General Regression Neural Network versus Back Propagation Neural Network for Prediction of Reheater and Super Heater Sprays in Thermal Power Plants

K.S. Madhavan

Abstract**:** ***Neural Network models are used for Reheater and Super heater spray prediction in Thermal Power Plants. This paper makes a comparative study of the General Regression Neural Network (GRNN) model versus the Back propagation Neural Network (BPNN) model for the quality and accuracy of prediction of Reheater and Super heater Sprays in Thermal Power Plants. It proves that GRNN is better and gives more stable prediction within range; the glitches between the predicted and actual values being less in number as well as value.***

Keywords**:** ***Back propagation Neural Network, General Regression Neural Network, Reheater Spray, Super heater Spray.***

I. INTRODUCTION

Artificial Neural Network (ANN) has proved to be a contemporary technology for prediction, classification and function approximation type of problems in the domain of artificial intelligence. But ANN models broadly could be using the sigmoidal function or the Gaussian function for approximation. The sigmoidal function approximation forms the heart of the BPNN models, while the Gaussian function is the essence of the GRNN models.

BPNN model essentially comprises of input neurons, hidden neurons and output neurons connected by weight matrices that are summed up systematically and fed through a sigmoidal activation function that works in two passes: In the forward pass the weights are initialized and the errors for the initial weights are calculated, in the backward pass the weights are updated and the error minimized. GRNN model essentially is a three-layer network. Learning takes place in the second and third layers in two stages. In the second layer learning is unsupervised through k-means clustering and Gaussian approximation. If the predicted value is nearer the middle of the Gaussian function, the predicted value is closer to the actual value. In the third layer the learning is logistic and uses the supervised method for approximation.

II. BACK PROPAGATION NEURAL NETWORK (BPNN)

A. The Historical Development and Architecture of BPNN

Author (s) can send paper in the given email address of the journal. There are two email addresses.

Revised Manuscript Received on March 18, 2020.

K.S Madhavan Sr. DGM (SBE), BHEL Corporate R&D, Vikasnagar, Hyderabad, India.

It is compulsory to send paper in both email address the first version of the formal neuron was introduced by Sir Warren Mc Culloch and Walter Pitts. Hebb introduced the first empirical rule for the modification of the synaptic weights [1]. The concept of artificial neural networks (ANN) introduced by Mc Culloch and Pitts [2] is a concept which is based on mathematical and data-processing models, assemblies of calculating units called formal neurons, and whose original inspiration was a model of the human nervous cell [3-4].

The neurons are inter-connected by one-way networks called "connections" and realize an algebraic function of its inputs [5]. Each calculating unit has only one output connection which can be duplicated in as many specimens as wished, the duplicates transmitting the same signal.

Back Propagation Neural Network (BPNN) comprises of input layer, hidden layer(s) and output layer. The learning process involves the finding of connection weights and their patterns. **Figure 1** presents the BPNN architecture. It can be divided in three sets: input neurons unit which receives the input data in the shape of a scalar vector values, the input vector represents the parameters of the problem. These values are communicated to the neurons via their external input values. Thus, they influence their activation, and by extension, the neural networks behavior; output neurons layer whose activations constitute the output vector. They are collectively interpreted like the neural networks result. The same neuron can be the input and the output of the neural networks at the same time; therefore, input unit and output unit are not necessarily disjoint; hidden neurons layer which connecting the input unit and output unit represents the correlations encoded by the system. In general, the presence of hidden neurons in neural networks reduces its computing power, and allows it to tackle more complex problems [6].

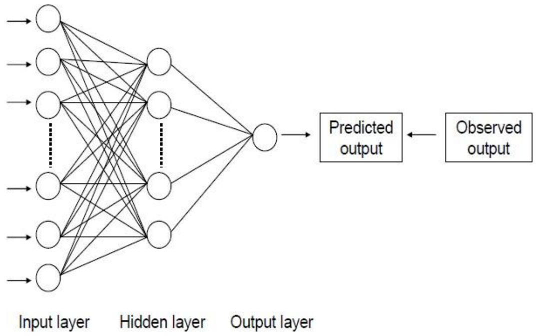

Figure 1: The Directed Graph of BPNN

The mathematical framework of Backpropagation

Back-propagation or Multilayer Perceptron is a very popular concept in the entire gamut of Neural Network Technology. It is a multilayer feedforward network with gradient descent based delta-learning rule [7]. Back-propagation provides a computationally effective method for updating and changing the weights in a feedforward network, to learn a training set of input-output examples. The nature of training is supervised. It uses a sigmoidal activation function for the somatic operation. The sigmoidal function used is of the form [8]:

$$f(x) = 1/1 + e^{-(\Sigma XiWi + \theta i)} \quad (1)$$

Where Wi is the weights linking the neurons

θi is the threshold

The output varies from 0 to 1 It can also be of the form:

$$\mathrm{F(x)} = \tanh(x) \text{ or}$$
$$\mathrm{F(x)} = 1 - e^{-(\Sigma XiWi + \theta i)} / 1 + e^{-(\Sigma XiWi + \theta i)} = 2 * f(2x) - 1 \quad (2)$$

The output varies from -1 to 1.

The Backpropagation Algorithm and delta learning rule

$$Wji(t+1) = Wji(t) + \Delta Wji + \delta(Wji(t) - Wji(t-1)) \quad (3)$$

where η is the learning rate and δ is the momentum

The learning rate is used to ensure better learning in the algorithm and the momentum term is introduced to ensure faster convergence 0 <δ< 1.

$$\Delta Wji = \eta\delta Xj \quad (4)$$

$$\delta Xj = Xj(1 - Xj)\,(Tj - Xj) \quad (5)$$

The disadvantage of backpropagation algorithm is the solution can get trapped in local minima. It also takes longer time for convergence to the target solution.

III. GENERAL REGRESSION NEURAL NETWORK (GRNN)

General Regression Neural Network estimates the vector Y in terms of X. In control theory, the dependent variable, Y. is the system output and the independent variable, X, is the system input [9]. In order to implement effective neural network control, it is usually necessary to assume some functional form with unknown parameters a_i.

In the case of linear regression, for example, the output, Y, is assumed to be a linear function of the input, X, and the unknown parameters, a_i are linear coefficients

In this paper, the joint Probability Density Function (PDF) will be estimated from examples using nonparametric estimators. The learning takes place in a very short time which can be estimated by the propagation time.

The mathematical framework of General Regression

Assume that $f(x, y)$ represents the known joint continuous probability density function of a vector random variable, **x**, and a scalar random variable, y.

For a nonparametric estimate of $f(x, y)$, we will use the consistent estimators proposed by Parzen[10] that has been shown to be in conformity to the multidimensional case by Cacoullos[11]. These estimators are a good choice for estimating the probability density function, **f**, provided the assumption that the underlying density is continuous is true and that the first order partial derivatives of the function evaluated at any **x** are negligible.

The estimate $\hat{Y}(X)$ can be perceived as a weighted average of all of the observed values, $\mathbf{Y_i}$, where each observed value is weighted exponentially according to its Euclidean distance from **X**. When the smoothing parameter **σ** is increased, the estimated probability density is forced to be smooth and in the limit becomes a

$\hat{Y}(X)$ assumes the value of the $\mathbf{Y^i}$ associated with the observation closest to **X**.

The norm is to find **σ** on an empirical basis as it is not possible to compute it optimally for a given number of observations. Grabec[12] used the same estimator to predict chaotic behavior.

Implementation of General Regression Neural Network

Figure 2 shows the directed graph of GRNN in its adaptive form [13]. The estimate of Y on X is given by the formula:

$$\hat{Y}(X) = \frac{\sum_{i=1}^{m} A^i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{m} B_i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)} \quad (6)$$

The output unit merely divides $\hat{Y}f(X)K$ by **f(X)K** to yield the desired estimate of Y as given by the following equation.

$$E(Y|X) = \frac{\hat{Y}f(X)K}{F(X)K} \quad (7)$$

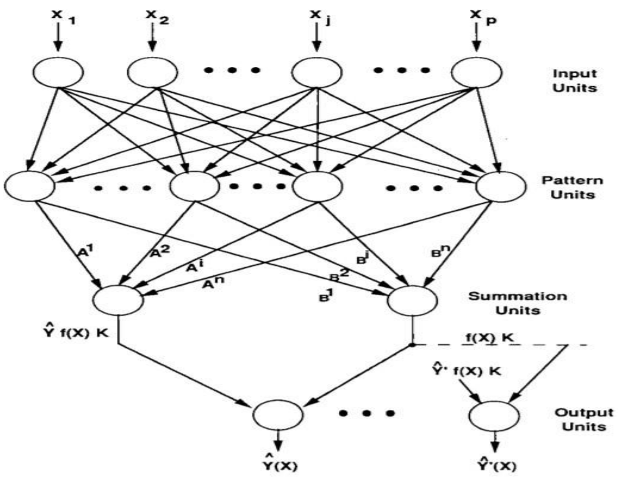

Figure 2: The directed graph of GRNN

In **Figure 2** is shown a feedforward network that can be used to estimate a vector **Y** from a measurement vector **X**.

IV. COMPARISON OF BPNN & GRNN FOR PREDICTION OF SPRAYS IN A BOILER

Illustrated below in **Figure 3** is a snapshot of the neural network model used to predict Reheater and superheater sprays with the inputs, outputs, hidden layer, weights at one instance of time. These are the results of a supervised learning algorithm. The weights represent the strength of the signal while the hidden neurons reduce computational power. Burner Tilt, Mill combination, Excess Air and Load are the four inputs to the neural network. Superheater spray and reheater spray are the two outputs of the BPNN.

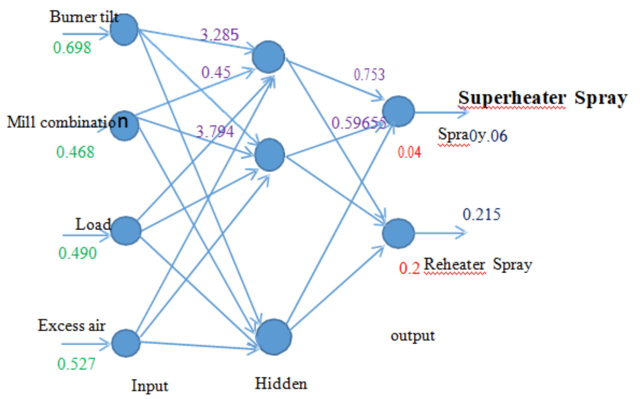

Figure 3: Artificial Neural Network Model for predicting Reheater Spray and Superheater Spray

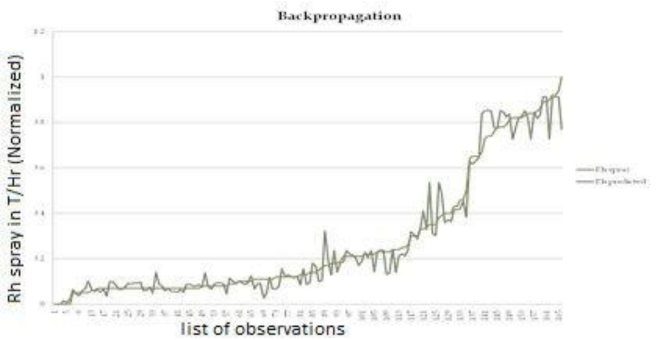

Figure 4: Results of Backpropagation prediction of Reheater Spray in normalized units

The work done in this paper is based on laboratory scale simulation with performance guarantee test data taken from our sites. The backpropagation neural network model & general Regression neural network model are generated through NeuroShell2 software from Maryland USA. The smoothing factor of the general regression neural network was obtained through a genetic algorithm.

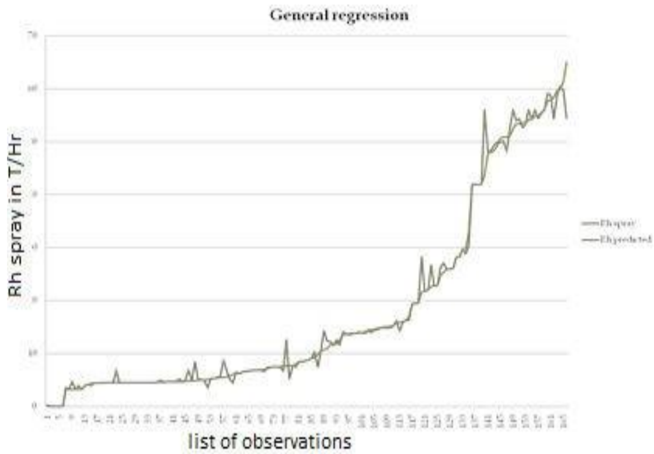

Figure 5: Results of General Regression prediction of Reheater Spray in absolute units

Figures 4, 5 illustratively capture the effect of prediction of Reheater spray through Backpropagation and General Regression Neural Network models. On examining the figures, we perceive that the stability of prediction through GRNN is better and more accurate.

V. EXPERIMENTAL RESULTS

The three-layer backpropagation training was executed with the following parameters and rendered the below predictions [14].

1) Table 1: Experimental results of Backpropagation model

Training Parameters	Output	Reheater Spray	Superheate Spray
Best Test set: 20%	R squared	0.9624	0.9067
Learning Rate: 0.1	r squared	0.9633	0.9069
Incremental Learning Rate: 0.0002	Mean Sq. Error	12.230	31.049
	Mean Abs Error	2.182	3.802
Momentum: 0.1	Min abs error	0	0
	Max abs error	17.803	20.289
Initial Weights: 0.3	Corr. Coeff.(r)	0.9815	0.9523
Learning Epochs: 5047			

The General Regression Neural Network was executed with a smoothing factor of 0.0255294 and rendered the following predictions [15]:

2) Table 2: Experimental results of General Regression Model

Output	ReHeater Spray	SuperHeater Spray
R squared	0.9908	0.9738
r squared	0.9909	0.9740
Mean squared error	3.006	8.718
Mean absolute error	0.699	1.279
Min absolute error	0	0
Max absolute error	12.450	22.200
Correlation Coefficient (r)	0.9954	0.9869

VI. CONCLUSION

The experimental results along with the graphical results conclusively prove that the GRNN model gives superior results to the BPNN model. The correlation coefficient is a measure of the match in predictions and clearly indicates that GRNN is superior to BPNN. The graphical results also indicate less number of glitches and better prediction quality with more stable prediction for GRNN model.

ACKNOWLEDGMENT

The author is grateful to the BHEL Management for according permission to publish this work..

REFERENCES

1. Pelissier, A. & Tête, A. Sciences cognitives : textes fondateurs (1943-1950), Paris, Presse Universitaire Française (1995).
2. Mc Culloch, W. C. & Pitts, W. H. A logical calculus ideas immanent in nervousactivity, Bulletin of Mathematical Biophysics, vol. 5, 115-133 (1943).
3. Jodouin, J. F.Les réseaux neuromimétiques, Paris, Edition Hermès (1994).
4. Nelson, M. M. & Illingworth, W. T. A practical guide to neural nets, New York, Addison Wesley, 3rd edition (1991).
5. Personnaz, L. & Rivals, I. Réseaux de neurones formels pour la modélisation, la commande et la classification, Paris, Collection sciences et techniques de l"ingénieur,CNRS Edition (2003).
6. Kaul, M., Hill, R. L. & Walthall, C. Artificial neural networks for corn and soybean yield prediction. Agric. Sys., 85, 1-18 (2005).
7. D.E. Rumelhart, G.E. Hinton, and R.J. Williams, Learning representations by back- propagating errors, Nature, Vol. 323, pp. 533-536 (1986).
8. G. Cybenko, Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals, and Systems, Vol. 2, No. 4, pp. 303-314 (1989).
9. Donald F. Specht, "A General Regression Neural Network, IEEE Transactions on Neural Networks", Vol. 2, No. 6, November (1991).
10. Emmanuel Parzen, "On estimation of probability density function and mode", The Annals of Mathematical Statistics, Stanford University (1962).
11. Theophillos Cacoullos, "Estimation of Multivariate Density", Technical Report No. 40, University of Minnesota (1964).
12. I. Grabec, "Explanation of Random Vibrations in Cutting on Grounds of Deterministic Chaos," Robotics and Computer& Integrated Manufacturing, Vol. 4, pp. 129-134 (1988).
13. LiminFu, "Neural Networks in Computer Intelligence", Tata McGraw-Hill Edition (2003).
14. Madhavan K.S. et al, "ANN Prediction Tool for Reheater and Superheater Sprays in Boiler Performance", Vol. 6, pp 335 – 337, ICECT, April (2011).
15. S. Sujit Kumar, "Performance Prediction of Boilers in Thermal Power Plants using Neural Networks", M. Tech thesis in Computer Science and technology, GITAM University (2013).

AUTHORS PROFILE

K.S Madhavan Sr. DGM (SBE), BHEL Corporate R&D, Vikasnagar, Hyderabad 500093