Hardware Realization of Artificial Neural Networks Using Analogue Devices

A. I. Khuder , Sh. H. Husain
Department of Electrical Engineering,
College of Engineering, University of Mosul

Abstract

In this work the analogue neural network has been realized by electronic devices like Operational Amplifiers and Field Effect Transistor (FET) . The FET transistor has been utilized to self adjust weight function for neural network. By use of drain and source resistance R_{ds} as function of applied voltage on the gate in linear characteristic region , this resistance has been connected to the input of Operational Amplifiers which becomes as weight function of neural network. Implementing these mentioned characteristics for both FET transistor and operational amplifier , the analogue neural network structure like neuron , weights function and activation function have been realized individually by using the National Instrument multisim 10 (NI) Software, then the analogue neural network has been trained successfully by using supervised learning rule like single layer perceptron learning rule and delta learning rule. The results, show good fulfillment of a neural network with analogue hardware devices and verifying the learning rules to train network.

Keywords: Analogue neural network, BP learning rule, Perceptron learning rule, NI circuit design suits software (NI multisim 10 software) .

تحقيق الكيان المادي للشبكات العصبية الاصطناعية باستخدام الدوائر التناظرية

عبد ألاله خضر و شامل حمزة حسين
قسم الهندسة الكهربائية

ألمستخلص

في هذا البحث تمّ تحقيق الشبكات العصبية الاصطناعية التناظرية باستخدام الدوائر الالكترونية مثل المكبرات العملية (Operational amplifier) والترانزستور تأثير المجال(Field Effect Transistor)[FET] . وتمّ استخدام الترانزستور FET كدالة الضبط الذاتي للأوزان الخاصة للشبكة العصبية بواسطة المقاومة بين المصدر والمصرف ($R_{\rm ds}$) للترانزستور والتي تمثل كدالة للفولتية المسلطة بين البوابة والمصدر $V_{\rm gs}$ للترانزستور والعمل في المنطقة الخطية , والمقاومة $R_{\rm ds}$ تستخدم كمقاومة الإدخال لمكبر العملية (Operational amplifier) والتي تمثل دالة الأوزان للشبكة العصبية . وتمّ تحقيق كل وحدة من وحدات البناء الشبكة العصبية الاصطناعية بشكل منفرد مثل الخلية العصبية (Neuron) ودالة الأوزان (Weight function) ودالة التحويل أو التفعيل (Activation function) باستخدام برنامج المحاكاة National Instrument multisim 10 software . وعملية تدريب الشبكة العصبية بشكل عام نعني بها ضبط الأوزان الخاصة للشبكة , والشبكة العصبية من النوع التناظري تم تدريبها بنجاح باستخدام طريقتين من الطرق التعلم الإشرافي, الأولى Perceptron learning rule التي تنفذ مع الشبكات العصبية ذات التغذية الأمامية وأحادية الطبقة, والثانية هي طريقة Delta learning rule التي تنفذ مع الشبكات العصبية متعددة الطبقات, ولأجل تحقيق ذلك تم استخدام برنامج المحاكاة المذكورة أعلاه . والنتائج التي تمّ الحصول عليها في هذا البحث كانت جيدة .

Received: 13 – 11 - 2011 **Accepted: 10 – 3 - 2012**

1- Introduction:

The artificial neural network (ANN) is an intelligent technique which can be used in signal processing, signal conditioning, signal recognition, and others. It may be implemented and realized by two methods, one is by analogue devices which realize the continuous time signal processing and conditioning circuit which can be realized by FPAA, and the other method is by digital processing or discrete signal processing which realized by software or by FPGA [1] . The analogue devices like operational amplifiers and combination of CMOS or FET transistor can be implemented to realize the architecture of the neural network [2].

There are some problems and drawbacks in hardware implementation of digital combinational circuits for the neural networks verification like FPGA [3] and software [4]. In contrast, analog design has more advantages as compared with digital, these advantages of analogue hardware realizations make it possible to execute the forward pass operation of neural network at high speed, thus making neural network possible candidates for real time application, other advantages lower per unit cost and small size system .The publication on analogue circuit realization on ANN have many ways.

The authors of paper [5] mainly deal with neural network proposed a novel design of arithmetic units that including full-adder, full-subtractor , 2-bit digital multiplier and 2-bit digital divider and neural network has been realized by hardware analogue components like operational amplifier represented as a multiplier , summer and subtractor .

The work presented in [6] explores the relatively new devices double- gate MOSFET for the implementation of the hysteresis neural network. It is the first attempt to use these devices in the neural network field, presenting a large scale application of double gate MOSFETs.

In this work a NI circuit design suits software (NI multisim10 software) has been used to perform and realize each part of the proposed neural network by using analogue devices like operational amplifier and FET transistor, the neural network has been trained by supervised learning method like delta learning rule and perceptron learning rule to self adjust the weights of neural network

2- Basic neural networks:

In general any artificial single layer neural network has a structure shown in figure (1) with (P_1 , P_2 , …. , P_n) are input connected to the neuron body through weights (W_1 ; W_2 , … , W_n) .The neuron output is function of inputs and associated weights as shown in the following equations [7][8].

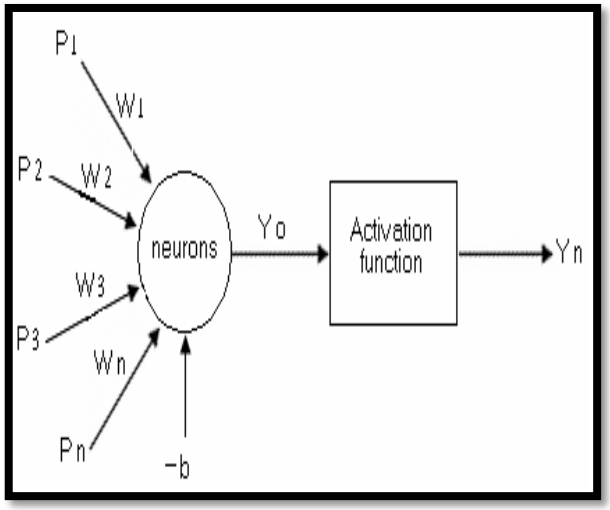


Figure (1) Single layer artificial neural network

$$Y_o = (P * W) - b \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Where

$P = [P_1, P_2, .., P_n]$: Input vector to the neuron.

$W = [W_1, W_2, .., W_n]$: Weight matrix.

b : Bias. Threshold level.

Then result,

$$Y_o = (P_1 * W_1 + P_2 * W_2 + \ldots + W_n * P_n) - b$$

The output of the neuron is entered to the activation function which may be linear or nonlinear function as represent in equation 2.

$$Y_n = f(Y_o) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

To implement each part of neural network by analogue devices like the weight and the neuron body and the activation function, first each part is realized and tested individually then the overall ANN is constructed later.

3- Weights Function Realization of Neural Network:

The major parts of analogue ANN are the weight function, neuron body (summing node) and activation function, the weight function can be realized by:

- **Resistive linear type as a weight function:**

The analogue circuit that implement the simple neural network can be build with operational amplifier as shown in figure 2 , where the inputs are $P = [P_1, P_2, \ldots., P_n]$ and weight function is produced by the ratio R1/Rx, and the output of operation amplifier is given as equation .

$$U = \frac{R_1}{R_{X1}} P1 + \frac{R_1}{R_{X2}} P2 + \text{-------} \frac{R_2}{Rxn} Pn \quad \ldots.(3)$$

This outputs is becomes inputs to the activation function, which in this case linear type [9].

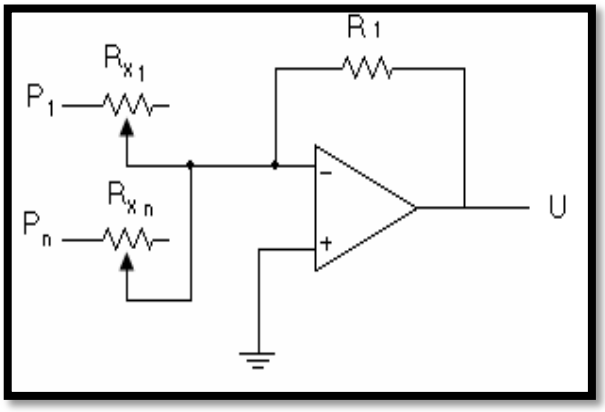
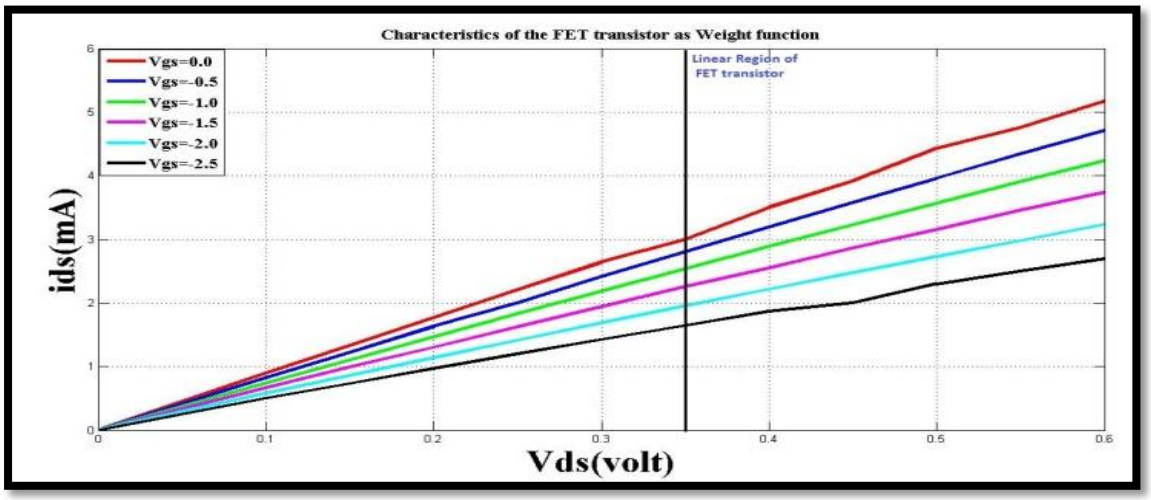


Figure (2) Resistive linear as a weight function of ANN

- Employing FET transistor as a weight function:

The drain to source voltage V_{ds} and drain current I_{ds} characteristic of FET transistor for various value of gate to source voltage V_{gs} has been chosen in the linear region of these characteristic in order to emulate the resistance. Both n_channel type and p_channel type of the field effect transistor (FET) have been used as a weight function of the proposed neural network, in other words the FET transistor has been used to produce a adjusted resistance between Drain and Source R_{ds} as a function of V_{gs} . The V_{ds} and i_{ds} characteristic for the n_channel and p-channel FET transistor are shown in figure 3 and figure 4 respectively [1][9].

Figure (3) n_channel FET transistor characteristic



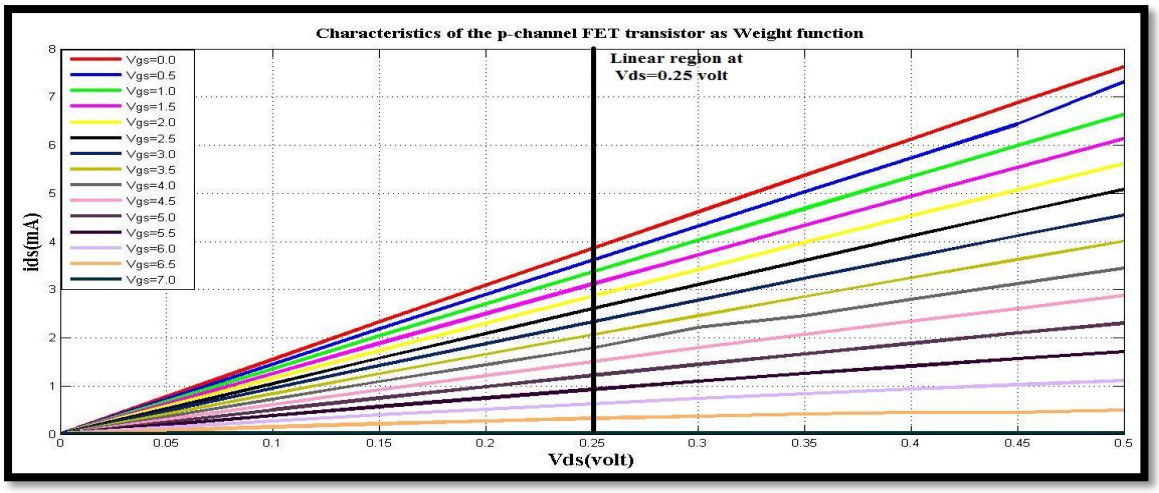


Figure (4) p_channel FET transistor characteristic

The resistance between drain and source R_{ds} for both type n_channel and p_channel of the FET transistor is used weight function of the artificial neural network can be calculated by [1].

$$R_{ds} = \frac{V_{ds}}{I_{ds}} \|_{Vgs} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Theoretically the Drain and Source resistance R_{ds} for the both type of the FET transistor as expressed by equation 5 [1][7].

$$R_{ds} = \frac{R_o}{\left(1-\frac{V_{gs}}{V_p}\right)^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Where R_o is the minimum resistance value when $V_{gs} = 0$.

V_p is the pinch off voltage of the transistor .

The neuron weight can be adjusted by changing the gate to source voltage of the FET transistor V_{gs} then the output has been calculated by equations 6, and weight by equation 7, as shown below :-

$$V_o = -\frac{R_f}{R_{ds}} * X_i$$

$$V_o = -\frac{R_f}{R_o} * (1 - \frac{V_{gs}}{V_p})^2 * X_i \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Therefor

$$\text{Weight}(W) = -K\left(1 - \frac{V_{gs}}{V_p}\right)^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

Where K is a constant.

The result of the resistance value R_{ds} of the FET transistor type n_channel is given in the table (1) for fixed VDS = 0.35 volt and for V_{gs} between 0 volt to – 6volt. Then the resistance value R_{ds} of the FET transistor type p_channel is given table (2) for fixed V_{ds}= 0.25volt and for V_{gs} between 0 volt to 7.5volt.

Table (1) Relationship between $\boldsymbol{V_{gs}}$, $\boldsymbol{R_{ds}}$ and Positive weights of specify n-channel FET transistor

V_{gs}	0	-0.5	-1	-1.5	-2	-2.5	-3	-3.5	-4	-4.5	-4.75	-5	-6
R_{ds}	114	124.8	138.3	156	180	213	263	350	528	1130	1950	35500	36000
W	1.1416	1.132	1.12	1.1	1.087	1.061	1.025	0.967	0.867	0.642	0.5	0.04	0

Table (2) Relationship betweenV_{gs} , R_{ds} and Negative weights of specify p-channel FET transistor

V_{gs}	0	0.5	1	1.5	2	2.5	3	3.5	4	5	5.85	6.7	7.5
R_{ds}	64.84	69.2	74.27	80.3	87.74	96.3	107.2	121.3	140	205.3	330	5500	9000
W	1.186	1.181	1.177	1.17	1.165	1.157	1.147	1.135	1.12	1.067	1	0.55	0.147

The relationship between Gate-Source voltage V_{gs} and positive weights for n-channel FET transistor is shown in figure 5. The relationship between Gate - Source voltage V_{gs} and negative weights for p-channel FET transistor is shown in figure 6.

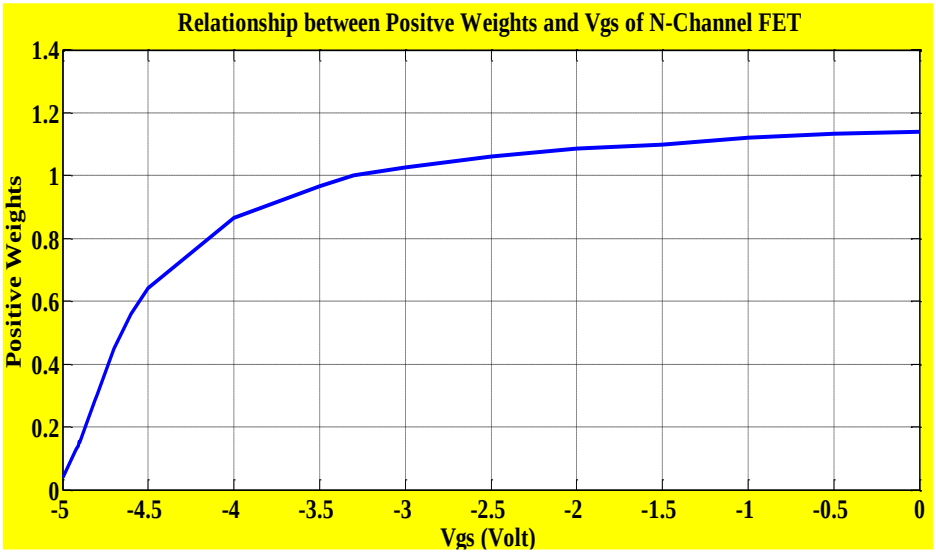


Figure (5) Relationship between V_{gs} and Positive Weights

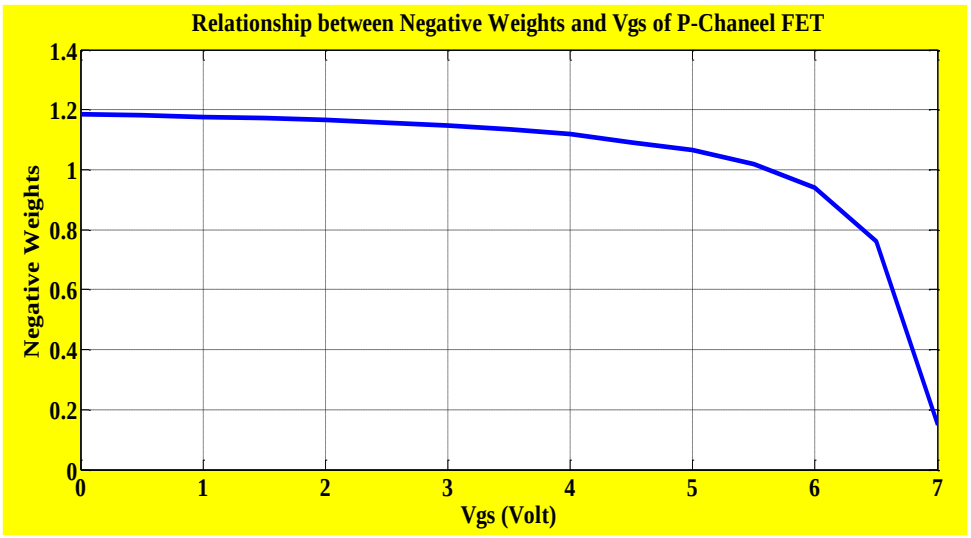


Figure (6) Relationship between V_{gs} and Negative Weights

Characteristics shown in figure (5) and figure (6) can be used to adjust the weights of the ANN by choosing appropriateV_{gs}. The circuit diagram for the weight function of ANN using field effect transistor FET is shown in figure (7).

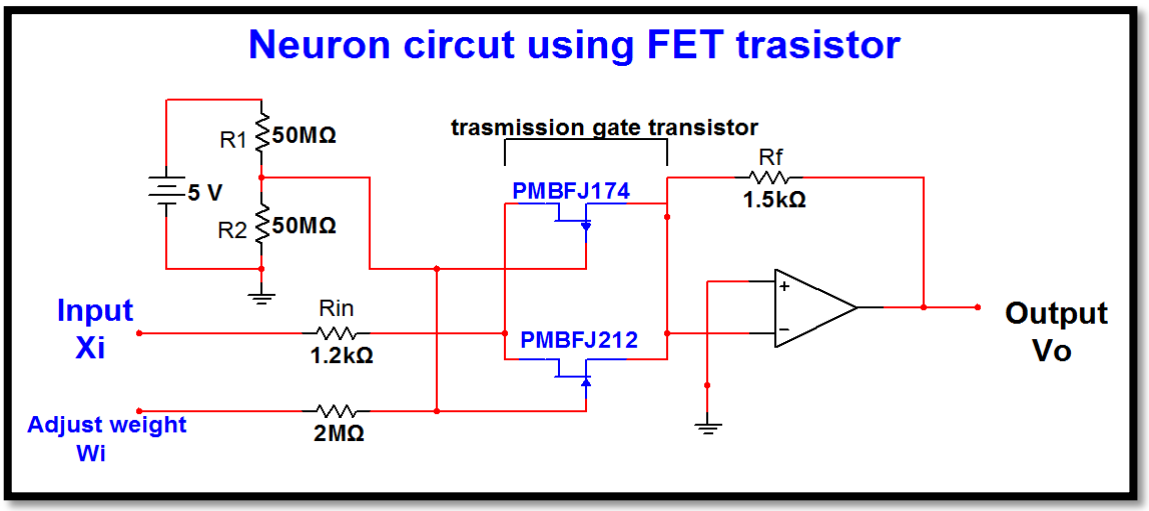


Figure (7) Weight function of ANN using FET

In which the output voltage V_o has been calculated by the following equation.

$$V_o = W * X_i \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

Where

$$R_f = 1.5K\Omega \ , R_{in} = 1.2K\Omega \ , \quad R_{ds} = \frac{V_{ds}}{i_{ds}} \parallel \text{ at Linear region}$$

$$W = \frac{R_f}{R_{ds} + R_{in}}$$

4- REALIZING EACH PART OF THE ANN:

Each part of the proposed analogue artificial neural network has been realized and tested individually by using National Instrument software , these parts can be classified to:

i. Neurons inputs:-

The neurons are built using n_channel and p_channel type of the field effect transistor FET and operational amplifier [1]. Figure 8 shows the construction of ANN and it represents four input artificial neural network realization by using electronic devices like operational amplifier and FET. The circuit diagram of the ANN has been realized by using National Instrument software.

ii. Neuron Activation Function:-

The activation function is chosen such that the output fires, in other words, the activation functions can be defined as a function that transforms the activation level of a unit (neuron) in to an output signal depending on its input/output behavior. The neuron activation functions can be classified in to five broad categories [1][8].

1. Unit step function.
2. Identity function.
3. Linear threshold function.
4. Sigmoid function.
5. Gaussian function.

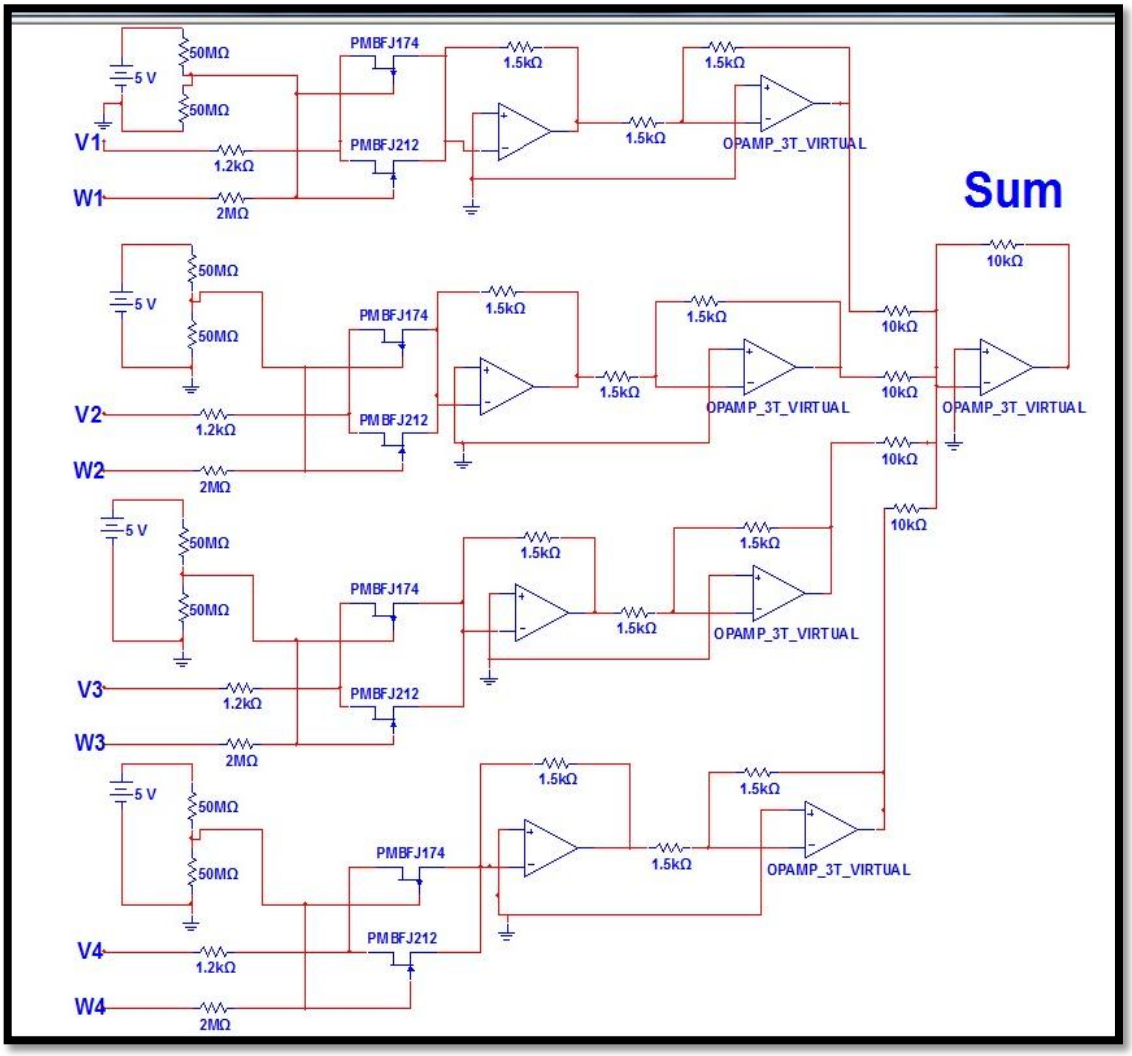


Figure (8) Four input artificial neural network

In this work the sigmoid neuron activation function and linear threshold activation function have been used because sigmoid activation function has a wide range of applications in Sigma-pi and Hopfield neural networks in addition to being employed in multilayer perceptron neural network. While linear threshold function is used in single layer feed forward neural networks and single layer perceptron neural network. The sigmoid function has two major variants that are widely used in the Processing Elements (PE), They are the logistic and the hyperbolic functions [4]. The mathematical definition of the hyperbolic sigmoid function has been used in this work as shown in equation.

$$V = f(\alpha x) = \tanh(x) = \frac{1-e^{-2\alpha x}}{1+e^{2\alpha x}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (9)$$

Where, x is the input voltage α is the voltage gain and V is the output voltage. Theoretically, for positive values of the input voltage x , the function discussed above approaches +1.65 for large values of α. Similarly, for negative values of the input voltage x , the function approaches -1.65 for large values of α. Thus, the hyperbolic sigmoid function resembles the unit step function for high gain values and converges to the unit step function point wise as $\alpha \rightarrow \infty$. But unlike the unit step function the sigmoid function is everywhere

differentiable slope for every α [7][8]. The sigmoid neuron activation function with variable gain is shown in figure (9), the experimental input/output relation of sigmoid neuron activation function is given in figure (10)

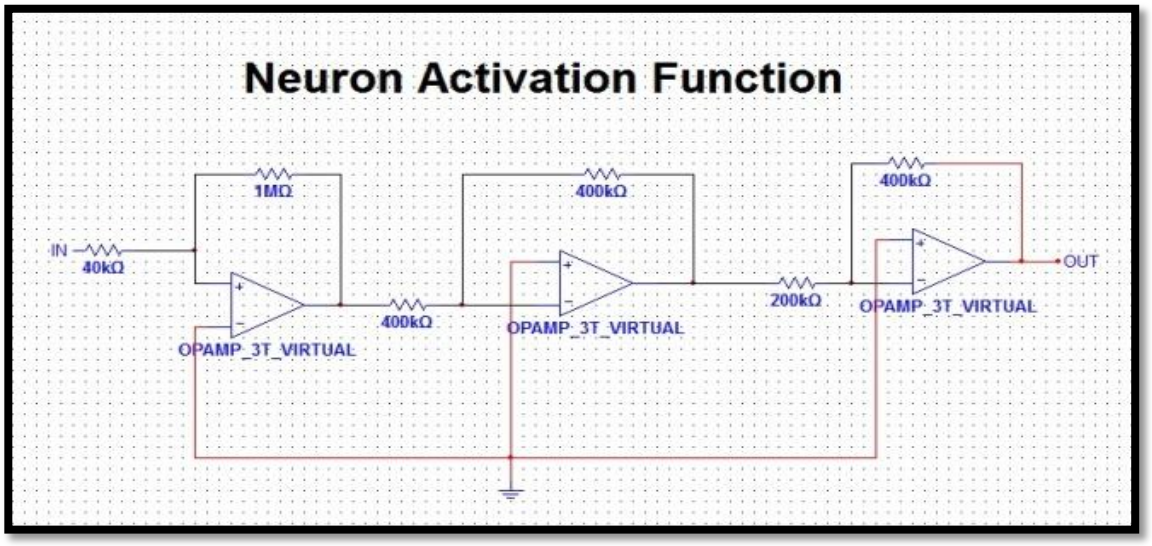


Figure (9) sigmoid neuron activation function with variable gain

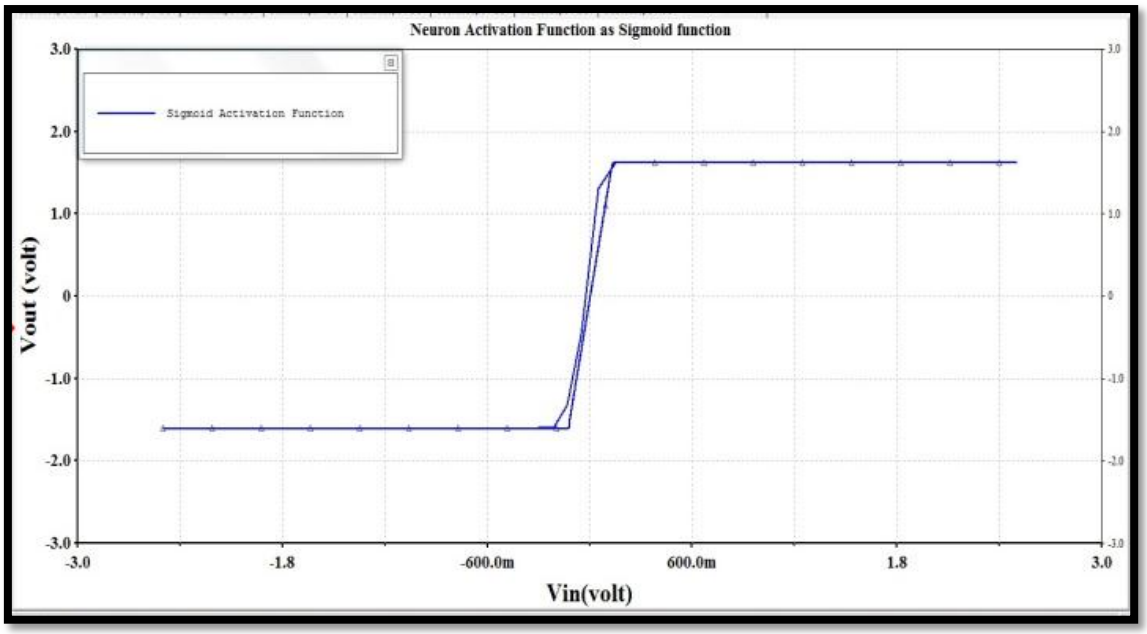


Figure (10) experimental input/output relation of the sigmoid function

On the other hand, the builder electronic circuit that represents linear threshold neuron activation function with variable gain is shown in figure (11). The experimental input output relation are given in figure (12) that realize the linear threshold function [1][8].

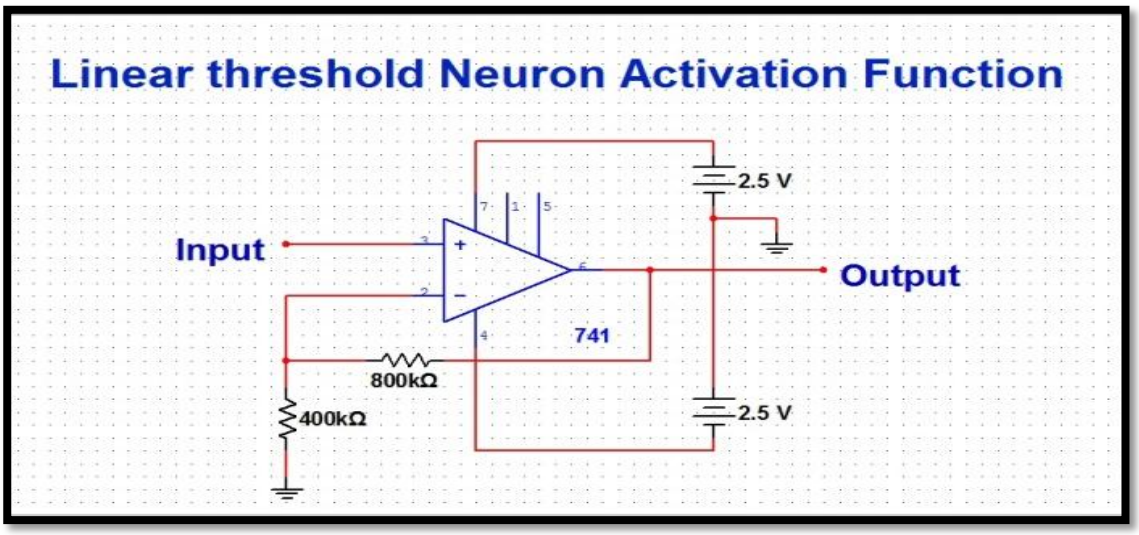


Figure (11) linear threshold Neuron Activation Function

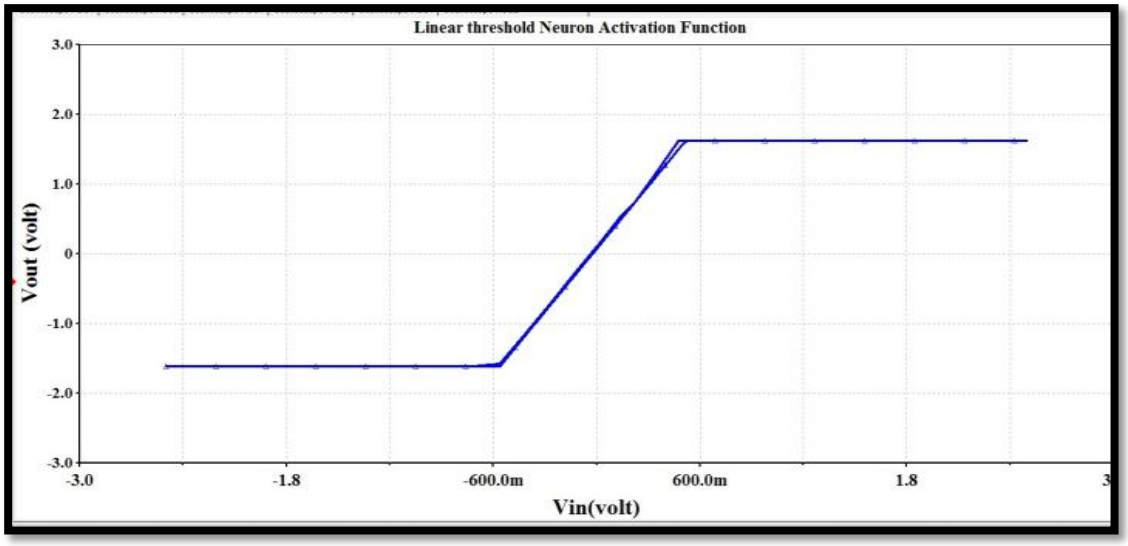


Figure (12) Graphical representation of the linear threshold

iii. Training of the Artificial Neural Network:-

The Back propagation (BP) learning rule and single layer perceptron learning rule have been realized and implemented of the proposed analogue artificial neural network. The demonstration of the limitation of single layer neural network was a significant factor in the decline of interest in neural network in the 1970. The discovery by (several researches independently) and wide spread dissemination of an effective general method of training a multilayer neural network such as Rumelhart , Hinton , Williams, 1986a , 1986b ; MecClelland & bRumelhart , 1988 are played a major role in the reemergence of neural network as a tool for solving a wide variety of problems [1].

The BP networks are among the most popular and widely used neural networks because they are relatively simple and powerful, then a BP networks is a multilayer , feed forward network that is trained by propagating the error between the output of ANN f(net) and the desired value (target value d) using the generalized delta rule [1][9][10].

As a configuration Figure (13) represent the two stage of the BP learning algorithm feedforward stage and backward Propagation stage [2][10]. And figure (14) represent the perceptron learning rule algorithm.

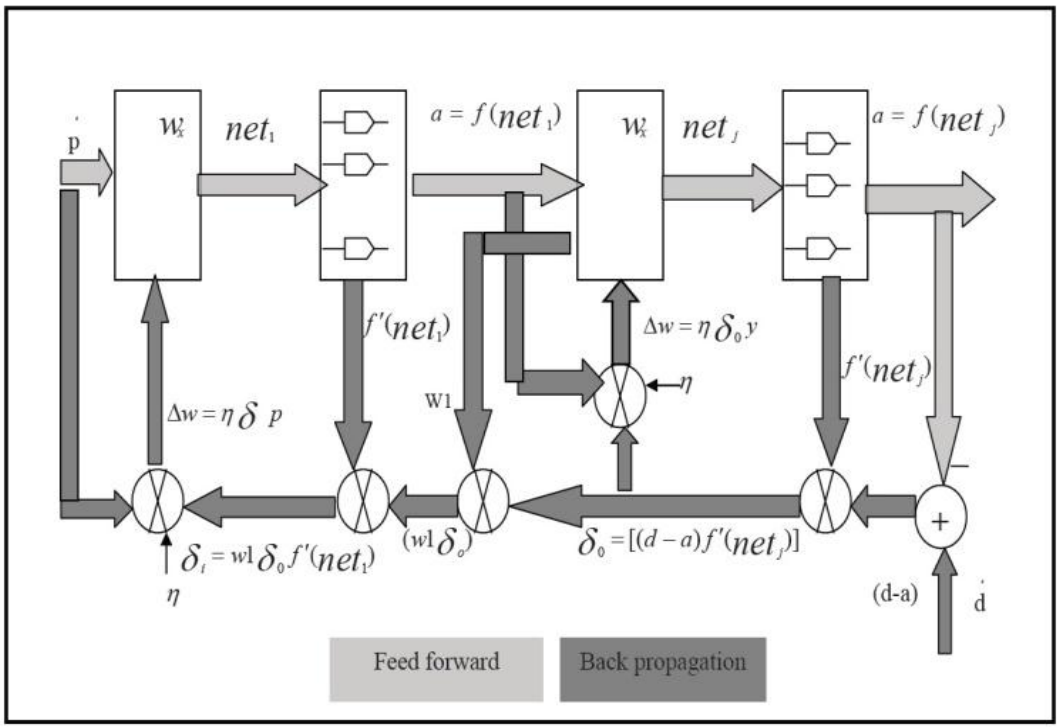


Figure (13) Feed forward and backward stage of the BP learning rule [10]

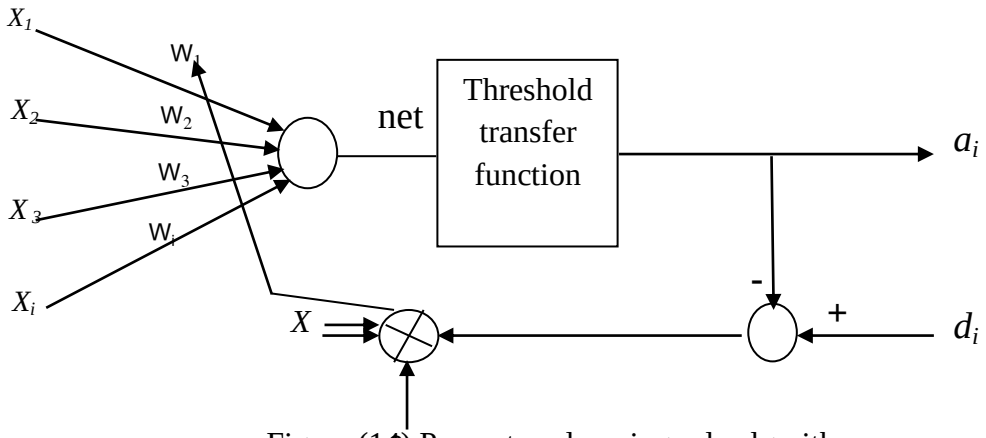


Figure (14) Perceptron learning rule algorithm

The steps of learning algorithms for the BP learning rule can be expressed by following steps [10] :-

Initialize Weights:-
$W_i = [W_1, W_2, W_3, W_4]$
$W_L = [W_1, W_2]$

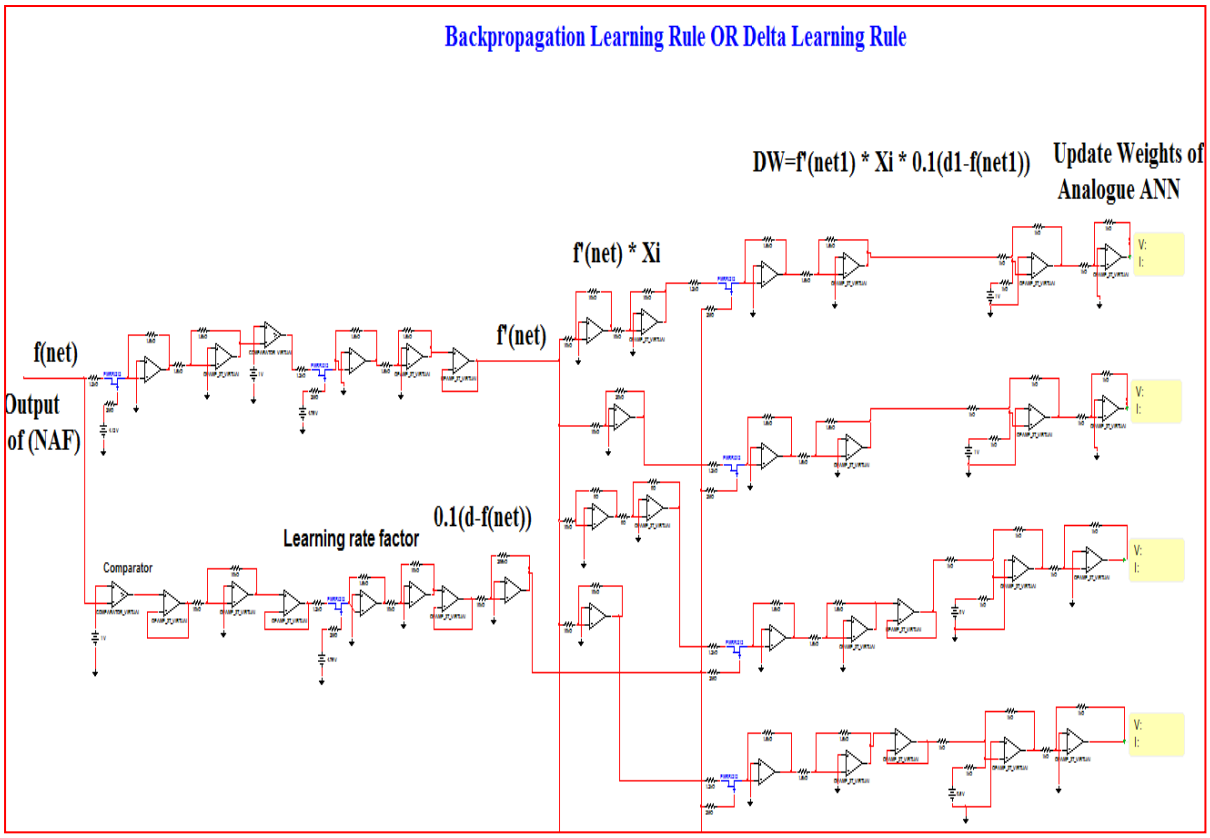
Four inputs of ANN.
$V = [V_1, V_2, V_3, V_4]$
Feed forward steps;

$net^1 = (W'_i * V) + b_i$
$f(net^1) = sgn(net^1)$
$net^2 = (W'_L * f(net^1) + b_L)$
$f(net^2) = sgn(net^2)$
Back propagation steps:-
$\delta_L = ((t - f(net^2) * f'(net^2))$
$\Delta W_L = \alpha * \delta_L * f(net^1)$
$\Delta b_L = \alpha * \delta_L$
$\delta_i = \delta_L * W_L * f'(net^1)$
$\Delta W_i = \alpha * \delta_i * V$
$\Delta b_i = \alpha * \delta_i$
Update Weights and biases:
$W_i(new) = W_i(old) + \Delta W_i$
$W_L(new) = W_L(old) + \Delta W_L$
$b_i(new) = b_i(old) + \Delta b_i$
$b_L(new) = b_L(old) + \Delta b_L$

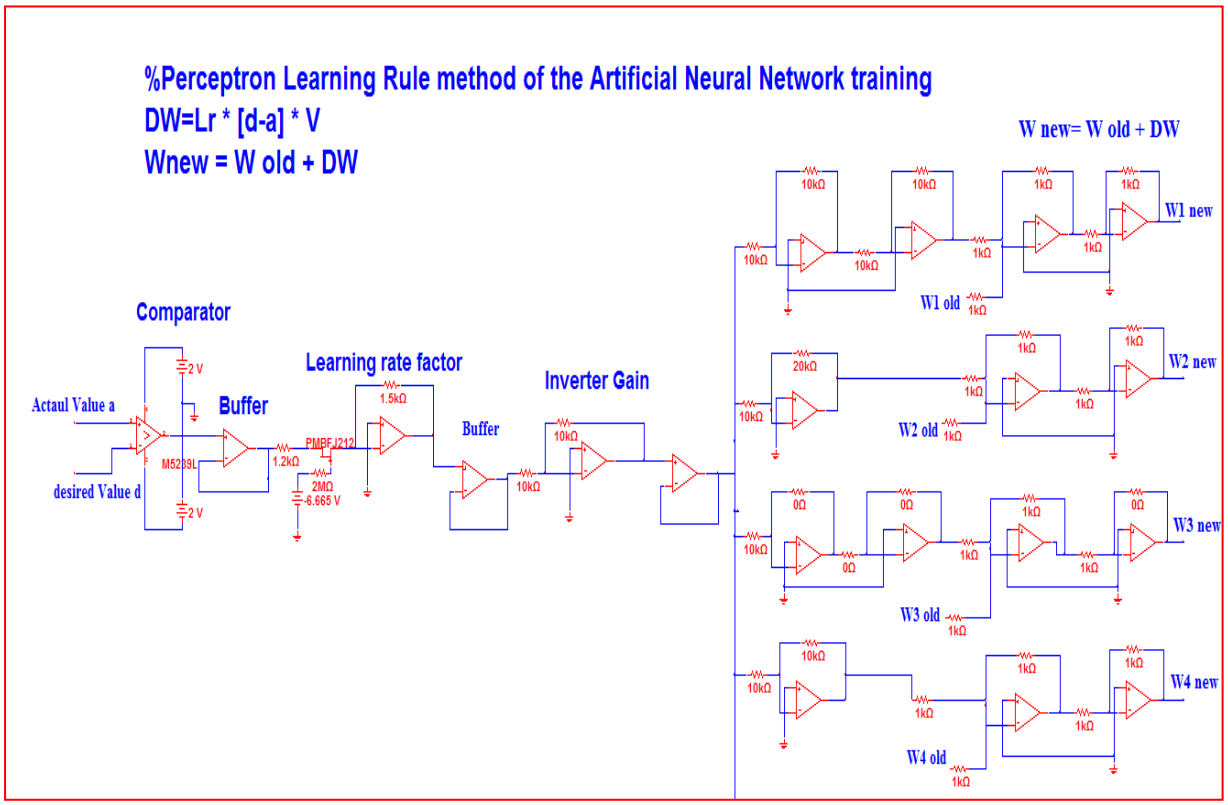
The steps of learning algorithms for the Perceptron learning rule can be expressed by following steps [1]:-

Initialize Weights:-
$W_i = [W_1 , W_2 , W_3 , W_4]$
Four inputs of ANN:-
$V_i = [V_1 , V_2 , V_3 , V_4]$
Perceptron learning procedure:-
$net^i = (W'_i * V_i) + b$
$f(net^1) = hardlim(net^1)$
$E_i = (t - f(net^i)$
$\Delta W_i = \alpha * E_i * V_i$
$W_i(new) = W_i(old) + \Delta W_i$

The Back propagation learning rule and Perceptron Learning rule for training proposed analogue ANN have been realized by using electronic devices like Operational amplifier and FET transistor utilizing the National Instrument software(NI multisim 10 software). The figure 15 represent the update weights of four input ANN, using supervised learning rule, implementing Delta learning rule for trained network, specially the delta learning rule has been realized via backpropagation learning algorithms as shown in figure 13. The updated values of weights depended on propagation delay of the feedforward and backward stages. The figure 16 represent the update weights of ANN by using supervised learning rule, implementing Perceptron learning rule for trained network. This learning rule has been realized by using perceptron learning algorithms as mentioned previously (paragraph previous) .



Figure(15) Realization of the BP Learning rule



Figure(16) Realization of Perceptron learning rule

5- Results and Discussion

To verify realization of analogue ANN and training by supervised learning rules we have tested simple ANN which has four inputs and one output , the initial weight vector W'1 for proposed ANN is assumed to be $W'_1 = [1, -1, 0, 0.5]$, the set of input training vectors is as follows $X_1 = [1 \; -2 \; 0 \; -1]$, $X_2 = [0 \; 1.5 \; -0.5 \; -1]$, $X_3 = [-1 \; 1 \; 0.5 \; -1]$, the learning constant is assumed to be c=0.1 . The teacher 's desired response for X_1, X_2, X_3 are $d_1 = -1$, $d_2 = -1$ and $d_3 = 1$,respectively .The weight vectors for analogue ANN have been updated with varying the set of input training vectors X1,X2,X3. The results for the update weight vectors have been computed by MATLAB NN tool program and by an analogue ANN which is realized by NI Multisim 10 software .these results are given in table (2) .

Table 2. Update weight vectors of ANN for test input vectors utilizing MATLAB program and National Instrument Multisim 10 software

Set of input training for ANN	Digital implementation of ANN		Analogue implementation of ANN	
	Updated weights by BP trained in MATLAB	Updated weights by Perceptron trained in MATLAB	Updated weights by BP trained in NI Software	Updated weights by Perceptron trained in NI Software
	W'_1=[1 -1 0 0.5]	**W'_1=[1 -1 0 0.5]**	**W'_1=[1 -1 0 0.5]**	**W'_1=[1 -1 0 0.5]**
X_1=[1 -2 0 -1]	**W'_2=[0.974 -0.94 0 0.526]**	**W'_2=[0.8 -0.6 0 0.7]**	**W'_2=[1 -1 0 0.5]**	**W'_2=[0.794 -0.586 0 0.7]**
X_2=[0 1.5 -0.5 -1]	**W'_3=[0.974 -0.95 0.002 0.531]**	**W'_3=[0.8 -0.6 0 0.7]**	**W'_3=[1 -0.99 0.002 0.5]**	**W'_3=[0.79 -0.6 0 0.699]**
X_3=[-1 1 -0.5 -1]	**W'_4=[0.947 -0.929 0.016 0.505]**	**W'_4=[0.6 -0.4 0.1 0.5]**	**W'_4=[1 -0.99 0.002 0.5]**	**W'_4=[0.59 -0.382 0.1 0.5]**

As another test, the complete ANN has been constructed by using electronic components such as operational amplifiers and FET transistor utilizing NI multisim 10 program simulator to verify XOR Gate work, this network is called (ANN like XOR gate). The table 3. Represent update weights and biases for input layer and output layer of this network. The initial weight vector V for input layer of proposed ANN is assumed to be $V = [0.1 \, , 0.1 \, , 0.3 \, , 0.2]$, and biases $b_1 = b_2 = 0.5$. The initial weight vector W for output layer of analog ANN is assumed to be $W = [0.1 \, , 0.2]$, and bias $b_3 = 0.5$.

Table 3. Updating weight vector and biases of analog ANN like XOR gate

Weights / Inputs	Update Weights of I/P Layer(Volt)						Update Weights of O/P Layer(Volt)		
	V_{11}	V_{12}	V_{21}	V_{22}	b_1	b_2	W_1	W_2	b_3
	0.1	0.1	0.3	0.2	0.5	0.5	0.1	0.2	0.5
x_1=1,x_2=-1,t_1=1	0.1	0.54m	0.301	0.201	0.507	0.512	0.154	0.275	0.61
x_1=1,x_2=1,t_2=-1	0.1	0.5m	0.285	0.211	0.5	0.51	0.83	0.182	0.6

6- Conclusion:

The ANN can be implemented and realized like operational amplifier and FET transistor which is not very expensive devices and available anywhere.

The practical circuit has been built in this work for very simple four input neural network, where only one chip of operational amplifiers and a little component are used. This network is useful and easy so it can be implemented generally for any neural network. Field effect transistor FET characteristic, the resistance between drain and source (R_{ds}) can be utilized in

the linear region to be as voltage controlled weight function of neural network. Self adjustment of the weight can be done via gate source voltage V_{gs} and self-taught net by use of supervised learning rules. So with more than a few dozen neurons is virtually can be realized and constructed.

Positive values and negative values for weights function of ANN can be realized by using electronic devices like operational amplifier and transmission gate transistors (n_channel FET for positive weights and p_channel FET for negative weights) .

6- References:

[1] Jacek M.Zurada "Introduction To Artificial Neural System" Jaico Publishing House, Second Edition 1996.

[2] Laurene Fausett "Fundamentals of Neural Networks architectures , algorithms and application", Ohio State University , 1992 .

[3] Savran, Aydoğan. & Ünsal, Serkan. "HARDWARE IMPLEMENTATION OF AFEED FORWARD NEURAL NETWORK USING FPGAs", Ego University, Department of Electrical and Electronics Engineering, 2005.

[4] Srinivasan, Vikram, " HDL Descriptions of Artificial Neuron Activation Functions " , Thesis in partial fulfillment of the requirements for the degree of Master of Science, in the Department of Electrical and Computer Engineering and Computer Science of the College of Engineering , University Of Cincinnati, 2005.

[5] El-Bakry, Hazem M. & Mastorakis, Nikos," A Simple Design and Implementation of Reconfigurable Neural Networks ", thesis in partial fulfillment of the requirements for the degree of Master of Science, Mansoura University, Egypt, 2008.

[6] Borundiya , Amit Parasmal, "Implementation of Hopfield Neural Network Using Double Gate MOSFET", Thesis in partial fulfillment of the requirements for the degree of Master of Science, the Russ College of Engineering and Technology of Ohio University, 2008.

[7] Martin T .Hagan, Howard B .Demuth &Mark Beal , "Neural Network Design", china machine press. ISBN 7-111-10841-8/TP-2583, 2001.

[8] Demuth And Beal "Neural Network Tool Box For Use With Matlab" , The Math. Works Inc., MA. USA, 1998.

[9] Robert L. Boylestad, Louis Nashelsy. "Electronic Devices and Circuit Theory", MagroHill, 2000.

[10] Jothiram Jayamani Athreya ," A Component Library of Artificial Neuron Activation Functions" thesis in partial fulfillment of the requirements for the degree of Master of Science, University of Cincinnati , Cincinnati, 2005.

[11] Dr. Basma MK . Younis , Omar Salah Shakar, "Backpropagation Neural Networks", Technical college /Mosul, 2011.

[12] Nada Abd Al-Ghani, "Speaker Recognition Using Neuro-Fuzzy Method" , M.Sc. thesis, University of Mosul/ Iraq, 2004.

The work was carried out at the college of Engineering. University of Mosul