Implementation of Neural Network for High Impedance Fault Detection*

Dhiraj Ahuja

YMCA UNIVERSITY OF SCIENCE & TECHNOLOGY, FARIDABAD, HARYANA, INDIA

ABSTRACT

In the thesis the aim was to detect the high impedance fault occurring on radial distribution system using neural network. A multilayer perceptron was used for distinguishing the linear and nonlinear high impedance faults by taking the feature vector as input R.M.S value of third and fifth harmonic components of feeder voltage and feeder current were used as a feature vector obtained by applying the fast Fourier Transformation on the feeder voltage and feeder current.

The values of feeder voltage and feeder current are obtained for two kinds of fault cases (i.e. linear and nonlinier) by simulating the model of high impedance fault system. The values of third and fifth harmonics were obtained by applying the Fast Fourier Transformation .RMS values of these harmonics were used to train the Multilayer Perceptron Neural Network for classification of these two type of faults. It consists of total three layers, two hidden layers and one output layer. Each hidden layer consists of four neurons and one output layer consists of two neuron. This network was trained by using the Back propagation algorithm .Many types of back propagation algorithms were tested and it's found that trainlm and trainbr were classifying the two kinds of fault s more perfectly compared to other algorithms. As well as for selecting the no. of neurons the network is tested for different number of neuron in each layer and it's found that the network consisting of four neuron in each hidden layer performing well. The network was tested for different transfer function and it was found that it's performance is good when log-sigmoid transfer function is used in all three layers or when tan-sigmoid transfer function is used by the neuron in two hidden layer and linear transfer function is used by neuron in output layer.

***A thesis submitted for partial fulfillment for the award of Master degree of Technology in Electronics Engineering to YMCA UNIVERSITY OF SCIENCE & TECHNOLOGY, FARIDABAD, HARYANA, INDIA**

INTRODUCTION

1.1 Introduction to High impedance fault

High impedance fault (HIFs) on distribution systems creates unique challenge for the protection engineer. HIFs often occur when an overhead conductor breaks down and touches high impedance surfaces (such as asphalt, sand or grass) or where the conductors become in contact with a high impedance object such as tree. The main purpose of in the HIF detection ,in the contrary to short circuit faults, is not to protect system, but to protect the lives and preventing fire hazards due to acting phenomenon[1]. These faults are characterized by intermittent arc-type nature and very low current rich in low harmonic content and high frequency noise spectra.

High Impedance faults are safety hazard to Humans, Live stocks and Electric Utility Personnel. If left undetected for hours, weeks, and possibly months can prove fatal to humans and animals at typical current level of 50 mill ampere or above. HIFs that occur do not produce enough fault current detectable of low-level ground current using any conventional over-current or ground fault type relays[2] is both difficult and sometimes inaccurate therefore need to develop another method to solve this problem engineering effort for the development of a reliable method for the detection of high impedance arc-type faults led during the last two decades to important progress in understanding the electrical characteristics of these faults and in the evaluation of several detection concept[3]. Various technique of fault detection encompasses fractal techniques, expert system, neural networks and dominant harmonic vector [4, 5]. The use of high frequency harmonics is not feasible in practical relay because of the filtering by the subtraction current transformers. Other methods that try to reduce the limitation of frequency domain methods include Kalman filtering [6] and wavelet transforms based methods [7]. Among many technique proposed by different research groups, use of information contained in the low frequency spectral behavior, in terms of both magnitude and phase seems to be the most promising approach for the high impedance arc-type faults on a radial distribution system. In this thesis project work during M.Tech, there shall be a approach for the back propagation algorithm such as nural network

1.2 Literature Review

B.M. Aucoin,B. Don Russell [1] from Texas instrumentation discussed loop holes in Warrington's ground fault detector which used increase in the magnitude of fundamental

harmonic. He revealed that this approach can't work practically as the normal switching events like feeder switching ,capacitor bank switching also produce large increase in lower order harmonics and these will falsely identify the load rich in harmonics as high impedance fault, and they have proposed that high impedance faults exhibit marked increase in the high frequency current components over normal system conditions which persist for the entire duration of the arc ,and for his experimentation on high impedance faults he considered the high frequency current component of frequency greater than 2 Hz. rather then than taking low frequency components which vary widely under different loads. This report became very valuable for the lateral research on High impedance fault detection.Lateral Emanuel ,A.E.,Cyganski,D.,Orr,J.A.,Shiller,S.Gulachenski,E.M[2] from their experimentation work proposed that if we consider the harmonics of very high frequency it's becoming difficult to guess the fault as their magnitude is very low, and suggested that its' better to consider the Second and Third harmonic current as their magnitude is also considerable for detection of the fault and as well as the requirement of much variation in magnitude throughout the duration of high impedance fault will also be fulfilled if second and third harmonics are considered.

Aucoin,B.Michael ,and Jones,Robert H.,[3] considered the implementation issues of High impedance fault detection procedures. In this paper they discussed background research on this topic. They revealed that at the beginning is effective approach is a substation based detector operating on electrical parameters using existing transducers and interrupters. One detector is to be incorporated in each feeder,this cause tripping of entire feeder during fault. Later a microcomputer based technique was developed but one of these were being able to distinguish high impedance fault from other faults. They gave some examples for fault and hazardous conditions. An electrical hazard is a condition in which electrocution or fire may readily occurred the conductor is near ground or an object or an object or within reach a public. A hazardous condition can be one in which no fault is present. An intact primary conductor fallen off insulators hanging one meter above the ground is not a fault but hazardous since it will not convey any electrical information upon which action can be taken .Conversely a circuit may be faulted but may not be hazard. E.g: A tree a limb is contact be a threat through it's a fault. One will expect to have high impedance fault detectors on every feeder to have at most safety, but it's not practically possible. So they have classified few areas for locating the HIFD on each feeder, they are urban and suburban to break: feeders with prior history to downed conductor trouble:

area with heavy tree growth, very dry areas or feeders with numerous, lengthy single phase laterals. They suggested the operating time of HIFD as 15 to 16 sec they also told that it's necessary to bring awareness in the people as most of them can't distinguish among power, telephone and cable T.V lines. Later many researchers started using different harmonic components from fundamental to fifth harmonic.A.M. Sharaf, L.A. Sinder,K.Debnath [4] obtained positive ,negative and zero sequence components of third harmonic components voltage and current values. They have trained the neural network using these components and succeeded in well detection of the high impedance fault compared to this predecessor.L.A.Snider [5-7] in all his papers modeled the high impedance fault by a diode mode; I Sharaf A.M. used no modeling for the fault. He just modeled the transmission line using nominal pi model. All most all the researches using ANN have used Back propagation algorithm for it's simplicity.

A.M Sharaf, L.A Snider.K.Debnath [12] used negative and zero sequence components of second, third,fifth harmonic components for training the neural network. T.M. Lai, L.A. Snider, E.Lo,D.Sutanto [13-14] developed a High Impedance fault detection technique based on Discrete Wavelet transform. But this can't determine the properties of output coefficients.M.M Eissa, G.MA. Sowilam, A.M Sharaf [15] used third and fifth harmonic components of feeder voltage and current obtained by applying FFT for training the neural network. They have also specified some patterns observed in this thesis for obtaining the feature vector which will be used for training the neural network.Howard Demuth, Mark Beale [16] discussed different transfer function, back propagation algorithms.

HIGH IMPEDANCE FAULT SIMULATION

Introduction: When a conductor comes in physical contact with the ground but does not draw enough current to operate protective devices is called High Impedance Fault.

In an overhead wire breaks and falls to ground. If the phase wire misses the grounded neutral or another ground as it falls, the circuit path is completed by the high impedance path provided by the contact surface and earth. The detail of HIF is described in the next section with the help of a radial distribution model of the power system.

2.1 Single Line Diagram Representation of HIF System

The single line diagram of sample system used for the study of high impedance fault detection shown in fig 2.1

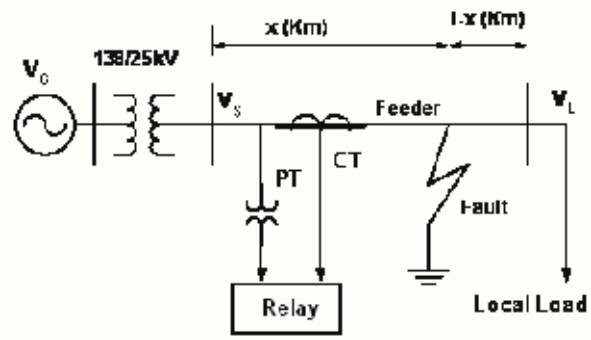


Figure2.1 Single Line Diagram Representation of HIF System

Figure2.1 shows a generator connected to load by transmission line in which an ordinary over current relay is placed.

Where V_G= source voltage.

V_S=voltage generated after excluding the drop in source.

V_F=Voltage at fault point.

V_L=Load Voltage.

I=Feeder length

X=distance of fault location from the source point.

2.2 Per Phase Equivalent Circuit of HIF Model

The per phase equivalent circuit of the above system is modeled is shown below in figure 2.2 it's obvious that the transmission line is modeled using nominal pi model which will be used generally for medium length transmission.

Here

Vs=Source voltage

Ls=Source inductance

Rs=Source impedance

X*R_line=Resistance of the line up to fault point

X*L_line=Inductance of the line up to fault point

X*C_line=Capacitance of the line up to fault point

Lf=Fault inductance

RF=Fault resistance

(1-x)*R_line=Resistance of the line beyond fault point and up to load

(1-x)*L_line=Inductance of the line beyond fault point and up to load

(1-x)*C_line=Capacitance of the line beyond fault point and up to load

R_load=Load resistance

L_load= Load inductance

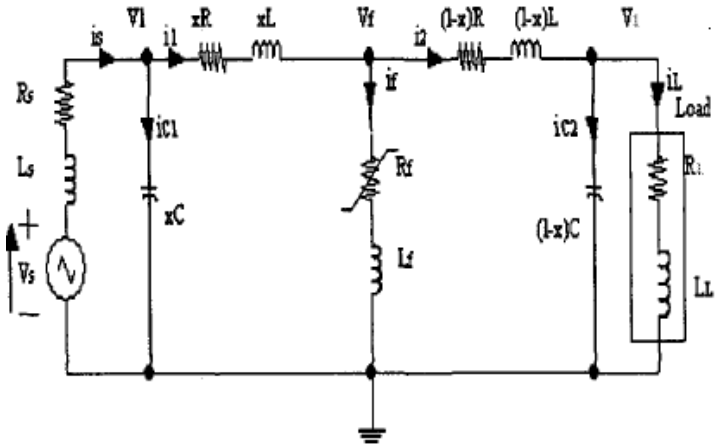


Figure 2.2 Per phase equivalent circuit of HIF model

In the general the high impedance faut can be linear or nonlinear.In case of linear fault the resistance is constant and not a function of any parameter.

In case of nonlinear fault resistance is a function of current and is given by Rf=Rf0(1+α(If/If0)^β) where α and β are constants.

2.3 Simulation Diagram of HIF Model

The simulation diagram is developed for fault from figure 2.2 and it is shown in figure 2.3 given below.

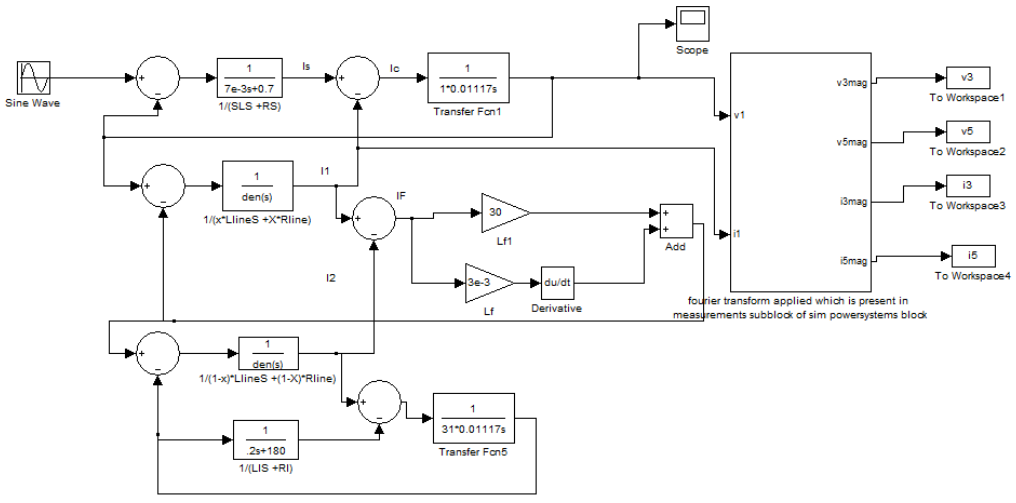


Fig 2.3 MATLAB functional model of High Impedance Linear Fault radial system

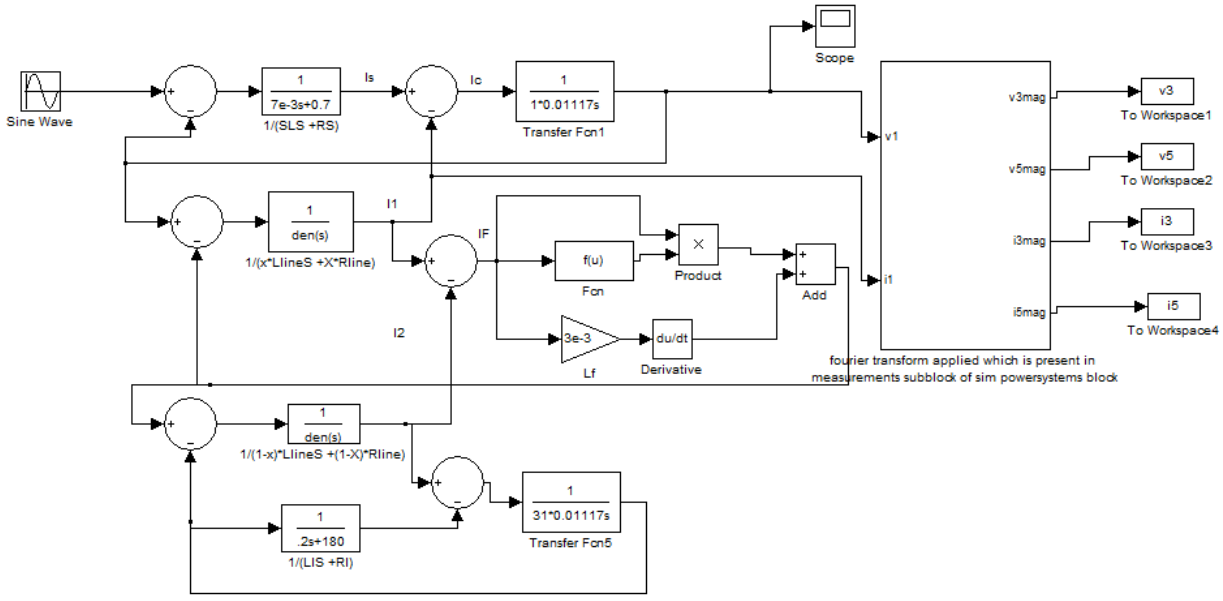


Fig 2.4 MATLAB functional model of High Impedance Non Linear Fault radial system

The diagram shown above is obtained by modeling the per phase equivalent circuit shown in figure2.2 by nodal analysis. The equation used in the modeling of above diagram as shown below:

Source current $(I_S) = \frac{V_S - V_1}{L_S S + R_S}$

Current through capacitor x*C line $(I_{C1}) = I_S - I_1$

$$V_1 = \frac{I_{C1}}{xCs}$$

$$I_1 = \frac{V_1 - V_F}{xL_S + xR}$$

Fault Current $I_F = I_1 - I_2$

$$I_2=\frac{V_F-V_2}{(1-x)L_S-(1-X)R}$$

Current through capacitor (1-x)*C line $I_{C2}=I_2-I_L$

Fault Voltage $V_F=R_F*I_F+L_F*\frac{dI_F}{dT}$

Load Voltage $V_L=\frac{I_{C2}}{(1-x)C_S}$

Load Current $I_L=\frac{V_L}{L_L+R_L}$

2.4. High Impedance Fault Pattern Characteristics:

The radial distribution system is subjected to an arc type fault at different locations by varying x, measured from the substation bus. The voltage and current signals at feeder terminals v1 and i1 are used as detection signals as shown in fig.4. The instantaneous values of these detection signals are captured and transformed in to frequency domain using Fast Fourier Transform FFT. The FFT-harmonic vectors v3, i3, v5, and i5 are processed to obtain feature vectors and are used to train the Back Propagation network.

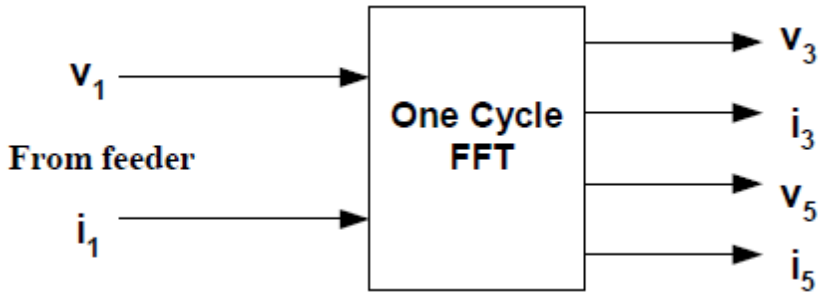


Figure2.5Feature vector extraction

These harmonics components exhibit certain pattern characters using which the linear and non linear high impedance faults can be distinguished .Many cases of linear and nonlinear faults are simulated by varying the fault locations, source impedances and fault resistance .Equal number of linear and nonlinear fault cases is simulated. The obtained data is cast into a classification problem by associating half of the samples into linear and the other half into nonlinear cases.

BACK PROPAGATION ALGORITHM

3.1 General Architecture of Neuron

Frank Rosenblatt devised Perceptron that operated much in the same way as the human mind. Perceptron could "learn" – and that was the breakthrough needed to pioneer today's current neural network adjusting of weights to produce a particular output is called "training" of the network which is the mechanism that allows the network to learn.

A neuron with R inputs is shown below. Each input is weights with an appropriate w.The sum of the weighted inputs and the bias forms the input to the transfer function f. Neuron can use any differentiable transfer function f to generate their output.

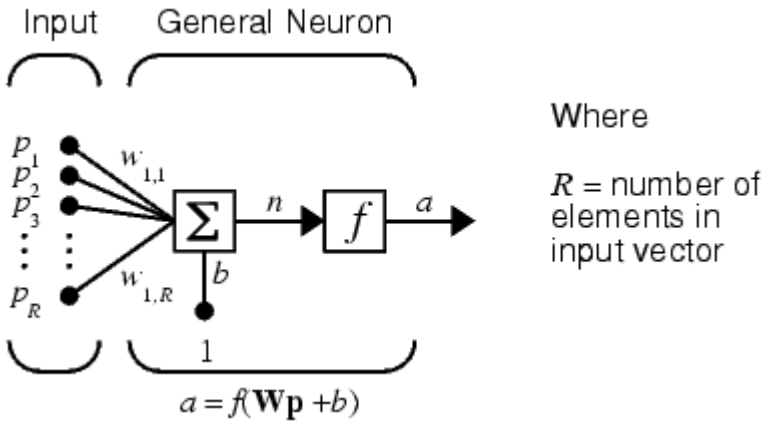


Fig3.1 An elementary neuron with R input

The sum of the weighted inputs with bias form the net input to the transfer function φ and is given by the following expression.

n=$p_1 w_{11} + p_2 w_{12} + p_3 w_{13} \ldots\ldots\ldots\ldots\ldots\ldots\ldots p_r w_{1r}$+b

Any diffrentiable transfer function can be used by the neuron to grnerate the output.

3.2 Types of transfer function:

There are many types of transfer functions use in the literature ,but generally only four types of transfer function will be used.There description is given below.3.2.1. Log sigmoid transfer funtion:The **log sigmoid transfer function** is shown in figure 3.4 given below.

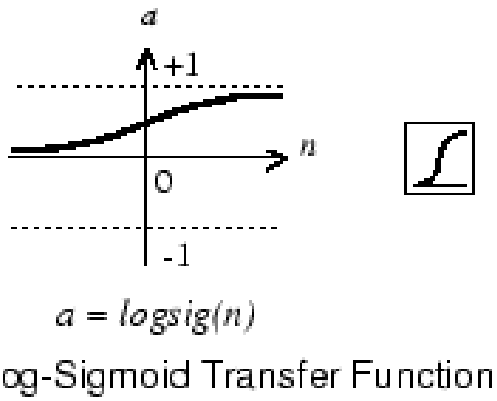


Figure 3.2 Log Sigmoid Transfer Function

The function logsig generates outputs between 0 and 1 as the neurons net input goes from negative to positive infinity.This transfer function is commonly used in back propagation networks since it is differentiable,

3.3 Tan sigmoid transfer function:The log sigmoid transfer function is shown in figure 3.5 given below.

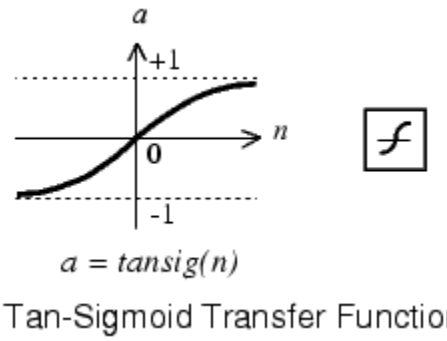


Figure 3.3 Tan sigmoid Transfer function

The tan sigmoid transfer function takes any value between negative to positive infinity as the input and also the output will be in the rang negative to positive infinity.

3.4 Linear transfer function: The linear transfer function does not change the output.It only passes the net output.The linear transfer function shown below.

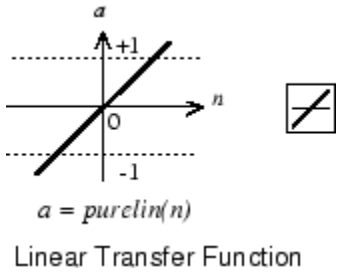
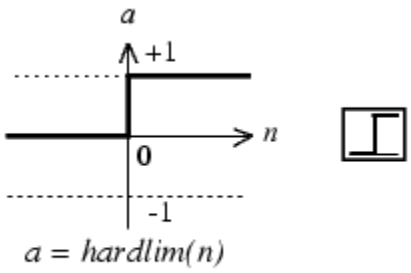


Figure 3.4 Linear Transfer Function

3.5 Hard limiter transfer function : The hard limiter transfer function shown in figure 3.5 below. This function limits the output of the neuron to 0 or 1 depending on the net input applied

to transfer function. If the net input n is less than 0 output of the transfer function is 0 and if n is greater than or equal to 0 the output is 1.



Hard-Limit Transfer Function

Fig 3.5. Hard Limiter Transfer Function

The multilayer perceptron is shown in fig 3.5 which have equal number of neuron in all 3 layers. A network can have several layers. Each layer has a weight matrix **W**, a bias vector **b**, and an output vector **a** ,referring to figure 3.6 each neuron receives a signal from the neurons in the previous layer ,and each of those signals is multiplied by separate weight value. The weighted inputs are summed ,and passed through activation function is then broadcast to all of the neurons in the next layer. So,to use the network to solve a problem, we apply the input values of the first layer allow the signal to propagate through the network, and read the output values. In the given figure the activation function is sigmoid function.

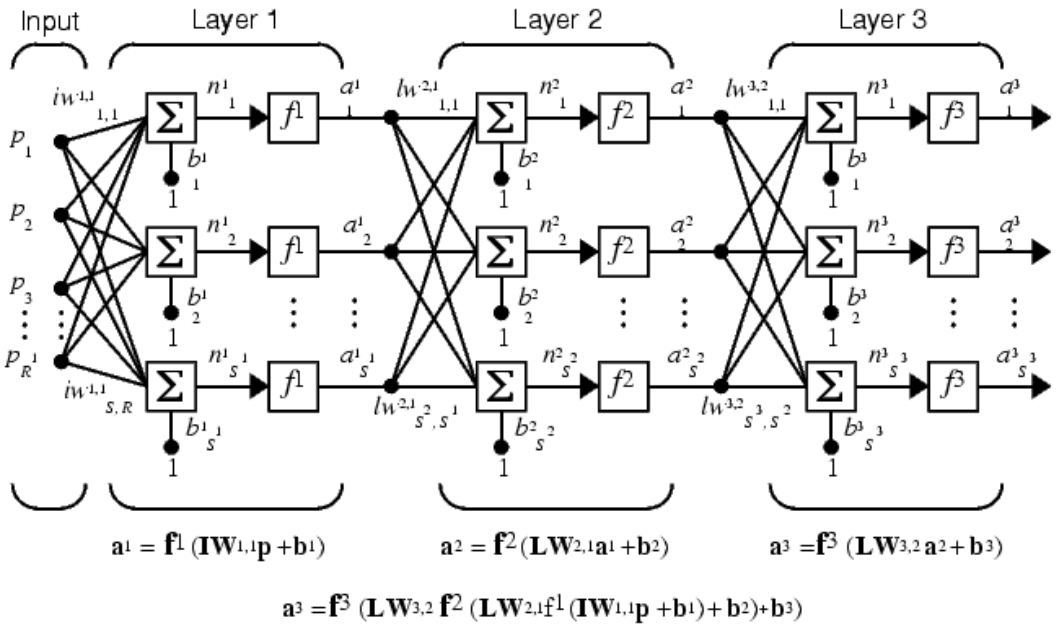


Figure 3.6: Representation of multilayer Perceptron architecture

The back propagation learning process works in small iterative steps: one of the example cases is applied to the network, and the network produces some output based on the current state of its synaptic weights (initially, the output will be random).This output is compared to the known to the known-good output ,and a mean squared error signal is calculated. The error value is then propagated backward through the network, and small changes are made to the weights in each layer. The weight changes are calculated to reduce the error signal. The whole process is repeated for each of the example cases, then back to the first case again and so on. The cycle is repeated until the overall error value drops below some pre-determined threshold. At this point we say that the network has learned the problem "well enough" –the network will never exactly learn the ideal function, but rather it will generalized the ideal function.

3.6 Back Propagation Algorithm

1. Start with random weights

2. Repeat

*For each example e in the training set do

a. O= neural net output; forward pass

b. T=teacher output for e

c. Calculate error (T-O) at the output units

d. Compute Δw for all weights from hidden layer to output layer; backward pass

e. Compute Δw for all weights from input layer to hidden layer to hidden layer ;backward pass

Continued

f. Modify the weights in the network

*End

3. Until classified correctly or stopping satisfied

4. Return

3.7 Derivation of Back Propagation

Let's begin with the root mean square of the errors in the output layer defined as:

$$E=\frac{1}{2}\sum(d_i - o_i)^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Here d_i= target vector = t_i & o_i= output of neuron

The net value is obtained by sum of weighted activations coming in to neuron i, before squashing

$$x_i = \sum_j wijaj \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

$$a_i = \frac{1}{(1+e^{-xi})}$$

Here a_i= activation of neuron i

w_{ij}=synaptic weight from neuron j to neuron i

x_i=excitation of neuron i

With the chain rule, we can obtain the rate of change in the RMS error E in respect to weight change:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial 0_i} \frac{\partial 0_i}{\partial X_i} \frac{\partial X_i}{\partial w_{ij}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

$$\frac{\partial E}{\partial o_i} = \frac{1}{2} * 2 * (d_i - o_i)(-1) = (0_i - t_i) \ldots\ldots\ldots\ldots\ldots(4)$$

$$\frac{\partial o}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\frac{1}{(1+e^{-xi})}\right]$$

$$= -[{}^{1}/{(1+e^{-xi})^2}](e^{-xi})$$

$$= \frac{-e^{-xi}}{(1+e^{-xi})^2}$$

$$= \frac{(1+e^{-xi})-1}{(1+e^{-xi})} * \frac{1}{(1+e^{-xi})}$$

$$= 1 - \frac{1}{(1+e^{-xi})} * \frac{1}{(1+e^{-xi})}$$

$$= (1-o_i)\, o_i \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

$$\frac{\partial o}{\partial a_i} = a_i \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(6)}$$

However, the first term is more difficult to understand for this hidden layer. It is what Minsky called the credit assignment problem, and is what stumped connectionsists for two decades. The trick is to reliaze that the hidden nodes do not themselves make errors; rather they contribute to the errors of the output nodes. So, the derivative of the total error w.r.t a hidden neuron's activation is the sum of that hidden neuron's activation is the sum of that hidden neuron's contributions to the errors in all the of the output neurons:

$$\frac{\partial E}{\partial a_i} = \sum_K \frac{\partial E}{\partial o_K} \frac{\partial o_K}{\partial x_K} \frac{\partial x_K}{\partial a_i}$$

(Where k indexes overall output units)

$\frac{\partial E}{\partial o_K}$ = contribution of each output neuron

$\frac{\partial o_K}{\partial x_K}$ =contribution of all inputs to the output neuron (from the hidden layer)

$\frac{\partial x_K}{\partial a_i}$ =contribution of the particular neuron in the hidden layer

From our previous derivations , the first two terms are easy;

$$\frac{\partial E}{\partial o_K} = (o_k - d_k)$$

$$\frac{\partial E}{\partial o_K} = (1 - o_k)\, o_k$$

For the third term:

$$x_k = \sum w_{ki} a_i$$

And since only one member of the sum involves a_i:

$$\frac{\partial x_k}{\partial o_i} = w_{ki}$$

Here w_{ki}= weight between hidden and output layers

And

$$\delta_k = (d_k - o_k)(1 - o_k) o_k w_{ki}$$

And combining with previous results yields:

$$\frac{\partial E}{\partial w_{ij}} = -\sum w_{ki} \delta_k (1 - a_i) a_i a_j$$

$$w_{ij}{}^{t+1} = w_{ij}{}^{t} + \eta(\sum_k \delta_k w_{ki}) (1 - a_i) a_i a_j$$

Thus For output neuron:

$$w_{ij}{}^{t+1} = w_{ij}{}^{t} + \eta(d_i - o_i)(1 - o_i) o_i a_j$$

For hidden neuron: $w_{ij}{}^{t+1} = w_{ij}{}^{t} + \eta(\sum_k \delta_k w_{ki})(1 - a_i)a_i a_j$

Here

$\delta_k = ((d_k - o_k)(1 - o_k)o_k w_{ki}$

Thus new weight is obtained.

3.8 Description of Back Propagation Algorithm using 3 layer neural network

To illustrate this process the three layer neural network with two inputs and one output, which is shown in figure 3.8 below is used:

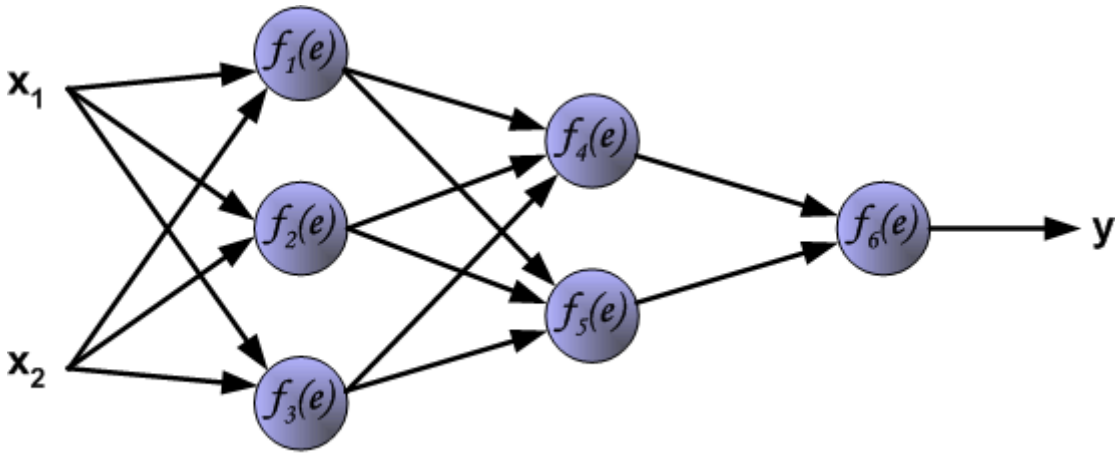


Figure 3.7: Three Layer Neural Network

Each neuron is composed of two units. First unit adds product of weights coefficients and input signals. The second unit realizes non linear function, called neuron activation function. Signal e is adder output signal, and y =f(e) is adder output signal of non linear element .Signal y is also output signal of neuron.

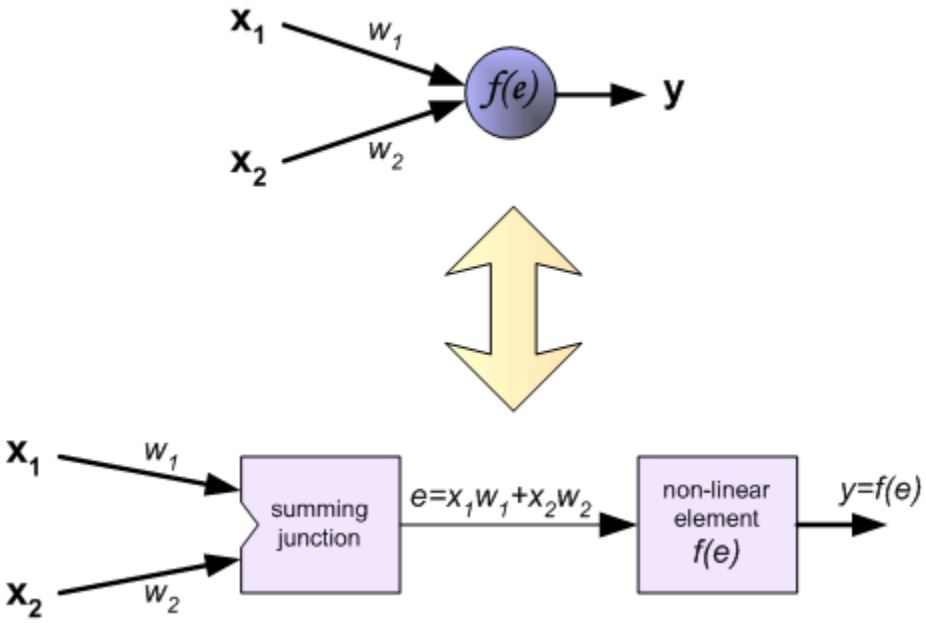
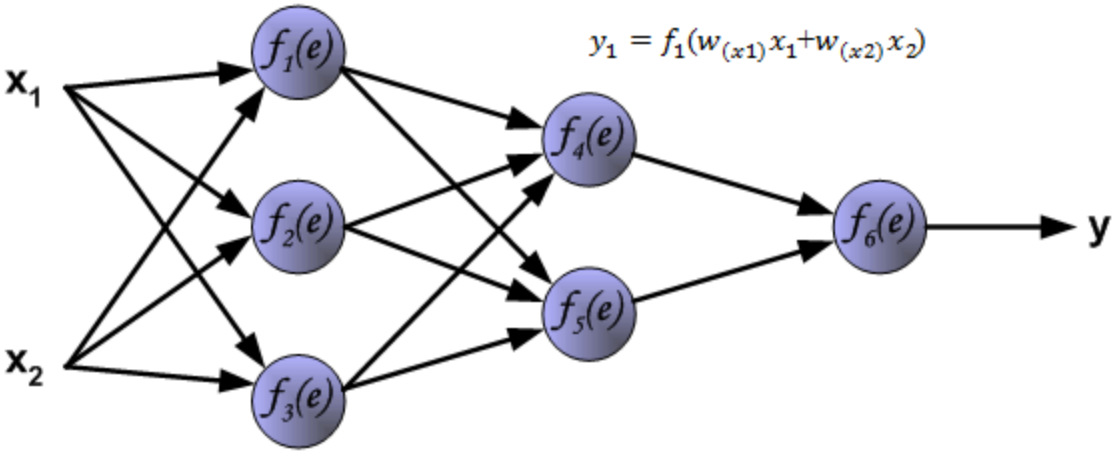
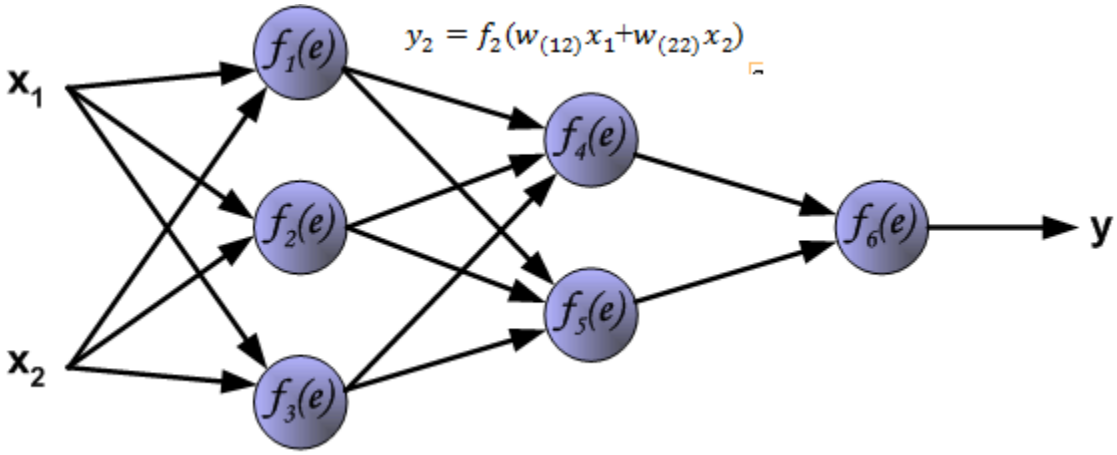


Figure 3.8 Description of neuron as a composition of 2 units

To teach the neural network we need training data set. The training data set consist of input signals (x_1 and x_2) assigned with corresponding target (desired output) z. The network training is an iterative process .In each iteration weight coefficients of nodes and modified using new data from training data set. Modification is calculated using algorithm described below: Each teaching step starts with forcing both input signals from training set. Modification is calculated using algorithm described below: Each teaching step starts with forcing both input signals from training set. After this stage we can determine output signals value for each neuron in each network layer. Picture below illustrate how signal is propagating through the network. Symbols

$w_{(xm)n}$ Represents weights of connections between network input $x_{(m)}$ and neuron n in input layer. Symbols y_n represents output signal of neuron n.


$y_1 = f_1(w_{(x1)}x_1 + w_{(x2)}x_2)$
x1
x2
f1(e)
f2(e)
f3(e)
f4(e)
f5(e)
f6(e)
y


$y_2 = f_2(w_{(12)}x_1 + w_{(22)}x_2)$
x1
x2
f1(e)
f2(e)
f3(e)
f4(e)
f5(e)
f6(e)
y

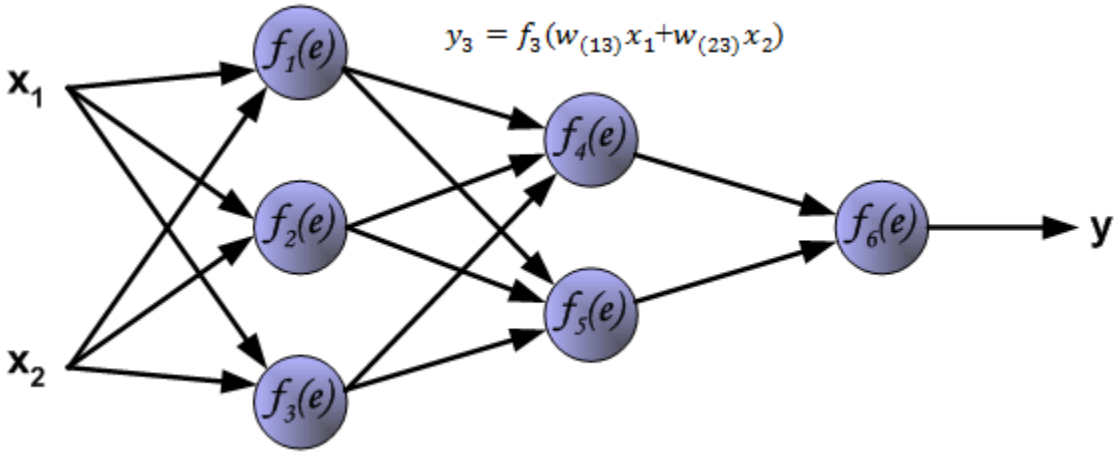

$y_3 = f_3(w_{(13)}x_1 + w_{(23)}x_2)$
x1
x2
f1(e)
f2(e)
f3(e)
f4(e)
f5(e)
f6(e)
y

Figure 3.9 output of 3 neuron in first hidden layer

$y_6 = f_6(w_{(16)}y_1 + w_{(26)}y_2 + w_{(36)}y_3)$

Output of all the 3 neuron in first layer shown in figure3.9. Propagation of signals through the hidden layer.Symbols w_{mn} represent weights of connections between output of neuron m and input of neuron n in the next layer. Output the 2 neurons in second layer are shown in figre 3.10

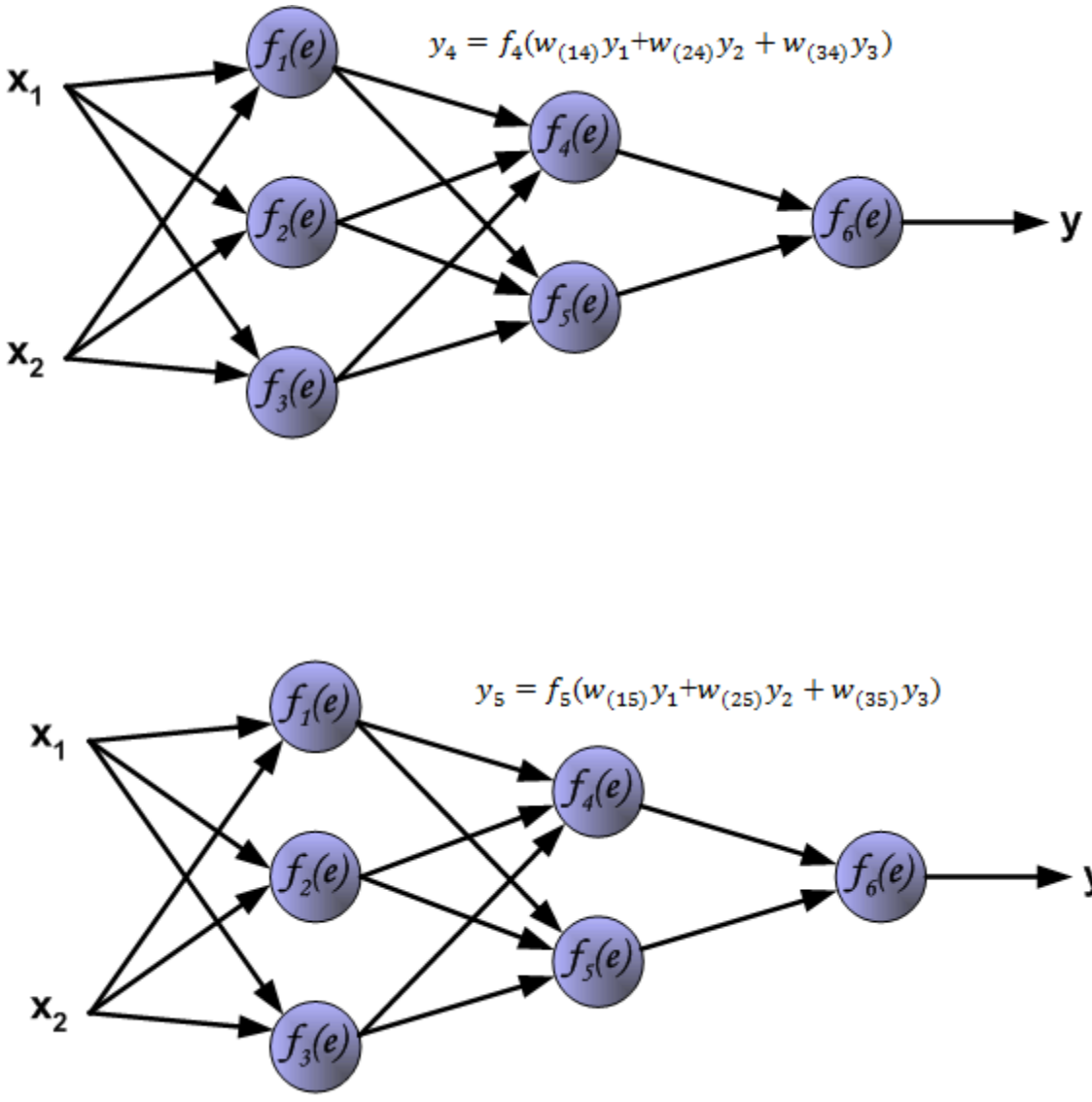


Fig 3.10 output of 3 neuron in the second hidden layer

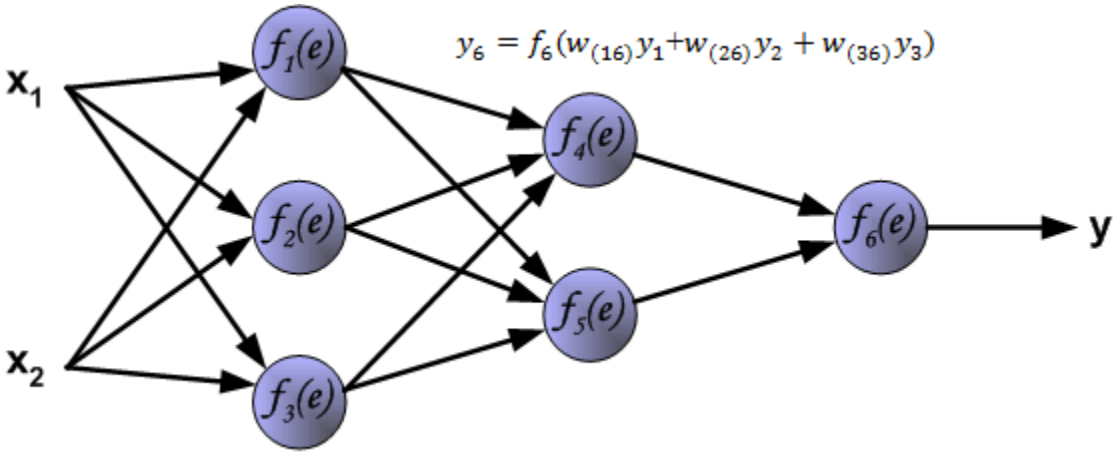


Fig 3.11 output of 3 neuron in the output layer

In the next algorithm step the output signal of the network y is compared with the desired output value (the target) , which is found in training data set.The diffrence is called error signal δ of output layer neuron.

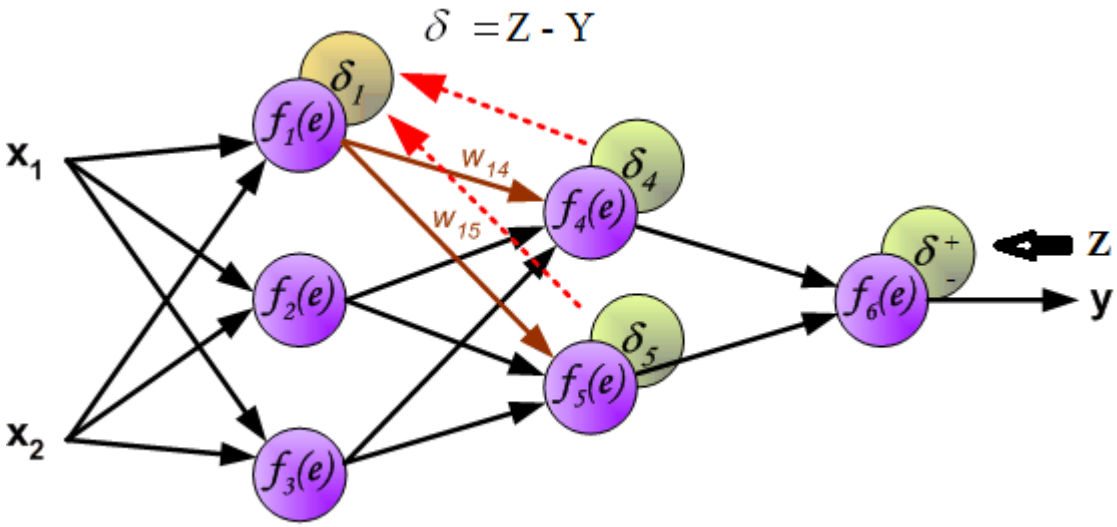


Fig 3.12 findind out the deviation in output of neural network from target value

It is impssible to compute error signal for internal neurons directly ,because output values of these neurons are unknown .For many years the effective methord for training multilayer networks has been unknown .Only the middle eighties the back propagation algorithm has been worked out .The idea is to propagate error signal δ (computed in single teaching step) back to all neurons , which output signals were input for discussed neurons.

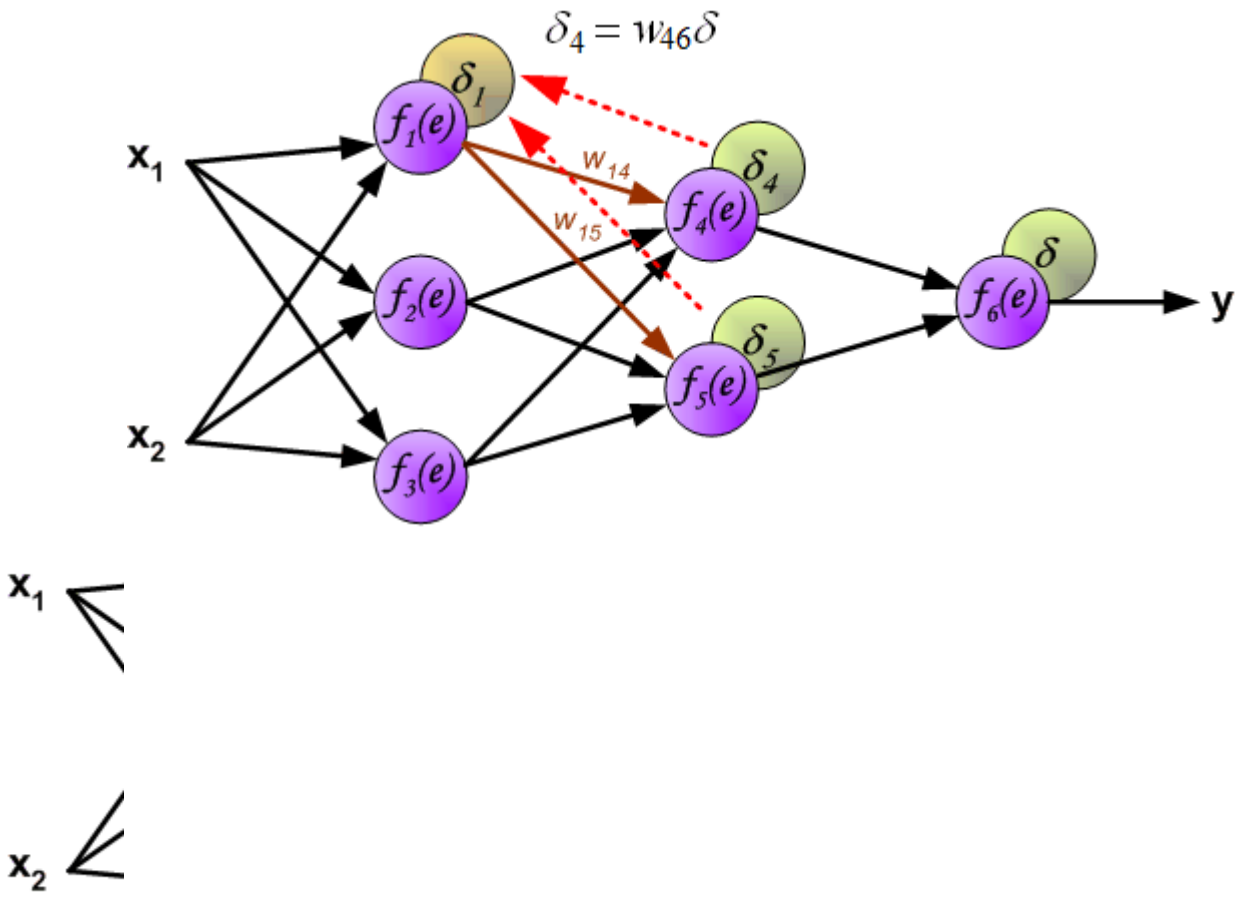
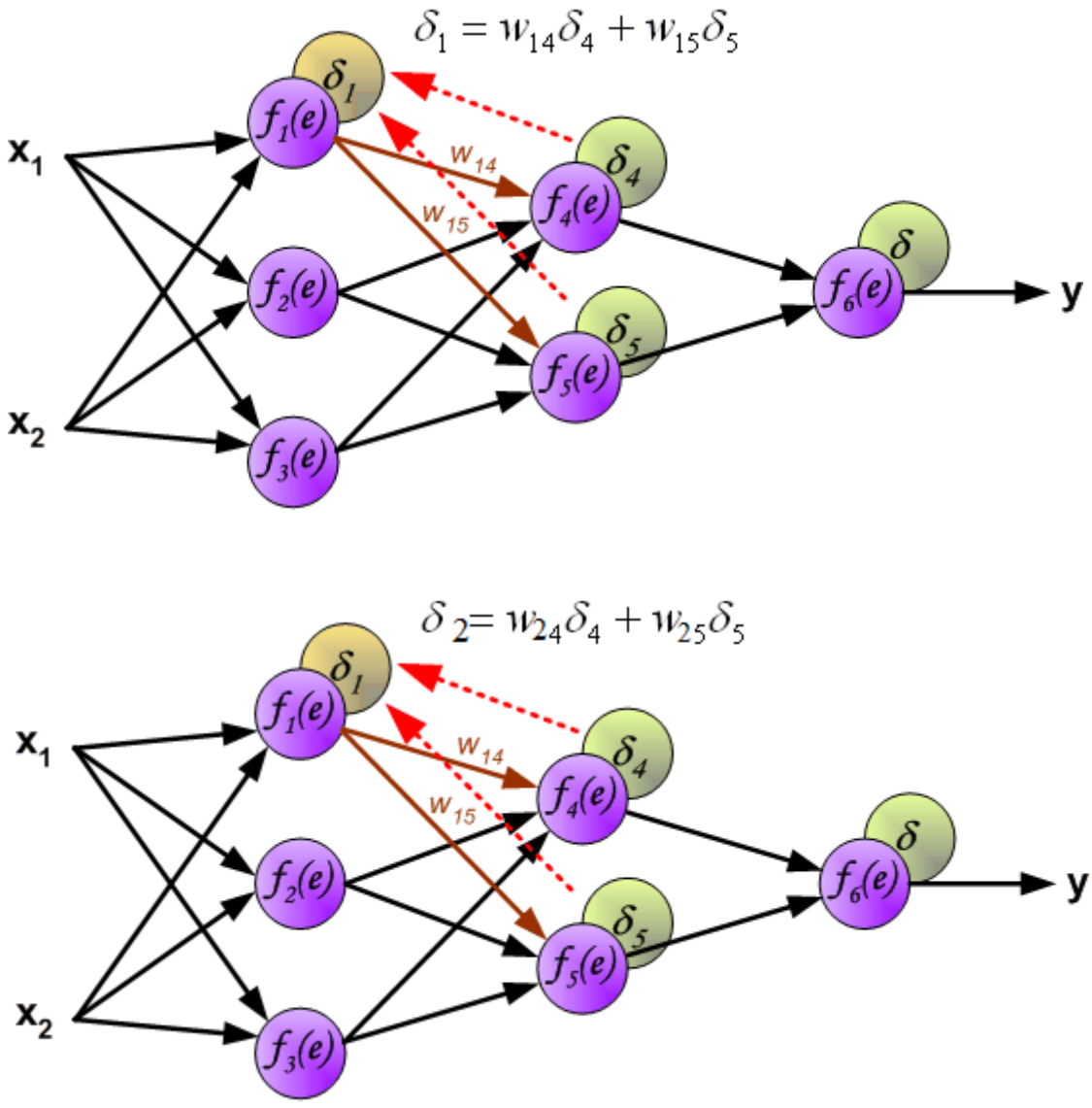


Figure 3.3propagation of error signal from output layer to second hidden layer

The weights' cofficients w_{mn} used to propagate errors back are equal to this used during computing output value. Only the direction of data flow is changed (signal are propagated from output to inputs one after the other). This technique is used for all network layers. If propagated errors came from new neurons they are added. The illustration is below:



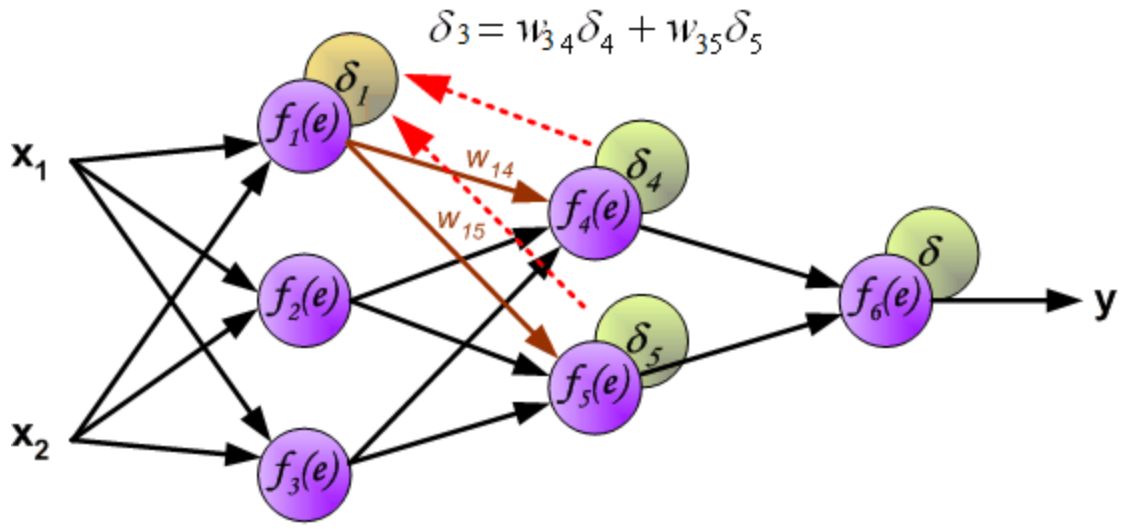
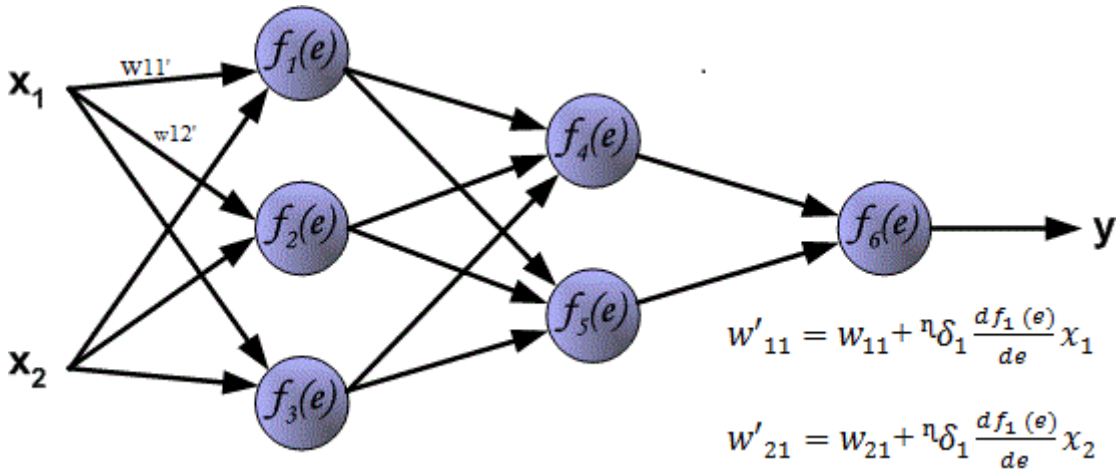
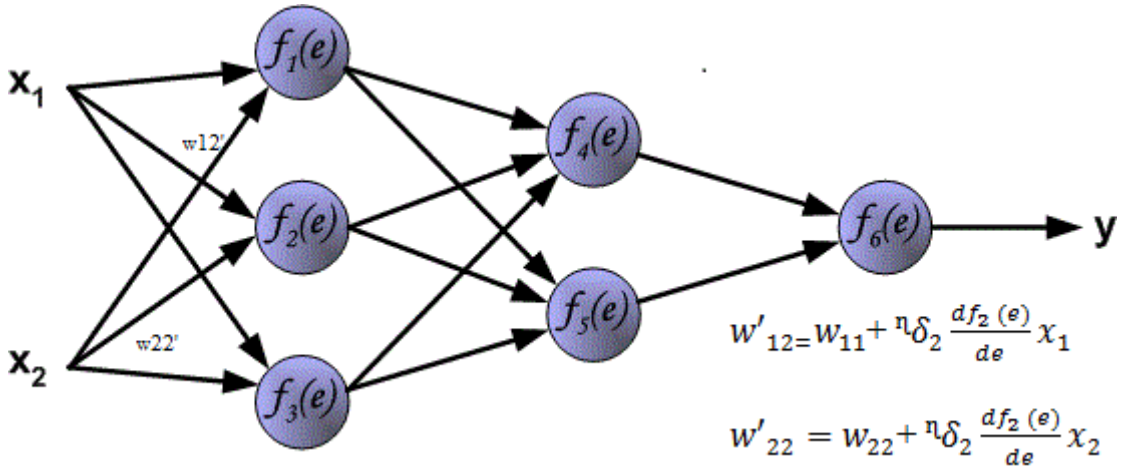


Figure 3.4 propagation of error signal from second layer to first hidden layer

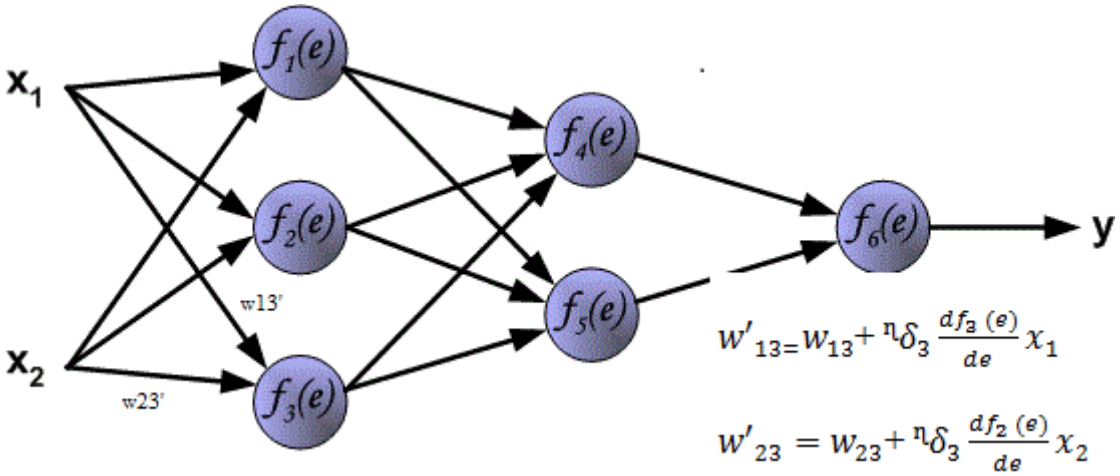
When the error signal for each neuron is computed ,the weights coefficient of each neuron output node input nade may be modified. In formulas beloe $\frac{df(e)}{de}$ represents derivative of neuron activation function (whichn weights are modified).





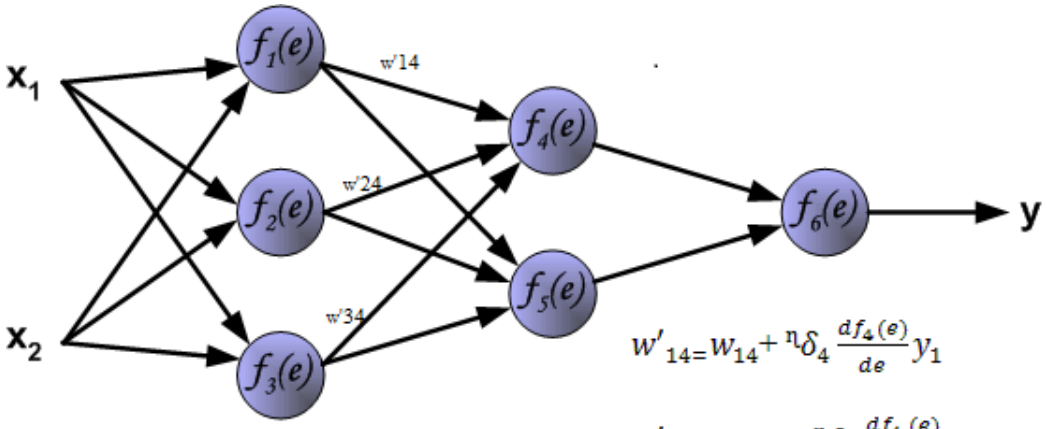
$$w'_{12} = w_{11} + \eta\delta_2 \frac{df_2(e)}{de} x_1$$

$$w'_{22} = w_{22} + \eta\delta_2 \frac{df_2(e)}{de} x_2$$



$$w'_{13} = w_{13} + \eta\delta_3 \frac{df_3(e)}{de} x_1$$

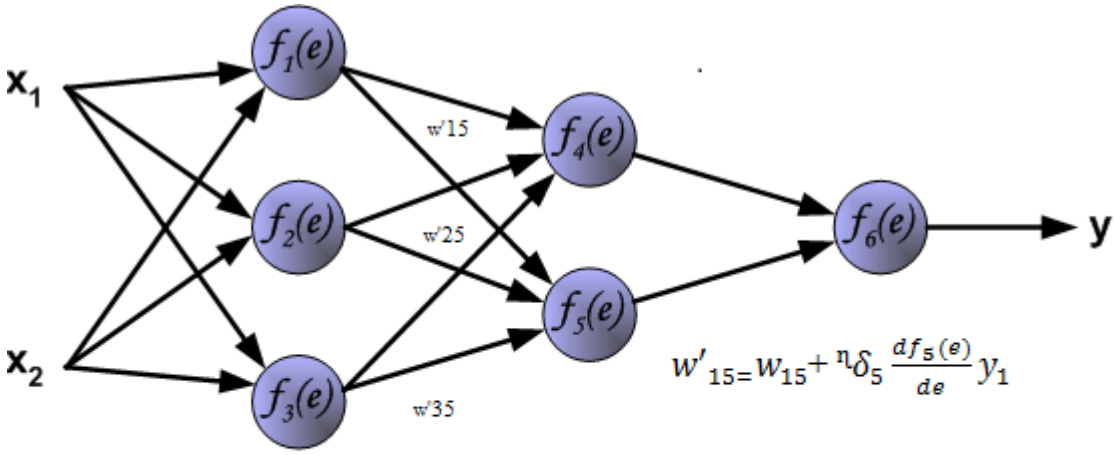
$$w'_{23} = w_{23} + \eta\delta_3 \frac{df_2(e)}{de} x_2$$



$$w'_{14} = w_{14} + \eta\delta_4 \frac{df_4(e)}{de} y_1$$

$$w'_{24} = w_{24} + \eta\delta_4 \frac{df_4(e)}{de} y_2$$

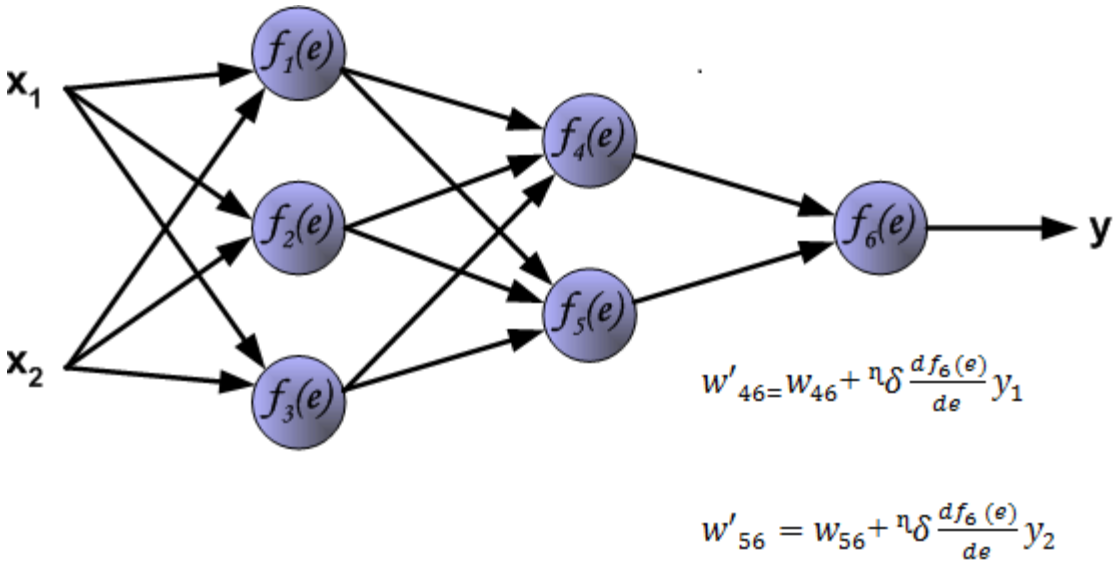
$$w'_{34} = w_{34} + \eta\delta_4 \frac{df_4(e)}{de} y_3$$



$$w'_{15} = w_{15} + \eta\delta_5 \frac{df_5(e)}{de} y_1$$

$$w'_{25} = w_{25} + \eta\delta_5 \frac{df_5(e)}{de} y_2$$

$$w'_{35} = w_{35} + \eta\delta_5 \frac{df_5(e)}{de} y_3$$



Cofficient

affects network teaching speed. There are a few technique to select this parameter . The first method is to start teaching process with large value of the parameter . While weights cofficient are being established the parameter is being decreased gradually. The second, more complicated ,method starts teaching with small parameter value. During the teaching process the parameter is being increased when the teaching is advanced and then decreased again in the final stage .Starting teaching process with low parameter value enables to determine weights cofficients signs.

RESULT AND DISCUSSION

4.1 Implementation in Matlab 7.0.4

The training and testing of the feed forward neural network for distinguishing the linear and nonlinear faults is done using back propagation algorithm in Matlab 7.0.4.The first step in training a feed forward is to create the network object. The function newff creates a feed forward network. Before training a feed forward network , the weight and biases must be initialized .The newff command will automatically initialize the weights.The function sim simulates the networks. Sim takes the network input p, and the network object net, and returns the network output a. Once the network weights and biases have been initialized,the network is ready for

training.The network can be trained for function approximation (nonlinear regression), pattern association ,or pattern classification. In this thesis training is being done for pattern classification. The training process requires a set of examples of proper network behavior – network input p and target output. During training the weights and the biases of the network iteratively adjusted to minimize the network performance function . The default performance function for feed forward is mean square error mse- the average squared error between the network outputs a and the target outputs t.

The implementation of back propagation algorithm updates the network weights and biases in the direction in which the performance function decreases most rapidly- the negative of gradient. One iteration of this algorithm can be written as

$$x_{k+1} = x_k - \alpha_k g_k$$

Where X_k a vector of current weights and biases is, g_k is the current gradient ,and α_k is learning rate.RMS values of third and fifth harmonic components of voltage and current at feeder out at each location at feeder are found out at each location of the fault. Half of the data is obtained like this is used for training the neural network by using back propagation algorithm and the remaining half is used for testing it. In the present case the neural network is trained to give an output value of 1 for linear fault and 0 for non linear fault. We have adapted 3 layer nural network consisting of 4 neuron in each hidden layer and 2 neuron in output layer and the log sigmoid function is used in all layers. Learning rate of 0.565 is used and momentum factor 0f 0.585 is used in the training phase of the neural network. Percentage of testing and training samples classified y different back propagation algorithms of the neural network are shown below in table 4.1.

Table 4.1 Different back propagation algorithm.

S.NO Used	Back Propagation Algorithm
1.	Trainlm
2.	Traingd
3.	Traingda
4.	Traingdx
5.	Traingdm
6.	Trainb
7.	Traibr
8.	Trainrp
9.	Trainbr
10.	Trainfg

REFRENCES

1. B.M Aucoin, B.Don Russell," Distribution High Impedance Fault Detection Utilizing High Frequency Current Component "IEEE Trans. On PAS,Vol.PAS-101,No.6 June 1982.pp.1596-1606.

2.Emanuel,A.E.,Cyganski,D.,Orr,J.A,Gulachenski,E.M.,"High Impedance Fault Arcing on Sandy Soil in 15kv Distribution Feeders:Contribution to the evaluation of the low Frequency Spectrum,"IEEE Transaction On Power Delivery,Volume 5,issue 2,April 1990 pp676-686.

3.Aucoin B,Michel, and Jones,Robert H.,"High impedance Fault Detection Implementation Issues",IEEE Transaction on power Delivery,Vol.11,No.1 January,1996,pp.139-148.

4. A.M Sharaf, L.A. Snider, K.Debnath," A Third Harmonic Sequence ANN Based Detection Scheme For High Impedance Faults",Proceedings of the ISEDEM Singapore ,pp.802-806.

5.L.A Snider ,Yuen Yee Shan,"The Artificial Neural Networks based Relay Algorithm Distribution System High Impedance Fault Detection",Proceedings of the fourth International Conference on Advances in Power Sytem Control ,Operation and Management,APSCOM-97,Hong Kong,November 1997,pp 100-106.

6. L.A. Snider,Yuen yee Shan" The Artificial Neural Networks based Relay Algorithm For the Detection Of High Impedance Faults",ELSEVIER Transactions on Neuro Computing ,1988,pp 243-254

7. T.M. Li Snider ,L.A. Snider. Lo,C.H. Cheung and K.W.Chan "High Impedance fault detection Using Artificial Neural Network", Proceedings of the fourth International Conference on Advances in power System Control , Operation and Management,APSCOM ,Hong Kong, November 2003,pp 821-826

8. A.M Sharaf ,L.A Snider,K,Debnath ," A Neural Network based Back error Propagation Relay Algorithm for Distribution System High Impedance fault Detection", Proceeding the ISEDEM. Singapore,pp.613-20

9.Adel M.Sharaf,Guosheng Wang,"High Impedance Fault Detection using Low Order Pattern Harmonic Detection",IEEE TRANS.pp 883-886.

10.Adel M.Sharaf,Guosheng Wang,"High Impedance Fault Detection using Feature Pattern Based Relaying"IEEE trans.pp 222-226.

11. A.M,R M. EI-Sharkawy,H.E.A,Talaat,M.A.L.Badr"Novel Alpha-Transformation Distance Relaying Scheme".IEEE trans.pp 754-757.

12. A.M.Sharaf, L.A. Snider, K.Debnath," A Neural Network Based relaying Scheme For Distribution System High Impedance Fault Detection ",IEEE trans.pp 321-326

13.T.M Lai Snider,L.A.Snder.E.Lo"Wavelet transform based algorithm for the detection of stochastic high impedance faults",ELSEVIER Transaction on Electric Power System Research,pp 626-633.

14. Abhishek Bansal and G.N.pillai "High Impedance Fault Detection Using Artificial Neural Networks",trans.pp 148-152

15. M.M,Eissa (SMIEEE) G.MA. Sowilam,A.M Sharaf (SMIEEE)" Anew Protection Detection Technique For High Impedance Fault Using Neural Network" IEEE trans .pp 146-151.

16. Howard Demuth ,Mark Beale Artificial Neural Network Matlab User guide.

17.Back Propagation Learning Algorithm ,www.wikipedia.org.

18. A.A . Girgas, W. Chang, and E.B.Makram,"Analysis of high impedance fault generated signals using a Kalman Filtering Approach,"IEEE Trans.Power Deliv.5(October(4))(1990) pp.1714-1724.