A Study of Various Training Algorithms on Neural Network for Angle based Triangular Problem

Amarpal Singh
M.Tech (CS&E)
Amity University
Noida, India

Piyush Saxena
M.Tech (CS&E)
Amity University
Noida, India

Sangeeta Lalwani
M.Tech (CS&E)
Amity University
Noida, India

ABSTRACT
This paper examines the study of various feed forward back-propagation neural network training algorithms and performance of different radial basis function neural network for angle based triangular problem. The training algorithms in feed forward back-propagation neural network comprise of Scale Gradient Conjugate Back-Propagation (BP), Conjugate Gradient BP through Polak-Riebre updates, Conjugate Gradient BP through Fletcher-Reeves updates, One Secant BP and Resilent BP. The final result of each training algorithm for angle based triangular problem will also be discussed and compared.

General Terms
Artificial Neural Network (ANN), Feed Forward Backpropogation (FFB), Training Algorithm.

Keywords
Feed-forward back-propagation neural network, radial basis function, generalized regression neural network.

1. INTRODUCTION
Artificial Neural Network (ANN) learns by adjusting the weights so as to be able to correctly categorize the training data and hence, after testing phase, to classify unknown data. It needs long time for training. It has a high tolerance to noisy and incomplete data.

Some Salient Features of ANN are as follow:

- Adaptive learning, Self-organization
- Real-time operation, Massive parallelism
- Error acceptance by means of redundant information coding
- Learning and generalizing ability
- Distributed representation

There are many different definitions of neural networks that are quoted by some famous researcher as follow:

DARPA Neural Network (NN) Study:

A NN is a taxonomy collected of various straightforward dispensations of fundamentals in commission of parallel whose purpose is unwavering by network configuration, connection strengths, as well as the dispensation executed at computing fundamentals or nodes.

According to Haykin (1994):

A Neural Network is a vast analogous scattered processor which has an expected tendency for storing practical knowledge and making it available for use. It is similar to the brain in two aspects [1]:

- Knowledgebase is acquired by the network through a training process.
- Inter-neuron connection strengths known as synaptic weights are used to accumulate the knowledge.

According to Nigrin (1993):

- A NN is a circuit composed of a very large number of simple processing elements that are neurally based. Each element operates only on local information.

According to Zurada (1992):

- Artificial neural systems, or neural networks, are physical cellular systems which can acquire, store and utilize experiential knowledge.

This paper examines how the artificial neural network is implemented for angle based triangle problem. The performance (speed processing and high accuracy result) of training algorithm that been used in this problem is the research target. The following figure (Fig. 1) shows the system architecture for angle based triangular problem.

System structural design

System structural design starts with creating training database. After that neural network model such as training function, design and constraint were initialized.

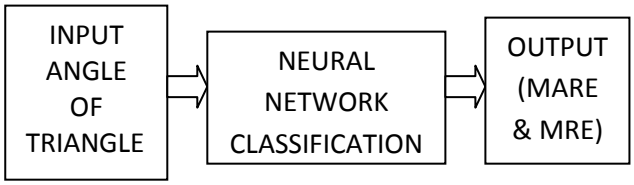


Fig 1: Block Diagram for Angle Based Triangular Problem

Fig 2: Block diagram for neural network

2. LITERATURE REVIEWED

In this section of paper, the various types of neural networks are discussed.

Basic Models of Artificial Neural Networks:

- **Single-Layer Feed-Forward Network:**

When a layer of the processing nodes is formed, the inputs can be connected to these nodes with various weights, resulting in a series of outputs, one per node.

- **Multilayer Feed-Forward Network:**

A Multilayer feed-forward network is formed by the interconnection of several layers. The input layer is that which receives the input and this layer has no function except buffering the input signal.

- **Single Node with its own Feedback:**

Single node with its own feedback is simple recurrent neural network having a single neuron with feedback itself.

- **Single-Layer Recurrent Network:**

Single-layer recurrent network with a feedback connection in which a processing element's output can be directed back to the processing element itself or the other processing element or to both.

- **Multilayer Recurrent Network:**

In Multilayer recurrent network, a processing element output can be directed back to the nodes in a preceding layer, forming a Multilayer recurrent network:

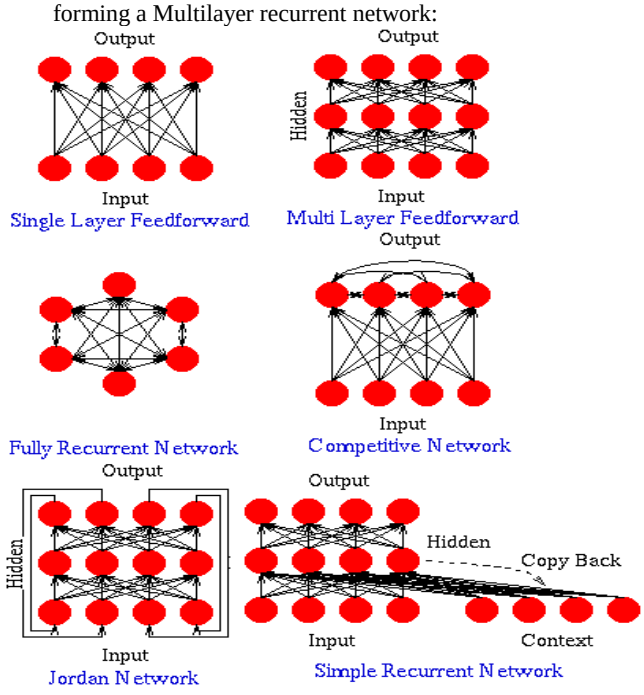


Fig 3: Basic Models of ANN

Feed-forward Back-Propagation (FFBP) Neural Network

This neural network was trained and validated for various feed-forward backprop training algorithms available in Matlab Neural Network toolbox [2].

SUPPORTED TRAINING FUNCTIONS IN FFBP NEURAL NETWORK [10]

There are many supported training functions as follow : **Trainb (**Batch training with weight and bias learning rules) , **Trainbfg (**BFGS quasi-Newton BP), **Trainbr** (Bayesian regularization) , **Trainc** (Cyclical order incremental update) , **Traincgb** (Powell-Beale conjugate gradient BP), **Traincgf** (Fletcher-Powell conjugate gradient BP), **Traincgp** (Polak-Ribiere conjugate gradient BP), **Traingd (**Gradient descent BP), **Traingda (**Gradient descent with adaptive learning rate BP), **Traingdm** (Gradient descent with momentum BP), **Traingdx** (Gradient descent with momentum & adaptive linear BP), **Trainlm** (Levenberg-Marquardt BP), **Trainoss** (One step secant BP), **Trainr** (Random order incremental update), **Trainrp** (Resilient backpropagation (Rprop)), **Trains** (Sequential order incremental update), **Trainscg** (Scaled conjugate gradient BP)

SUPPORTED LEARNING FUNCTIONS IN FFBP NEURAL NETWORK [10]

There are many supported learning functions as follow : **learncon (**Conscience bias learning), **learngd** (Gradient descent weight/bias learning), **learngdm (**Gradient descent with momentum weight/bias learning), **learnh (**Hebb weight learning), **learnhd (**Hebb with decay weight learning rule), **learnis (**Instar weight learning), **learnk (**Kohonen weight learning), **learnlv1 (**LVQ1 weight learning), **learnlv2 (**LVQ2 weight learning), **learnos (**Outstar weight learning), **learnp (**Perceptron weight and bias learning) **learnpn (**Normalized perceptron weight and bias learning), **learnsom (**Self-organizing map weight learning), **learnwh (**Widrow-Hoff weight and bias learning rule).

TRANSFER FUNCTIONS IN FFBP NEURAL NETWORK [10]

There are many transfer functions as follow : **compet** (Competitive), **hardlim** (Hard limit transfer), **hardlims** (Symmetric hard limit), **logsig** (Log sigmoid), **poslin** (Positive linear),**purelin** (Linear), **radbas** (Radial basis),**satlin** (Saturating linear)**satlins** (Symmetric saturating linear), **tansig** (Hyperbolic tangent sigmoid), **tribas** (Triangular basis)

TRANSFER DERIVATIVE FUNCTIONS [10]

There are many transfer derivative functions as follow : **dhardlim** (Hard limit transfer derivative), **dhardlms** (Symmetric hard limit transfer derivative), **dlogsig** (Log sigmoid transfer derivative), **dposlin** (Positive linear transfer derivative), **dpurelin (**Hard limit transfer derivative), **dradbas** (Radial basis transfer derivative), **dsatlin (**Saturating linear transfer derivative), **dsatlins** (Symmetric saturating linear transfer derivative), **dtansig** (Hyperbolic tangent sigmoid transfer derivative), **dtribas** (Triangular basis transfer derivative).

WEIGHT & BIAS INITIALIZATION FUNCTIONS [10]

There are many weight and bias initialization functions as follow: **initcon** (Conscience bias initialization), **initzero** (Zero weight/bias initialization), **midpoint** (Midpoint weight initialization), **randnc** (Normalized column weight initialization), **randnr** (Normalized row weight initialization), **rands** (Symmetric random weight/bias initialization)

WEIGHT DERIVATIVE FUNCTIONS [10]

The weight derivative function in ANN is **Ddotprod (**Dot product weight derivative function).

In this paper, fifteen training algorithms and two learning function namely "learngd" and "learngdm" are used.

RBF Neural Network

RBF's are embedded in a two layer neural network where each hidden unit implements a radial activated function. The output units implement a weighted sum of hidden unit outputs. The input into an RBF network is non linear while the output is linear. In order to use RBF network it need to specify the hidden unit activation function, the number of processing units, a criterion for modeling a given task and a training algorithm for finding the parameters of the network. After training the RBF network can be used with data whose underlying statistics is comparable to that of the training set. RBF networks have been successfully applied to a large diversity of applications including interpolation, time series modeling, speech recognition etc.

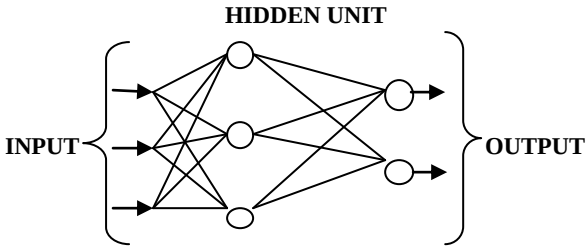


Fig 4: Network topology of RBF

GRNN Generalized Regression Neural Network

GRNN is consists of a RBF layer with a unique linear layer used for function approximation with adequate amount of unseen neurons. The MATLAB Neural Network Toolbox Function (newgrnn) has been used for testing and training the network performance via measure of GRNN for corresponding to validation data.

Fig 5 shows architecture of GRNN. It is comparable to the RBF network, but it has a somewhat dissimilar second layer.

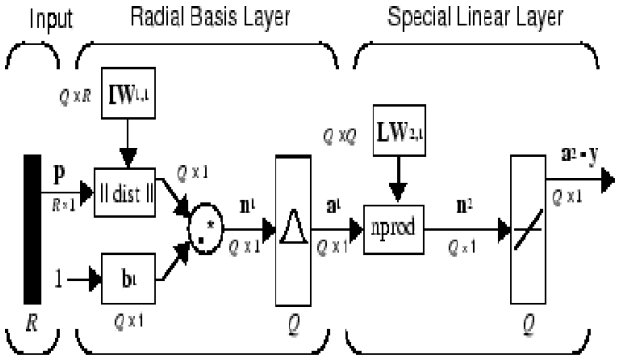


Fig. 5: GRNN Network Topology

Some researcher use these neural network fundamental to propose their own metrics regarding to their research areas. **Khoshgoftarr et al.** [3] introduced to apply the concept of the NN as a tool for predicting software quality. They presented a discriminated model and a NN representation of the large telecommunications system, classifying modules as not fault-prone or fault-prone. They compared the neural-network model with a non parametric disciminant model, and found the neural network model had better predictive accuracy. **Specht** [4] has stated that it is a memory-based network that provides estimates of continuous variables and converges to the underlying (linear or nonlinear) regression surface. This is a one-pass learning algorithm with a highly parallel structure. Even with sparse data is a multidimensional measurement space; the algorithm provides smooth transitions from one observed value to another.

3. RESEARCH METHODOLOGY

In this section of paper, the research methodology for the implementation of the problem is provided.

Feed-forward Back-propagation Neural Networks

Backprop implements a gradient descent search through a space of possible network weight, iteratively reducing the error E, between training example and target value and network output. It guaranteed to converge only towards some local minima. A training procedure which allows multilayer feed forward Neural Networks to be trained.

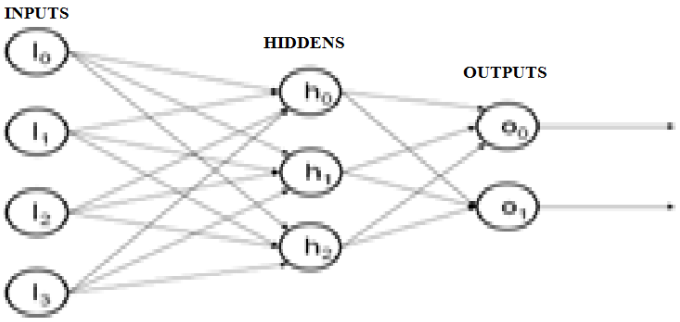


Fig 6: Architecture of Feed Forward Network

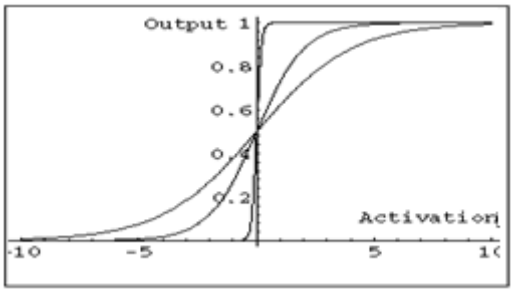
However, the major disadvantages of BP are its convergence rate relatively slow [11] and being trapped at the local minima.
many powerful optimization algorithms have been devised, most of which have been based on simple gradient descent algorithm as explain by C.M. Bishop [12] such as conjugate gradient decent, scaled conjugate gradient descent, quasi-Newton BFGS and Levenberg-Marquardt methods.

For feed forward networks:

A continuous function can be

- differentiated allowing
- Gradient-descent.
- Back propagation is an example of a gradient-descent technique.

Uses sigmoid (binary or bipolar) activation function.



In multilayer networks, the activation function is usually more complex than just a threshold function, like 1/[1+exp(-x)] or even 2/[1+exp(-x)] – 1 to allow for inhibition, etc.

Gradient Descent

- Gradient-Descent(training_examples, η)
- Each training example is a pair of the form $<(x_1,\ldots x_n),t>$ where $(x_1,\ldots,x_n)$ is the vector of input values, and t is the target output value, η is the learning rate (e.g. 0.1)
- Initialize each wi to some small random value
- Until the termination condition is met, Do
    - Initialize each Δwi to zero
    - For each $<(x_1,\ldots x_n),t>$ in training_examples Do
    - Input the instance (x1,…,xn) to the linear unit and compute the output o
    - For each linear unit weight wi Do
- $\Delta w_i = \Delta w_i + \eta\,(t-o)\,x_i$
- For each linear unit weight wi Do
- $w_i = w_i + \Delta w_i$

Sigmoid Activation Function

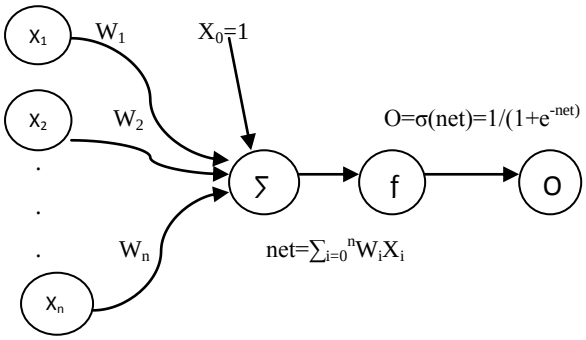


Fig. 7: Sigmoid Activation Function

Derive gradient decent rules to train:

- one sigmoid function
  $\partial E/\partial w_i = -\Sigma d(td-od)\ od\ (1-od)\ xi$
- Multilayer networks of sigmoid units backpropagation

Conjugate gradient

This is the well accepted iterative technique for solving huge systems of linear equations [6]. In the 1st iteration, the conjugate gradient algorithm will find the steep descent direction.

Description of 3 types of conjugate Gradient Algorithms:-

- Scaled Gradient Conjugate Backpropogation(SCG),
- Conjugate Gradient BP with Polak-Riebre Updates(CGP) and
- Conjugate Gradient BP with Fletcher-Reeves updates(CGF).

Approximate solution, x_k for conjugate gradient iteration is described as formulas below [7]:

$$X_k = X_{k-1} + \alpha_k d_{k-1}$$

Scaled Gradient Conjugate Backpropogation (SCG)

SCG calculate the second order Conjugate Gradient Algorithm that will help to reduce goal functions for some variables. Moller [7] proved this theoretical foundation in which remain its first order techniques in first derivatives like standard backpropogation. This helps to find way for local minimum in second order techniques in second derivatives.

Conjugate Gradient Backpropagation with Fletcher-Reeves Updates (CGF)

The 2nd edition for Conjugate Gradient algorithm was projected by Fletcher-Reeves. As with the Polak and Ribiére algorithm, the explore path at per iteration is computed by equation below.

$$\beta_k = \frac{g_k^T \ g_k}{g_{k-1}^T g_{k-1}}$$

Conjugate Gradient Backpropagation with Polak-Riebre Updates (CGP)

One more edition of the Conjugate Gradient algorithm was projected by Polak along with Ribiére. The explore path at per iteration is same like SCG search direction equation. However for the Polak-Ribiére update, the constant beta, βk is computed by equation below.

$$\beta_k = \frac{\Delta g_{k-1}^T \ g_k}{g_{k-1}^T g_{k-1}}$$

Quasi-Newton Algorithms (One-Step Secant Backpropagation (OSS))

An alternative way to speed up the conjugate gradient optimization is Newton's method. The fundamental footstep of Newton's technique shows in equation following.

$$x_{k-1} = x_k - A_k^{-1} g_k$$

Heuristics Algorithms (Resilent Backpropagation(RP))

The reason for resilient backpropagation training algorithm is to get rid of these destructive sound effects of the magnitudes of the fractional derivatives [5].

Performance evaluation Using Feed-Forward Back-propagation Neural Networks [8]:

The first examination was to contrast the predictive accuracy of Feed – Forward Neural Network trained with various back-propagation training algorithms. This neural network was trained and validated for various feedforward backprop training algorithms available in Matlab Neural Network toolbox [2].

The predictive accuracy of training algorithms was compared:

Mean Absolute relative error (MARE) [9]: This is the preferred measure used by software engineering researchers and is given as

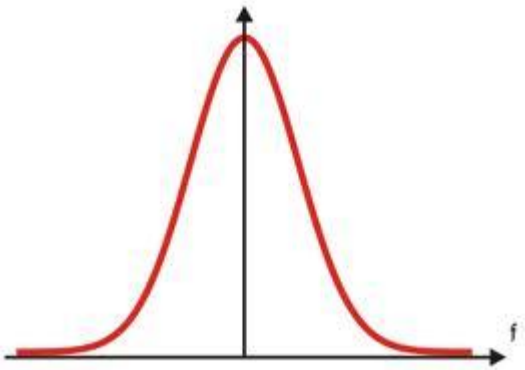
$$MARE = \left(\sum_{i=1}^{n} \left| \frac{estimate - actual}{actual} \right| \right) \% \, n$$

Mean Relative Error (MRE) [9]: This measure is used to estimate whether models are biased and tend to overestimate or underestimate and is designed as follows

$$MRE = \left(\sum_{i=1}^{n} \left| \frac{estimate - actual}{n} \right| \right) \% \, n$$

Radial Basis Function (RBF) Neural Network

The RBF is a classification and functional approximation neural network developed by M.J.D. Powell. The network uses the most common nonlinearities such as sigmoidal and Gaussian kernel functions. The Gaussian functions are also used in regularization networks. The Gaussian function is generally defined as



$$f(y) = e^{-y^2}$$

Fig 8: Gaussian function

Performance evaluation using Radial Basis Function (RBF) Neural Networks [8]:

The second investigation was to construct performance prediction models and compare their predictive accuracy using different radial basis function neural networks available in Matlab Neural Network toolbox [2]. Three radial basis functions are available in the toolbox. They are

i. **Exact design radial basis networks**
ii. **Efficient design radial basis networks**
iii. **Generalized Regression Neural Network**

4. PROBLEM STATMENT

This part of paper will explore the problem statement for the implementation of neural network in feed-forward backpropogation and radial basis function. In this problem triangle is identified based on their angle input. In this problem different types of angle based triangle is identified using radial basis function neural networks and feed-forward back-propagation neural network.

For neural networks the **training data, test data and target data is given to as input**. Input is given in form of **matrices**. For this problem the training data is given as learning set for the network. The learning data can be changed according to different problem statement.

5. IMPLIMENTATION

Feed-forward backpropogation NN using Matlab

On **Training Info**, choose Inputs and Targets.

lying on **Training Parameters**, specify:

epochs = 1000 (as the learning would be better when there are large no. of epochs and long durations of training)

goal = 0.000000000000001

max_fail = 50

This will give you a learning and performance graph.

Once graph is decomposed (since you are trying to minimize the error) similar to that in figure 9.

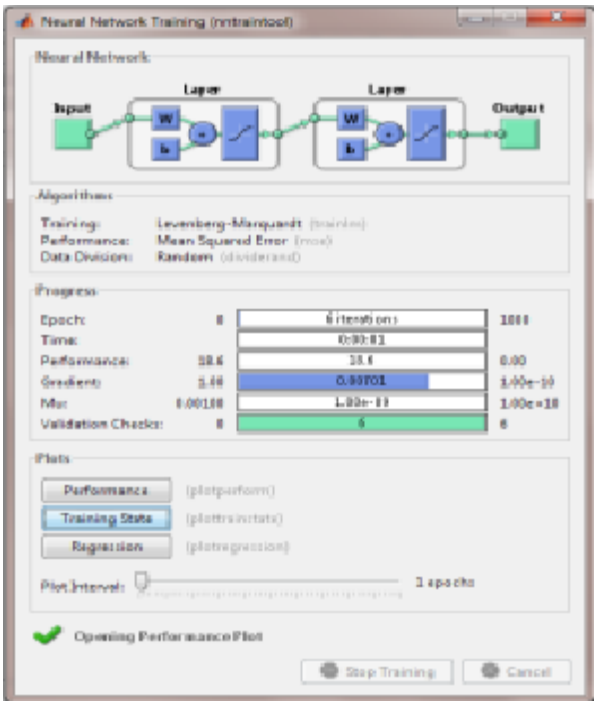


Fig 9: Train Network

Radial basis function Neural Networks using Matlab

Make sure the parameters are as follows:

Network Type = radial basis function (exact fit)
Spread constant = 1.0

Network Training:

on clicking the **Create** button, the network is automatically trained which concludes the network implementation phrase.

Network Simulation:

Now, test the test_data **S** on the NN Network Manager and track the identical method indicated before (like for input P).

Now perform the same steps with **radial basis function (fewer neurons) neural network** and **Generalized Regression Neural Network (GRNN).**

6. OBSERVATIONS AND RESULTS

The Observations through Different Training Functions is formulated in form of table. Here all training algorithms with "LEARNGDM" Adaption Learning Function, Performance Function is "MSE" and Numbers of Layers = 2 are used :

- No. of neurons= 10
- Transfer function=tansig

Training function	No. of iterations	MARE	MRE
TRAINBFG	1	0.36	-0.192
TRAINBR	6	1.3	-0.21
TRAINCGB	2	0.07	0.09
TRAINCGF	3	0.14	-0.24
TRAINGD	7	1.03	0.16
TRAINGDM	8	35.7	-0.06
TRAINGDA	7	0.07	0.12
TRAINGDX	6	8.5	0.34
TRAINLM	1000	39.3	0.18
TRAINOSS	2	0.58	0.15
TRAINRP	6	0.3	-0.29
TRAINR	1000	0.36	-0.12
TRAINSCG	206	67.2	0.091
TRAINCGP	3	8.59	-0.4

TABLE 1: Results of error Prediction with different training function

Adaption Learning Function = LEARNGD

- No. of neurons= 10
- Transfer function= pure liner

Training function	No. of iterations	MARE	MRE
TRAINBFG	2	0.0018	0.2
TRAINBR	7	0.27	0.005
TRAINCGB	27	48.05	0.028
TRAINCGF	5	55.9	88.7
TRAINGD	68	0.38	0.14
TRAINGDM	6	1.18	0.14
TRAINGDA	6	8.3	0.42
TRAINGDX	6	3.1	0.14
TRAINLM	1000	416.4	0.28
TRAINOSS	3	1.0	-0.38
TRAINRP	8	0.26	-0.15
TRAINR	1000	0.14	-0.2
TRAINSCG	22	8.0	0.27
TRAINCGP	4	124.89	0.15

TABLE 2: Performance of training functions after change in parameters.

RBF Network	MARE	MRE
Generalized Regression	0.18	0.27
RBF (Exact Fit)	0.5	-0.33
RBF(Fewer neurons)	0.49	-0.14

TABLE 3: Performance with different RBF Networks

7. CONCLUSION

In this paper, the performance of RBF network is observed, the simulated neural network for the prediction accuracy using radical basis function neural network. The simulated network for prediction of errors by changing any metric can analyze the effect of change in that metric. MARE and MRE value can be predicted by increasing or decreasing different metric. This concludes that radial basis function (exact fit) has better accuracy than any other radial basis network. This work can be applied to predict MARE and MRE by using feed-forward backpropogation neural network. Using feed-forward backpropogation neural network, it concludes that the performance of TRAINCGB (Powell-Beale conjugate gradient back-propagation) training function predicts better accuracy than any other training functions. Several different algorithms for training the networks were evaluated : classical backpropagation with variable learning rates, scale gradient conjugate, conjugate gradient with Polak- Riebre Updates, conjugate gradient with Fletcher-Reeves updates, one secant and resilent were tested as well, but were found to require too much RAM memory to allow PC compatible computers to be used efficiently. All tested Artificial Neural Network (ANN) were feedforward networks in which the neurons were processed by a hidden layer of units, which used the so-called tan-sigmoid output function [13] that fed into a linear output layer that predicted the errors. The conjugate gradient algorithms, in mainly trainscg, appear to execute fine more than a extensive diversity of problems, mainly for networks with a huge no. of weights. The SCG algorithm is approximately as quick as the LM algorithm on function approximation problems (more rapidly for large networks) plus is approximately as quick as RP on pattern recognition problems. The performance of SCG algorithm does not degrade as quickly as RP performance does when the error is minimized. The conjugate gradient algorithms have comparatively humble memory necessities [5].

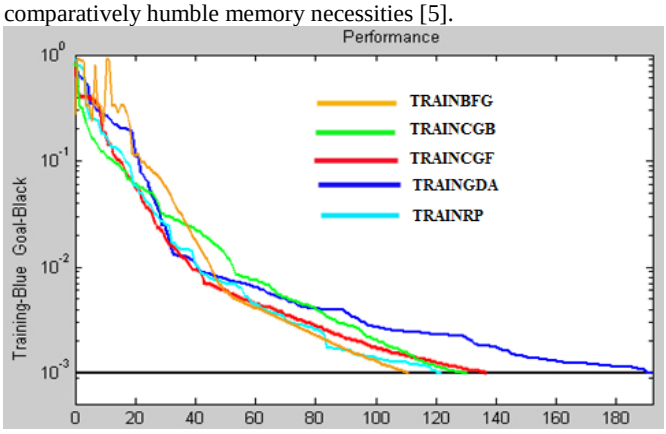


Fig 10: Graphical comparison of some training functions

8. FUTURE SCOPE

In the future work, The RP function is the fastest algorithm and can be used for wide varieties of problems such as pattern recognition. The main drawback of this algorithm is as it does not perform well on function approximation problems and performance also degrades as the error goal is reduced. The memory necessities for this algorithm are comparatively less in contrast with other algorithms considered. In this work neural networks (based on neural networks algorithms), further can design by own training algorithm for better results focusing on accuracy, those may be based on Fuzzy Logic algorithms, Artificial Intelligence based algorithms or Support vector based algorithms etc.

9. REFERENCES

[1] S. Haykins, "A Comprehensive Foundation on Neural Networks," Prentice Hall, 1999 .

[2] Neural network specification, available at URL: http://www.mathworks.in/products/neuralnetwork/description3.html

[3] T.M. Khoshgoftaar, E.B. Allen, J.P. Hudepohl, and S.J. Aud, "Application of neural networks to software quality modeling of a very large telecommunications system", IEEE Transactions on Neural Network, vol. 8, pp. 902-909, 1997.

[4] D.F, Specht, "A general regression neural network".IEEE Transactions on Neural Networks, vol. 2, Issue: 6, pp. 568-576, 1991.

[5] Mathworks Online. Available: http://www. www.mathworks.com, 2009.

[6] Jonathan Richard Shewchuk, an Introduction to the Conjugate Gradient Method without the Agonizing Pain, August 1994.

[7] Martin Fodslette Moller, A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks, 6:525-533, 1993.

[8] S.N. Sivanadam, S. Sumathi, S.N. Deepa ,"Introduction to Neural Networks Using Matlab 6.0 " , Tata McGraw Hill , NewDelhi, 2006.

[9] G.Finnie and G. witting, "AI tools for software Devlopment Effort Estimation", International Conference on Software Engineering: Education and practice, 1996.

[10] S.N. Sivanandam & S.N. Deepa, Principles of Soft Computing, Wiley Publications

[11] Zweiri, Y.H., Whidborne, J.F. and Sceviratne, L.D. A Three-term Backpropagation Algorithm. Neurocomputing. 50: 305-318. 2002.

[12] 10. C.M. Bishop. Neural Networks for Pattern Recognition. Chapter 7, pp.253-294. Oxford University Press. 1995.

[13] Demuth, H., Beale, M. Neural Network Toolbox (Mathworks, Natick,Massachusetts, 1998). World Academy of Science, Engineering and Technology