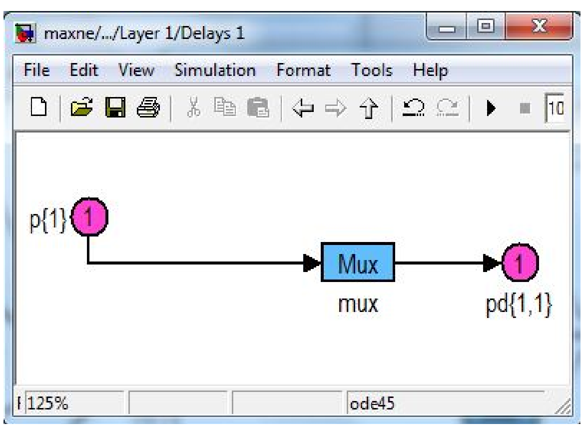

Figure 8: Block Diagram of Delay 1 in Simulink Max-Net Neural Network

In Figure 9, represents the Simulink Max – Net, the dot product from the input values with four neurons in the hidden layer one with input weight {1,1}, and the multiplexer between them.

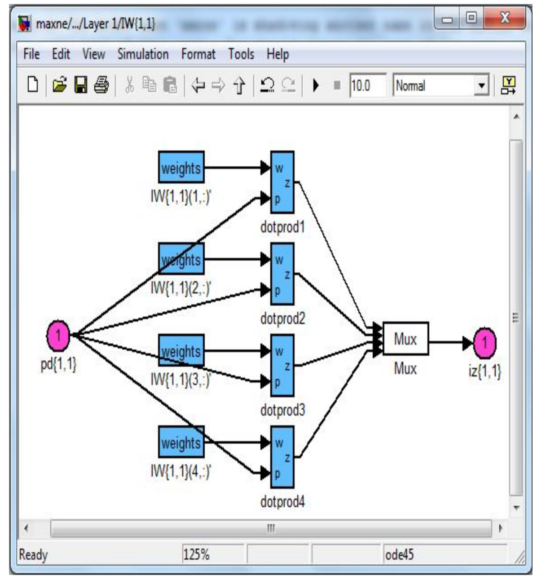

Figure 9: Block Diagram of IW Layer 1 in Simulink Max-Net Neural Network

In Figure.10, represents the Simulink Max – Net, the dot product from the input values with four neurons in the hidden layer one with Lower weight {1,1}, and the multiplexer between them.

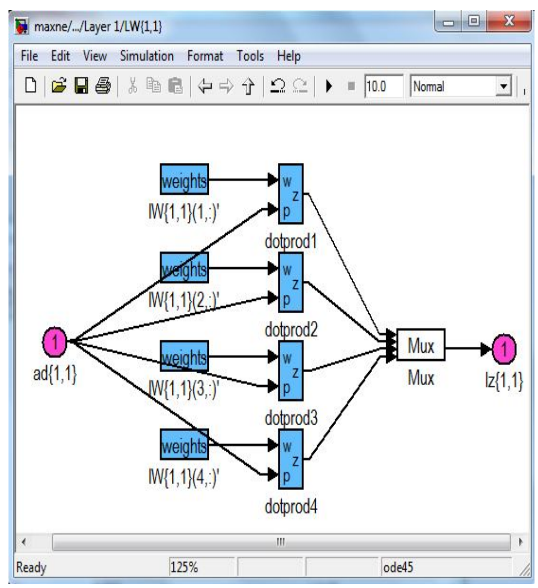

Figure 10: Block Diagram of LW Layer 1 in Simulink Max-Net Neural Network

In Figure 11, represents the Simulink Max – Net, the multiply from the layer one with the weight matrix using single neurons. And then combining the result with the bias b {2}, each neuron calculating net signal, it gets to activate the update (transfer function used purelin (x) = x, and the output of the transfer function by using display of output .

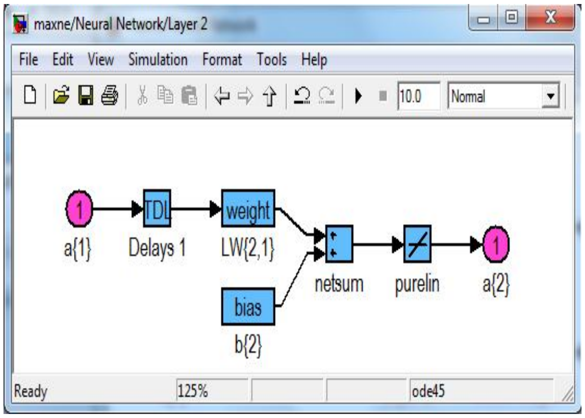

Figure 11: Block Diagram of Layer 2 in Simulink Max-Net Neural Network

In Figure 12, represents the Simulink Max – Net, the delay of layer two in Max- Net neural network = 2.

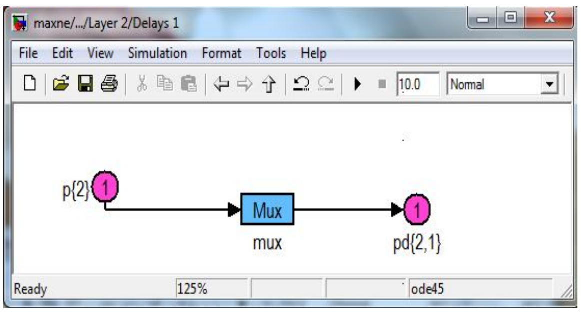

Figure 12: Block Diagram of Delay 1 in Layer 2 in Simulink Max- Net Neural Network

In Figure 13, represents the Simulink Max – Net, the dot product from the hidden layer with single neurons in layer 2 with the lower weight {2,1}by using weight matrix, and the multiplexer between them, then the output iz{2,1}.

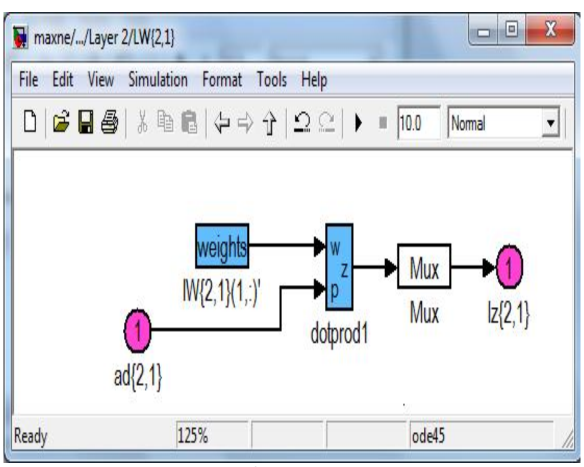

Figure 13: Block Diagram of LW {2,1} Layer 2 in Simulink Max- Net Neural Network

In Figure 14, represents the Simulink Max – Net, the process between the input layer and the output layer in Max- Net neural network and determined the minimum and maximum of the hidden layer values, then display the final output in the Max –Net.

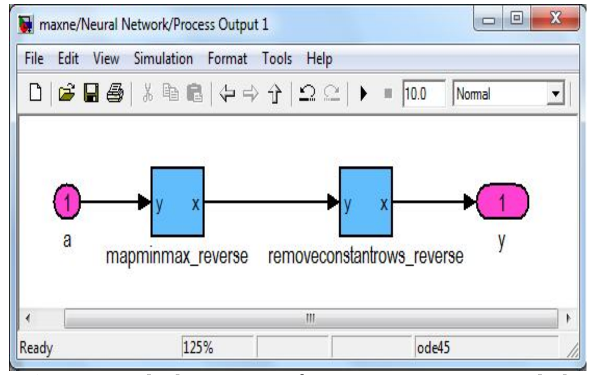

Figure 14: Block Diagram of Process Output in Simulink Max-Net Neural Network

In Figure 15, represents the output of Max- Net neural network using Simulink MATLAB by using neural network.

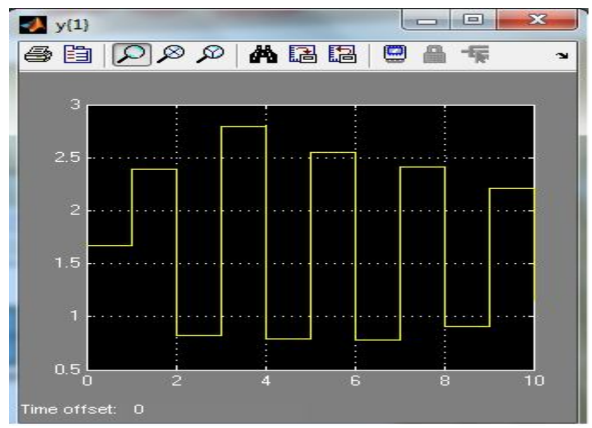

Figure 15: Result of Max-Net Neural Network using Simulink MATLAB

6.CONCLUSION

This paper presents software the design and implementation of a simple Max – Net neural network using MATLAB program and Simulink MATLAB using neural network tools.
This paper implemented using a Multi-Layer neural network with Max – Net algorithm is used in order to reduce the size of the implemented Neural Networks. The network model has a three-layer structure which consists of an input layer, a hidden layer and an output layer. The software implementation is carried out for a neuron with various inputs and activation functions (poslin).

This design may be described as a mapping from the input layer to output layer. The number of neurons, number of hidden layers and number of inputs are easily changed shown to be a very powerful embedded system design tool. Also used the simulation process to verify the design and the synthesis process is used to produce the block diagram. And presented was a new method for approximation of a positive linear function in Simulink MATLAB neural network tool.

REFERENCES

[1] BINGHUI, U., and TINGTING, D., *A Performance Comparison of Neural Networks in Forecasting Stock Price Trend*, International Journal of Computational Intelligence Systems, 2017, Vol. 10, PP 336–346.
https://doi.org/10.2991/ijcis.2017.10.1.23

[2] JOKO, S., ANTON, S., P., AZIZI, A., BAHARI, I., *A Linear Model Based on Kalman Filter for Improving Neural Network Classification Performance*, Journal of Expert Systems With Applications, January 2016, Vol. 11, No. 49, PP 112–122.
https://doi.org/10.1016/j.eswa.2015.12.012

[3] LACHLAN, L., H., A., KRISTER, J., STEVEN, H., L., MARTIN, S., RYAN, W., BARTEK, P., W., *MaxNet: Theory and Implementation,* WAN-in-Lab project,
2006, PP1-11.

[4] Wei-Yen, H., *Clustering-based compression connected to cloud databases in telemedicine and long-term care applications*, Journal of Telematics and Informatics, February 2017, Vol. 34, Issue 1, Pages 299–310.
https://doi.org/10.1016/j.tele.2016.05.010

[5] Hong Q., Zhang Y., and HuaJin T., *A columnar competitive model for solving multi-traveling salesman problem*." Chaos, Solitons & Fractals, 2007, Vol.31,
No. 4, PP1009-1019.
https://doi.org/10.1016/j.chaos.2005.10.059

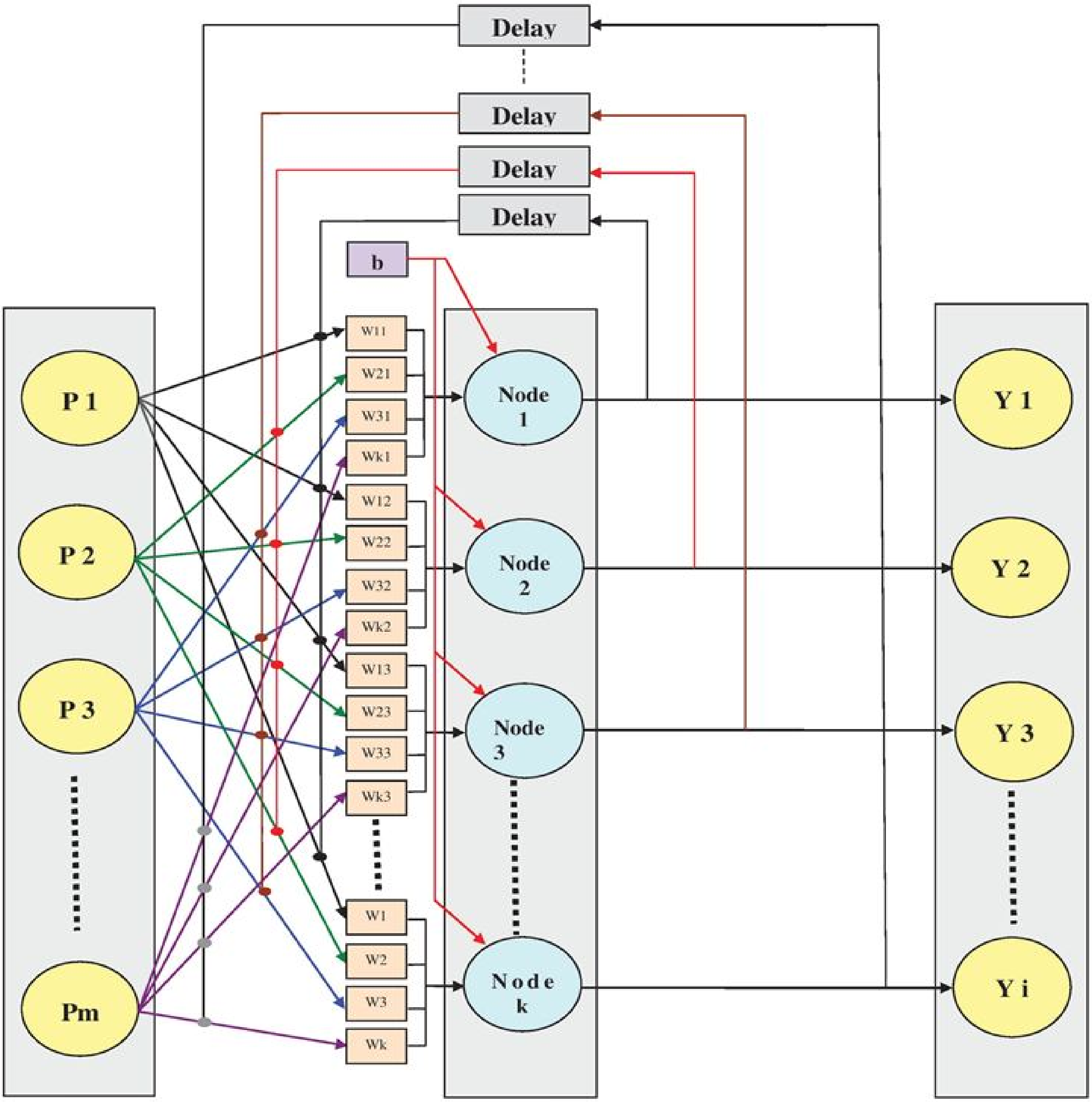

Figure 2: Structure Max-Net Neural Network