ISSN 2229-5518

Load Forecasting Using New Error Measures In Neural Networks

Devesh Pratap Singh, Mohd. Shiblee, Deepak Kumar Singh

Abstract— Load forecasting plays a key role in helping an electric utility to make important decisions on power, load switching, voltage control, network reconfiguration, and infrastructure development. It enhances the energy-efficient and reliable operation of a power system. This paper presents a study of short-term load forecasting using new error metrices for Artificial Neural Networks (ANNs) and applied it on the England ISO.

Index Terms— Load forecasting, Neural networks, New neuron Models, New error metrics for neural networks, England ISO

1 Introduction

There is a growing tendency towards unbundling the electricity system. This is continually confronting the different sectors of the industry (generation, transmission, and distribution) with increasing demand on planning management and operations of the network. The operation and planning of a power utility company requires an adequate model for electric power load forecasting. Load forecasting plays a key role in helping an electric utility to make important decisions on power, load switching, voltage control, network reconfiguration, and infrastructure development.

Methodologies of load forecasts can be divided into various categories that include short-term forecasts, medium-term forecasts, and long-term forecasts. Short-term forecasting which forms the focus of this paper, gives a forecast of electric load one hour ahead of time. Such forecast can help to make decisions aimed at preventing imbalance in the power generation and load demand, thus leading to greater network reliability and power quality.

Many methods have been used for load forecasting in the past. These include statistical methods such as regression and similar-day approach, fuzzy logic, expert systems, support vector machines, econometric models, end-use models, etc. [2].

A supervised artificial neural network has been used in this work. Here, the neural network is trained on input data as well as the associated target values. The trained network can then make predictions based on the relationships learned during training. A real life case study of the power industry in Nigeria was used in this work.

In this paper a supervised artificial neural network has been used for load forecasting.

- *Devesh Pratap Singh is a graduate in engineering from UP Technical University, India. E-mail: deveshpratap.85@gmail.com*
- *Mohd. Shiblee is currently pursuing PhD from IIT Kanpur, India, E-mail: shiblee@iitk.ac.in*
- *Deepak Kumar Singh is currently pursuing PhD from IIT Kanpur, India, E-mail: deepaks@iitk.ac.in*

Here, the neural network is trained on input data as well as the associated target values. The error measures used in the presented model is different from the conventional models. England ISO data has been used for the simulation.

In further sections of this paper, we discuss neural network in greater detail, introduce the conventional model of artificial neural network and the mathematical error models used in our model.

2 Neural Network

2.1 Introduction

Artificial neuron model is inspired by biological neuron. Biological neurons are the basic unit of brain for information processing system. Artificial Neural Network is a massive parallel distributed processing system that has a natural propensity for storing experimental knowledge and making it available for use [1]. The basic processing units in ANN are neurons. Artificial Neural Network can be divided in two categories viz. Feed-forward and Recurrent networks. In Feed-forward neural networks, data flows from input to output units in a feed-forward direction and no feedback connections are present in the network. Widely known feed-forward neural networks are Multilayer Perceptron (MLP) [3], Probabilistic Neural Network (PNN) [4], General Regression Neural Network (GRNN) [5] and Radial Basis Function (RBF) Neural Networks [6]. Multilayer Perceptron (MLP) is composed of a hierarchy of processing units (Perceptron), organized in series of two or more mutually exclusive sets layers. It consists of an input layer, which serves as the holding site for the input applied to the network. One or more hidden layers with desired number of neurons and an output layer at which the overall mapping of the network input is available [7]. There are three types of learning in Artificial Neural Networks, Supervised, unsupervised and Reinforcement learning. In supervised learning, the network is trained by providing it with input and target output patterns. In Unsupervised

ISSN 2229-5518

learning or Self-organization, an (output) unit is trained to respond to clusters of pattern within the input. In this paradigm the system is supposed to discover statistically salient features of the input population.Reinforcement learning may be considered as an intermediate form of supervised and unsupervised learning. Here the learning machine does some action on the environment and gets a feedback response from the environment. Most popular learning algorithm for feed-forward networks is Back-propagation. The first order optimization techniques in Back-propagation which uses steepest gradient decent algorithm show poor convergence. ANN is a very popular tool in the field of engineering. It has been applied for various problem like times series prediction, classification, optimization, function approximation, control systems and vector quantization. Many real life application fall into one of these categories. In the existing literature, various neural network learning algorithms have been used for these applications. In this paper, MLP neural network has been used successfully with different error measures for load forecasting.

2.2 New Error Measures

In Back-propagation algorithm, training of the neuron model is done by minimizing the error between target value and the observed value. In order to determine error between target and observed value, distance metric is used. It has been observed that Euclidean distance metric is the most commonly used for error measures in Neural Network applications. But it has been suggested that this distance metric is not appropriate for many problems [12]. In this work the aim is to find best error metric to use in Back propagation learning algorithm.

The likelihood and log-likelihood functions are the basis for deriving estimators for parameters, for given set of data. In maximum likelihood method we estimate the value of "*y*" for the given value of "*x*" in presence of error. (See equation (1))

$$y_i = x_i + e \tag{1}$$

Let x_i denote the data points in the distribution and let N denotes the number of data points. Then an estimator $\tilde{\mu}$ of µ can be estimated by minimizing the error metric with respect to $\tilde{\mu}$

$$\varepsilon = \sum_{i=1}^{N} f(x, \tilde{\mu}) \tag{2}$$

where $f(x, \tilde{\mu})$ is the distance metric. The error function ε is differentiated with respect to $\tilde{\mu}$ and equated to zero.

$$\frac{\partial \varepsilon}{\partial \tilde{\mu}} = \sum_{i=1}^{N} \frac{\partial}{\partial \tilde{\mu}} f(x, \tilde{\mu}) = 0 \tag{3}$$

Jie, et. al., [10-11] has proposed some new distance metrics based on different means. These distance metrics can be used to improve the performance of the neuron model for learning the best-fit weights of the neuron models. Distance metrics associated with the distribution models that imply the arithmetic mean, harmonic mean and geometric mean in (See Table 1) are inferred using equation (3).

TABLE 1
ERROR METRICS AND MEAN ESTIMATION

	Error metric	Mean
Arithmetic	$\varepsilon = \sum_{i=1}^{N} (x_i, \tilde{\mu})^2$	$\tilde{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$
Harmonic	$\varepsilon = \sum_{i=1}^{N} x_i \left(\frac{\tilde{\mu}}{x_i} - 1\right)^2 = 0$	$\tilde{\mu} = \frac{N}{\sum_{i=1}^{N} \frac{1}{x_i}}$
Geometric	$\varepsilon = \sum_{i=1}^{N} \left[\log\left(\frac{x_i}{\tilde{\mu}}\right)\right]^2 = 0$	$\tilde{\mu} = \left(\prod_{i=1}^{N} x_1\right)^{\frac{1}{N}}$

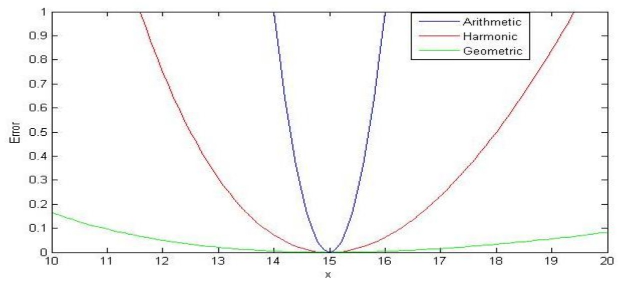

Fig. 1.the error criterion of the arithmetic mean, harmonic and geometric mean

Figure 1 illustrates the difference between the distance (error) metrics for the arithmetic mean, harmonic mean and geometric mean. For the sake of comparison, the value of µ is set to 15. It is found that in the distribution associated with the harmonic and geometric estimations, the observations x_i which are far away from $\tilde{\mu}$ will contribute less towards µ, in contrast to arithmetic mean and thus the estimated values will be less sensitive to the bad observations (i.e., observation with large variance), and therefore they are more robust in nature [9].

.Due to the robust property of harmonic and geometric distance metrics, generalizations of these distance metrics have also been done which may fit the distribution of data better [8-9]

2.2.1 Generalized Harmonic Type I Error Metric

Generalized Harmonic Type I Error metric derived from the Generalized Harmonic Type I mean estimation using equation (3) is given below.

Generalized Harmonic Type I Mean:

$$\tilde{\mu} = \frac{\sum_{i=1}^{N} \left(x_i \right)^{p-1}}{\sum_{i=1}^{N} \left(x_i \right)^{p-2}} \quad (4)$$

The corresponding error metric is given below:
Generalized Harmonic Type I Error Metric:

$$\varepsilon = \sum_{i=1}^{N} (x_i)^p \left(\frac{\tilde{\mu}}{x_i} - 1 \right)^2 \quad (5)$$

The parameter ‘p’ is used to define the specific distance metrics. If p=1, it becomes ordinary harmonic distance metric and for p=2, it will become Euclidean distance metric. Figure 2 illustrates the generalized harmonic Type I distance metric for the Generalized Harmonic Mean Type I.

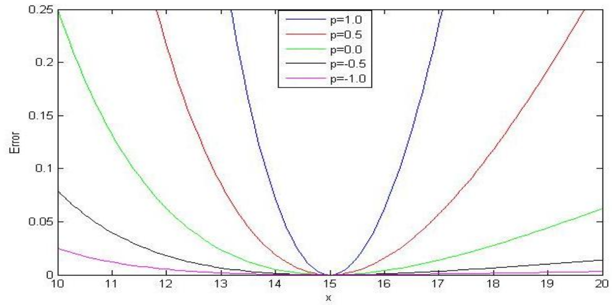

Fig. 2: The error criterion of the Generalized Harmonic Mean Type 1

2.2.2 Generalized HarmonicType II Error metric

Harmonic Type II Error metric derived from the Generalized Harmonic Type II mean estimation using equation (3) is given below.

GeneralizedHarmonicType II Mean:

$$\tilde{\mu} = \left[\frac{N}{\sum_{i=1}^{N} (x_i)^q} \right]^{-\frac{1}{q}} \quad (6)$$

Generalized Harmonic Type II Error Metric:

$$\varepsilon = \sum_{i=1}^{N} \left[(x_i)^q - (\tilde{\mu})^q \right]^2 \quad (7)$$

The parameter q is used to define the specific distance metrics. If q= -1, it becomes ordinary harmonic distance metric and for q=1, it will become Euclidean distance metric. Figure 3 illustrates the generalized harmonic Type II distance metric for the generalized harmonic mean Type II

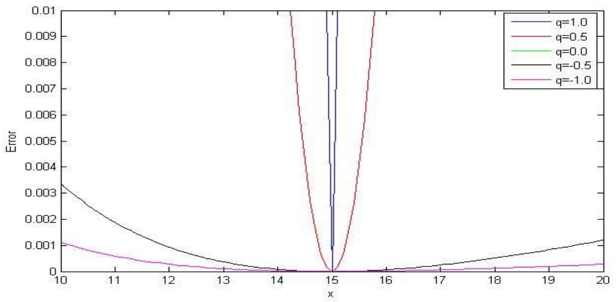

Fig. 3: The error criterion of the Generalized Harmonic Mean Type II

2.2.3 Generalized Geometric Error Metric

Generalized Geometric Error Metric derived from the Generalized Geometric Mean estimation using equation (3) is given below

.Generalized Geometric Mean:

$$\tilde{\mu} = \left[\prod_{i=1}^{N} (x_i)^{(x_i)^{2r}} \right]^{-\frac{1}{\sum_{i=1}^{N} (x_i)^{2r}}} \quad (8$$

The corresponding error metric is given below:
Generalized Geometric Error Metrics :

$$\varepsilon = \sum_{i=1}^{N} \left[(x_i)^r \log\left(\frac{x_i}{\tilde{\mu}} \right) \right]^2 \quad (9)$$

The parameter ‘r’ is used to define the specific distance metrics. For the generalized geometric mean estimation, if r = 0, it will become an ordinary geometric mean. Figure 4 illustrates the generalized geometric distance metric for the generalized geometric mean.

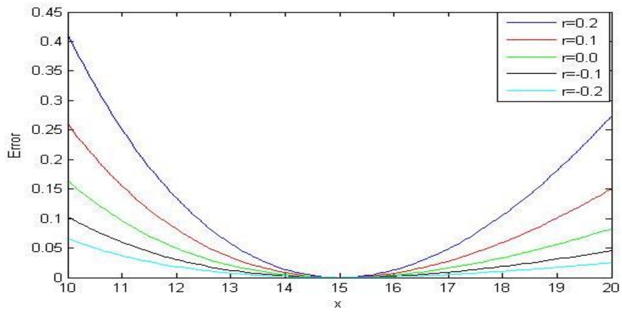

Fig. 4: The error criterion of the Generalized Geometric Mean

It is obvious that the generalized metrics correspond to a wide range of mean estimations and distribution models. These error metrics have been used in the Back propagation algorithm of MLP which enhances the prediction efficiency. Section 2.3; describes in brief the MLP model , the learning algorithm of MLP with new error metrics has been derived.

2.3 Multilayer Perceptron

Multilayer Perceptron (MLP) is composed of a hierarchy of processing units (Perceptron), organized in series of two or more mutually exclusive sets layers. It consists of an input layer, which serves as the holding site for the input applied to the network. One or more hidden layers with desired number of neurons and an output layer at which the overall mapping of the network input is available [13-14]. The input signal propagates through the network layer-by-layer [2].

The architecture of the MLP network is shown in Figure 5. It has been proved that a single hidden layer is sufficient to approximate any continuous function [15]. A three layer MLP is thus taken into account in this chapter. It has "nh" neurons in hidden layer and "no" neurons in output layer. It has "ni" inputs. The input and output vectors of the network are $X = [x_1, x_2, ..., x_{ni}]^T$ and $Y = [y_1, y_2, ..., y_{no}]^T$, respectively.

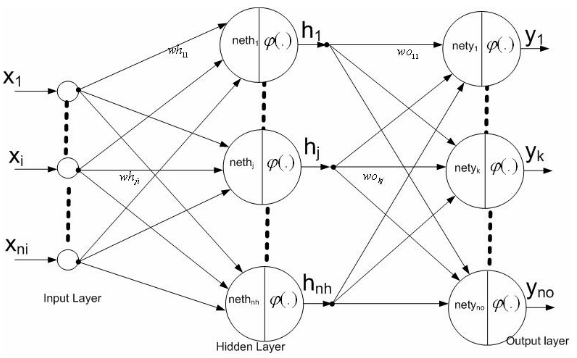

Fig. 5: Architecture of the MLP network

.

If the weight that connects the i^{th} neuron of the input layer with the j^{th} neuron of the hidden layer is w_{ij} and bias of the j^{th} neuron of the hidden layer is bh_j the net value of the j^{th} neuron can be given as:

$$neth_j = \sum_{i=1}^{ni} (wh_{ji} \cdot x_i + bh_j) \qquad j = 1, 2, ..., nh \qquad (10)$$

The output of the j^{th} neutron h_j of the hidden layer after applying activation can be given as

$$h_j = \varphi(neth_j) = \frac{1}{1+e^{-neth_j}} \qquad (11)$$

Similarly the net value "nety$_k$ "and the final output y_k of k^{th} neuron of the output layer can be given as

$$nety_k = \sum_{i=1}^{nh} (wo_{kj} \cdot h_j + bh_k) \qquad k = 1, 2, ..., no \qquad (12)$$

where φ is the activation function.

- h_j is the output of j^{th} neuron of hidden layer.
- wo_{kj} is the connection weight of j^{th} hidden layer neuron with k^{th} output layer neuron.
- bo_k is the weight of the bias at k^{th} neuron of output layer.
- no is the number of output neuron.

Output of k^{th} neuron of output layer is

$$y_k = \varphi(nety_k) = \frac{1}{1+e^{-nety_k}} \qquad (13)$$

2.3.1 Training algorithm of MLP with new error metrics

In this section the error back-propagation learning of MLP with different error metrics have been derived. Let E denote the cumulative error at the output layer. In BP algorithm aim is to minimize the error at the output layer.

The weight update equations using gradient descent rule are given bellow:

$$wh_{ji}(new) = wh_{ji}(old) + \eta \cdot \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial wh_{ji}} \qquad (14)$$

$$bh_j(new) = bh_j(old) + \eta \cdot \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial bh_j} \qquad (15)$$

$$wo_{kj}(new) = wo_{kj}(old) + \eta \cdot \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial wo_{kj}} \qquad (16)$$

$$bo_k(new) = bo_k(old) + \eta \cdot \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial bk_k} \qquad (17)$$

where, η is the learning rate and

$$\frac{\partial y_k}{\partial wh_{ji}} = [\sum_{k=1}^{no} (1-y_k) \cdot y_k \cdot w_{kj}] \cdot (1-h_j) \cdot h_j \cdot x_i \qquad (18)$$

$$\frac{\partial y_k}{\partial bh_j} = [\sum_{k=1}^{no} (1-y_k) \cdot y_k \cdot w_{kj}] \cdot (1-h_j) \cdot h_j \qquad (19)$$

$$\frac{\partial y_k}{\partial bo_k} = (1-y_k) \cdot y_k \qquad (20)$$

$$\frac{\partial y_k}{\partial wo_{jk}} = (1-y_k) \cdot y_k \cdot h_j \qquad (21)$$

ISSN 2229-5518

For different error criterion only $\partial E / \partial y$ will change and is shown as follows: This is dependent on the distance metric used in computation of the total error E.

Case 1: Least Mean Square Error

$$LMSE= E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{no}(t_k - y_k)^2 \quad (23)$$

$$\text{Then, } \frac{\partial E}{\partial y_k} = -\eta \cdot (t_k - y_k) \quad (24)$$

Case 2: Geometric error metric

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{no}[\log(\frac{y_k}{t_k})]^2 \quad (25)$$

$$\text{Then } \frac{\partial E}{\partial y_k} = -\eta \cdot (\log(t_k) - \log(y_k)) \cdot \frac{1}{y_k} \quad (26)$$

Case 3: Harmonic error metric

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{no} t_k \cdot (\frac{y_k}{t_k} - 1)^2 \quad (27)$$

$$\text{Then } \frac{\partial E}{\partial y_k} = \eta \cdot t_k \cdot (y_k / t_k - 1) \quad (28)$$

Case 4: Generalized geometric error metric

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{no}[t_k^{\,r} \cdot \log(\frac{t_k}{y_k})]^2 \quad (29)$$

$$\text{Then } \frac{\partial E}{\partial y_k} = -\eta \cdot t_k^{\,r} \cdot (\log(t_k) - \log(y_k)) \cdot t_k^{\,r} \cdot \frac{1}{y_k} \quad (30)$$

Case 5: Generalized harmonic error metric Type I

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{no} t_k^{\,p} \cdot (\frac{y_k}{t_k} - 1)^2 \quad (31)$$

$$\text{Then } \frac{\partial E}{\partial y_k} = \eta \cdot (t_k^{\,p}) \cdot (y_k / t_k - 1) \quad (32)$$

Case 6: Generalized harmonic error metric Type II

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{no}(t_k^{\,q} - y_k^{\,q})^2 \quad (33)$$

$$\text{Then } \frac{\partial E}{\partial y_k} = -\eta \cdot (t_k^{\,q} - y_k^{\,q}) \cdot q \cdot y_k^{\,q} \quad (34)$$

Where, y_t denotes the desired vlue of neuron.and t_i notes the target value for the i^{th} pattern.

3 Results

The data is taken from England ISO, and the simulation is done using MATLAB.the network designed is of two lagers with 20 neuron .For converentional model mean absolutes error (MAE) is used. To display the result clearly only forcasting of one week is given. The results are shown in following figures. The x axis in all the figures below represents the days (data here shown is only for 1 week; 1st January 2008 to 8th January 2008) and the y axis depict the forecasted load demand (in MW).

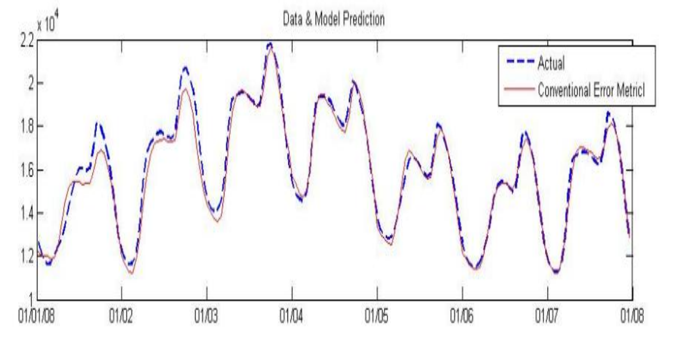

Fig. 6: Forecasting using MAE (conventional model).

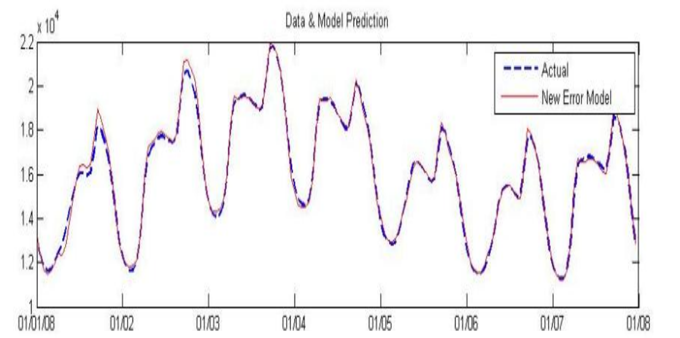

Fig. 7: Forecasting using Generalized Harmonic Type I Error Metric

ISSN 2229-5518

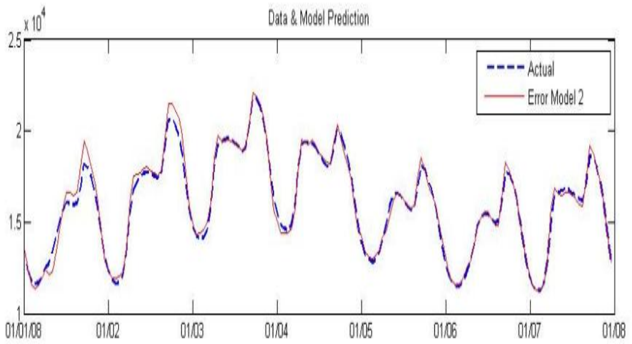

Fig. 8: Forecasting using Generalized Harmonic Type II Error Metric

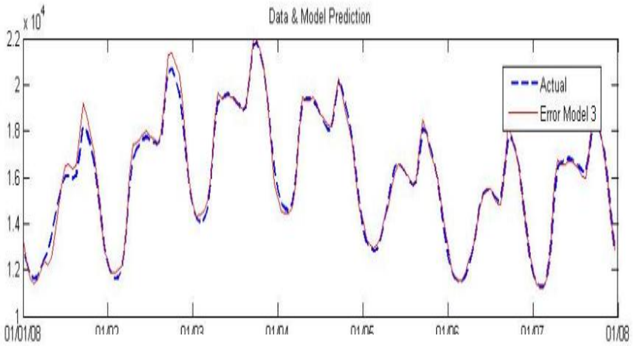

Fig. 9: Forecasting using Generalized Geometric Error Metric

4 CONCLUSION

As it is clearly analyazed from the result. The new models are giving better result from the convestional model.this result are justified from the theory that the generalized mean models will capture the variation well.the reason is clear that the variation are largely nonlinear & stocasting in nature so conventional linear models are unable to catch it.

References.

[1] S. Haykin. : Neural Networks: A Comprehensive Foundation, *Macmillan College Publishing Company,* New York, 1994

[2] Engle,R.F.,Mustafa,C.,andRice,J. 1992."Modelling peak electricity Demand".Journal of forecasting.11:241-251. & Eugene, A.F and Dora,G.2004.Load Forecasting.269-285.State University of New York :New York

[3] D.E. Rumelhart, "Parallel Distributed Processing", Plenary Lecture presented at Proc. *IEEE International Conference on Neural Networks,* San Diego, California, 1988.

[4] Donald F. Specht, "Probabilistic neural networks", in *Neural Networks*, 3, pp. 109-118, 1990

[5] Donald F. Specht, "A General Regression Neural Network", *IEEE Trans on Neural Networks,* Vol 2, 1991.

[6] S. Chen, C.F.N. Cowan, P.M. Grant, "Orthogonal least squares learningalgorithmfor radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, pp.302-309, 1991.

[7] Satish Kumar. : Neural Networks: A Classroom Approach, *Tata McGraw Hill Publishing Company,* New Delhi, 3rd Ed. 2007.

[8] W. Schiffmann, M. Joost, and R. Werner, , "Optimization of the backpropagation algorithm for multilayer perceptrons", Univ. Koblenz, *Inst. Physics,* Rheinau 3-4, Germany.

[9] R. Battiti, "First and second-order methods for learning: between steepest descent and Newton's method", *Neural Computation*, Vol. 2, 141-166, 1992.

[10] J. Yu, J. Amores, N. Sebe, Q. Tian, "Toward an improve Error metric", *International Conference on Image Processing (ICIP),* 2004.

[11] J. Yu, J. Amores, N. Sebe, Q. Tian, "Toward Robust Distance Metric Analysis for Similarity Estimation", *IEEE Computer Sosciety Conference on Computer Vision and Pattern Recognition,* 2006.

[12] W. J. J. Rey, "Introduction to Robust and Quasi-Robust Statistical Methods", *Springer-Verlag*, Berlin, 110–116, 1983.

[13] K. Hornik, M. Stinchombe and H. White, "Multilayer feedforward networks are universal approximators", *Neural networks*, vol. 2, pp. 359-366, 1989.

[14] J. M. Zurada, "Introduction to Artificial Neural Systems", *Jaicob publishing house, India,* 2002

[15] G. Cybenko, "Approximation by superpositions of a sigmoidal function", *Mathematics of Control, Signals, and Systems*vol. 2 pp.303-314, 1989.