A Novel Method to Select Hidden Neurons in ELMAN Neural Network for Wind Speed Prediction Application

M. MADHIARASAN[1*], S. N. DEEPA[2]
[1*, 2] Department of Electrical & Electronics Engineering
[1*]Research Scholar, [2]Associate Professor, Anna University
Anna University Regional Office, Navavoor, Coimbatore
INDIA
[1*]mmadhiarasan89@gmail.com, [2]deepapsg@gmail.com

Abstract: - This paper proposes a novel method to select hidden neurons in ELMAN neural networks for wind speed prediction application. Either over fitting or under fitting problem caused due to the random choice of hidden neuron numbers in artificial neural network. This paper suggests the solution to solve either over fitting or under fitting problems. In order to select proper hidden neuron numbers, 75 different criteria tested by the means of statistical errors. The simulation results proved that proposed approach improves the accuracy and reduce the error to the least. The perfect building of ELMAN network with five inputs using fixation criteria is validated based on convergence theorem. To evaluate the performance of the proposed approach simulation were performed on real-time wind data. Comparative analysis has performed to select the hidden neuron numbers in neural networks. The presented approach is very simple, with the least error, and more effective to select the amount of hidden neurons in ELMAN neural network.

Key-Words: - Novel Criteria; Hidden Neurons; ELMAN Neural Network; Prediction; Wind Speed.

1 Introduction

In neural network design process one of the major issues is fixation of hidden neuron numbers. This plays an important role to get a very lowest error in neural network for prediction of wind speed.

Artificial Neural Network (ANN) is nonlinear information processing system, which developed from interconnected elementary processing devices called neurons. ANN is an information processing model that inspired by the biological nervous system [1]. ANN widely used for various applications due to feature such as good self learning ability, adaptability, real-time operation, fault tolerance ability, easy implementation and cost effectiveness. ANN classified as feed-forward and feedback (recurrent) networks. A network arranged into layer with no feedback path called feed-forward network. The arranging is from input, through one (or) more hidden layers, to the output layer, example: MLP, BPN, RBFN, etc. A network with feedback path that occur between the layers (or) within the layer called feedback network, example: ELMAN, Hop-field, Boltzmann machine, etc. The feedback network has a profound effect on the modeling nonlinear dynamic phenomena performance and its learning capacity.

The following problem involved in neural network model designing and training for a specific application.

1) Fixing how many hidden layer in the network.
2) Fixing how many hidden neurons in each hidden layer.
3) Deciding how many training pair to use in the network.
4) Deciding which training algorithm use in the network.
5) Deciding what architecture use in the neural network.
6) Searching a global optimal solution that prevents local mini ma problem.
7) Evaluating the network to test for over fitting (or) under fitting problem [2, 3].

The neural network designed with too few hidden neuron produce large training errors and generalization errors because of the under fitting problem, while too many hidden neurons used in network cause low training and high generalization error due to over fitting issue [4]. So, the hidden neuron has a profound impact on the errors. The errors used to decide the network stability. Network with the least error has the better stability while the network with higher error has the poor stability [5]. The process of selecting the hidden layer numbers and amount of hidden neurons in each hidden layers is still confusing and a challenging task [6]. Many researchers design the neural network and evaluating the hidden neuron number in the hidden layer, but no one can do the right. The network training accuracy estimated by the characteristic such as neural network architecture, hidden layer numbers, hidden neuron numbers in hidden layers, activation function, inputs and outputs and updating of weights.

Due to the environmental degradation and depletion of conventional energy, much attention focused in the wind energy. Wind is one of the most flexible and tractable of all energy sources, the mechanical energy derived directly from wind readily and efficiently converted into other form of energy. The dual aims of global reduction of CO_2 emission and improving energy security (energy policy goals in many countries) coincides in the increasing use of wind energy for electricity generation. Due to the time varying and randomness the wind speed prediction is very important hot topic in research.

Need for wind speed prediction is given below. Small fraction deviations of wind speed will lead to a large error output of wind driving systems.

1) Reliable and high quality operation of power system.
2) Effective integration of wind power to the electrical power grid.
3) Reduces the operating cost of the wind power generation.
4) To make low spinning reserve.
5) To aid planning and control of power system and wind farm.

The statistical errors used to measure quality of predicted wind speed obtained by neural network. During training process of the network the generalization performance differ overtime.

In this paper different 75 criteria considered for proper selection of hidden neuron numbers in the hidden layer of ELMAN neural network, suggested all 75 criteria satisfied based on the convergence theorem. The proposed ELMAN neural design adapted for prediction of wind speed in renewable energy systems. The main aim is to meet the minimal error, improve the network stability and better accuracy compared to other existing approaches.

2 Literature Review

A lot of research experiments attempted and suggested to select the proper hidden neuron numbers in neural network by several researchers. A comprehensive review is made to fix the hidden neuron numbers in neural network design and are illustrated as follows.

Shih-Chi Huang and Yih-Fang 1991[7], analyzed problem associated for multilayer perceptron with one hidden layer and suggest a method to search the needed hidden neuron in multilayer perceptron. Arai 1993 [8], carried out work on a Two Parallel Hyper plane Methods (TPHM) to select the number of hidden neuron in network design, and stated that $2^n/3$ hidden neuron sufficient for network design. Masafumi Hagiwara 1994 [9], pointed out the consuming energy method and weight power method to get the size minimization. Noboru Murata et al 1994 [10], presented a new method to estimating the number of hidden neurons in artificial network model using Network Information Criterion (NIC). Jin-Yan Li et al. 1995 [11], performed estimation theory based investigation to search the optimal number of hidden units in the higher order feed-forward neural network and applied the presented method for time series prediction and system identification problems. Onoda, T 1995 [12], carried out work on a construction of optimal architecture of multi layered neural network based on the Neural Network Information Criterion (NNIC) to solve the selection of model and hidden neurons problems.

Tamura and Tateishi, M 1997 [13], pointed out the neural network capability using a finite number of hidden unit. The minimum error is obtained for three-layered feed-forward network and four-layered feed-forward network with $N-1$ and $(N/2)+3$ number of hidden neuron respectively and stated that compared to three-layered network the four –layered network superior. Osamu Fujita 1998 [14], suggested feed-forward neural network design with needed number of hidden neuron based on statistical estimation. The number of inputs and nonlinearity of hidden neuron has a profound impact on the output error. The necessary number of hidden neuron decide as $N_h = K\log\left(\left\|P_c^{(0)}Z\right\|/C\right)/\log S$ where, K - data set, C - allowable output error, S - total number of candidates that are randomly find optimal hidden neurons. Keeni, K et al. 1999 [15], presented fast learning multi-layer neural networks for pattern classification based on the initial weight and hidden neuron estimation.

Guang-Bin Huang 2003 [16], proposed a model for learning capability and storage capacity of two- hidden layer feed-forward neural networks. The needed number of hidden neuron in single and two hidden layers defined as $N_h = \sqrt{(m+2)N} + 2\sqrt{N/(m+2)}$ and $N_h = m\sqrt{N/(m+2)}$. H, C Yuan et al. 2003 [17], performed information entropy based estimation of hidden neuron in feed-forward neural network. Zhaozhi Zhang et al 2003 [18], carried out work on a three layer binary neural network design based on a set covering algorithm (unit sphere covering (USC) of the hamming space (HS)). Based on the SCA proper number of hidden neuron is $3L/2$ where, L - number of unit sphere contained in N - dimensional Hamming space. K, Z, Mao and Guang-Bin Huang 2005 [19], suggested data structure preserving criterion based selection of hidden neuron for RBF neural network classifier.

E, J, Teoh et al. 2006 [20], proposed singular value decomposition based estimation of hidden neuron in feed-forward neural network. Xiaoqin Zeng and Daniel S, Yeung 2006 [21], pointed out quantified sensitivity measures based pruning technique, to remove as many neurons to get the lowest relevance hidden neuron for MLP.

Bumghi Choi et al. 2008 [22], carried out work on a separate learning algorithm for two layered feed-forward neural network to solve the local mini ma issue with large number of hidden neurons. Jinchuan Ke and Xinzhe Liu 2008 [5], investigated optimal number of hidden neuron and hidden layer in neural network design for prediction of stock price. The training and generalization errors determined for forty cases with different hidden neuron using formula $N = \left(N_{in} + \sqrt{N_p}\right)/L$ where, N_{in} is the number of input neuron, N_p is the number of input sample, L is the number of hidden layer. The best optimal hidden neuron selected based on the lowest generalization error. Min Han and Jia Yin 2008 [23], suggested support vector machine and ridge regression based hidden neurons selection for wavelet network. Nan Jiang et al. 2008 [24], performed implementation of an arbitrary function by using the lower bound on the number of hidden units in three-layer multi-valued multi threshold neural networks. The presented approach number of hidden neuron $N_h = q^n$ where, q is valued upper bound function. Shuxiang Xu and Ling Chen 2008 [4], carried out work on a novel approach for estimating the optimal number of hidden neuron for FNN in data mining. The suggested number of hidden neuron is $N_h = C_f\left(N/(d \log N)\right)^{1/2}$ where, C_f - first absolute moment of the Fourier magnitude distribution of the target function, d - input dimension of target function, N - number of training pair. Suggested number of hidden neuron N_h lead to the lowest RMS errors. Limitation is local mini ma problem is not addressed. Stephen Trenn 2008 [25], presented an approximation order and needed number of hidden neurons for multi-layer perceptron network. The suggested formula is function of input variable and desired approximation order. The formula for number of hidden neuron is $N_h = (n + n_o - 1)/2$ where, n_o - number of inputs. Katsunari Shibata and Yusuke Ikeda 2009 [26], analyzed the neural network learning stability based on influence of number of hidden neuron (N_h) and learning rate (η).The number of hidden neuron and learning rate are $N_h = \sqrt{N^{(i)}N^{(0)}}$ and $\eta = 32/\left(\sqrt{N^{(i)}N^{(o)}}\right)$respectively where, $N^{(i)}$- number of input neurons, $N^{(o)}$- number of output neurons.

Doukim, C, A et al. 2010 [27], suggested a coarse to fine search method to search the number of hidden neuron in MLP neural network for skin detection application. Junfang Li et al. 2010 [28], suggested ELMAN neural network based one step ahead wind speed prediction. The result confirms that the pointed out approach suitable for various number of hidden neuron and various tested input data. David Hunter et al. 2012 [3], investigated proper selection of neural network architecture and size. The generalization problem, various network topologies (MLP, BMLP & FCC) efficiency and learning algorithm (EBP, LM & NBN) are analyzed to choose the suitable neural network architecture and size. The number of hidden neuron implemented for MLP, BMKP and FCC based on the following formula $N_h = N + 1$, $N_h = 2N + 1$ and $N_h = 2^n - 1$ respectively. The result confirms the following BMLP network trained easily compared to MLP, Many trials are not needed to select the number of hidden neuron. Jianye Sun 2012 [29], pointed out hidden node selection strategies and learning algorithm for newly designed local coupled feed-forward neural network classifier. Ramadevi, R et al. 2012 [30], investigate the hidden neuron role in ELMAN recurrent neural network for classification of cavitation signals and based on trial and error approach the optimal number of hidden neuron fixed. Saurabh Karsoliya 2012 [6], analyzed BPNN architecture designing problem of approximation of needed number of hidden layer and number of neurons and resolved issue of estimation of needed number of hidden neuron and number of hidden layer. Gnana Sheela, K and S, N, Deepa 2013 [31], analyzed the methods to fix number of hidden neurons in neural network and suggest $\left(4n^2+3\right)/\left(n^2-8\right)$ criteria among the various 101 criteria to select the number of hidden neuron in ELMAN neural network for wind speed prediction. Guo Qian and Hao Yong 2013 [32], presented BP neural network based rural per capita living consumption prediction. The number of hidden neurons estimated as $N_h = \sqrt{n+m+a}$ to make the better accuracy where, n - number of nodes in input layer, m - number of nodes in output layer, a - integer among 0 to 10. KuldipVora and Shruti Yagnik 2014 [33], pointed out a new algorithm having structural changes in feed-forward neural network to solve the local mini ma issue with problem of large hidden neurons. The presented approach applied for classical problems like soil data classification and parity problem.

According to the literature review the following points are observed:

1) Much research implemented different approach for finding the proper number of hidden neuron in network design to meet the faster convergence, better efficiency, minimal errors and improved accuracy.
2) In neural network design process the fixation of hidden neuron play an important role, it helps make the better performance with minimal error.

3 Problem Description

In this paper analyze proper fixation of hidden neuron numbers in ELMAN neural networks. Important problem to fix the hidden neuron numbers to solve the particular task. The neural network with few hidden neuron units may not have sufficient powers to satisfy the requirements such as accuracy, error precision and capacity. The problem of over fitting data caused by over training. Over training issue has occurred in the neural network design. Therefore, selection of hidden neuron numbers in neural network design is one of the important problems. Determination of optimal hidden neuron numbers and parameters crucial and important task for neural network design. It means that, it requires determining the divergence between neural network and an actual system. So many researcher emphases to make the better performance by tackling this problem, but in neural network there is no other way to fix the hidden neuron numbers without attempting and evaluating during the training process and calculating the generalization error. If the hidden neuron number is large the hidden output connection weights become so small, learn easily and also the trade-off in stability between input and hidden output connection exists. The output neurons become unstable for large hidden neuron numbers. If the hidden neuron number is small it may fall in the local mini ma because of slow learning of the network and hidden neuron become unstable. Hence, the selection of hidden neuron numbers is important critical problem for neural network design. The performance analysis used to verify the neural network properties such as stability and convergence.

3.1 Wind Speed Prediction

Exact prediction of wind speed is one of the important issues in renewable energy systems because of dilute and fluctuating nature of wind. The wind has the uncertain irregularity characteristic. In order to achieve the better generalization capabilities for the wind speed forecasting the input and output properly modeled and the hidden neuron numbers should properly selected for the neural network design. The dynamic system may not achieve a possible solution due to the insufficient hidden neuron numbers in the neural network. Many researchers tried to developed different strategies to select the proper hidden neuron numbers in the neural network, but yet none was effective and right. In the current scenario a lot of forecasting research fields has been heuristic.

Small size network offers simple structure and better generalization ability, but it may not learn the issue well. While in the large size network learn easily, but slow and poor generalization performance due to over fitting [34]. The proposed model confirms that even though large hidden neuron numbers in the ELMAN neural network get stable performance on training. The greatest aims to fix the hidden neuron numbers in the ELMAN neural network for prediction of wind speed with better accuracy and minimal statistical error. Following error criteria used to select the optimal hidden neuron numbers in neural network. The proposed approach performance analyzed based on the Mean Absolute Error (MAE), Mean Relative Error (MRE) and Mean Square Error (MSE) error criteria. Among the different 75 criteria the best hidden neuron number selected based on the least error performance. The statistical error criteria formulas are defined as below:

$$\text{MAE}=\frac{1}{N}\sum_{i=1}^{N}(Y_i^{'}-Y_i)\ , \quad (1)$$

$$\text{MRE}=\frac{1}{N}\sum_{i=1}^{N}\left|(Y_i^{'}-Y_i)/\overline{Y_i}\right| \quad (2)$$

$$\text{MSE}=\frac{1}{N}\sum_{i=1}^{N}(Y_i^{'}-Y_i)^2 \quad (3)$$

Where Y_i is predicted output, $Y_i^{'}$ is actual output, $\overline{Y_i}$ is average actual output, N is number of samples. The neural network designing process plays a vital role in the network performance.

4 Proposed ELMAN Neural Network Architecture

In the literature review different heuristics exists, there is no hard and fast correct rule for estimation of hidden neuron numbers in the neural network design. Based on the knowledge gained from the existing approach propose the new criteria to select the optimal hidden neuron numbers in ELMAN neural network. The main aim is developed the ELMAN network model for wind speed prediction application and to estimate the optimal hidden neuron numbers in ELMAN network based on the different 75 criteria, where these criteria framed as a function of n (input neurons). The single proper topology is used.

4.1 ELMAN Neural Network Conceptual Overview

ELMAN neural network (ENN) is a partial recurrent network model first pointed out by ELMAN in 1990 [35]. ELMAN neural network consists of input layer, hidden layer, recurrent link (feedback) layer and output layer. The recurrent layer copies one step delay of hidden layer [31]. The ELMAN network is a recurrent network, recurrent links are added in to the hidden layer as a feedback connection. ELMAN network dynamic characteristics is provided by internal connections, ELMAN neural network superior to static feed-forward neural network because it does not require utilizing the states as training (or) input signal [36-38]. ELMAN neural network widely used for different application such as time series prediction, modeling, control and speech recognition. Output obtained from the hidden layer. The recurrent link layer stores the feedback and retains the memory. Hyperbolic tangent sigmoid activation function is adapted for hidden layer and purelin activation function used for output layer.

The proposed ELMAN network based wind speed prediction model possess inputs Temperature (TD_w), Wind direction (WD_w), Relative Humidity (RH_w), Precipitation of water content (PW_w) and Wind speed (N_w). As a result, five input neurons developed in the output layer. The output layer has a single output neuron (i.e.) predicted wind speed. Goal of the suggested approach is to select the proper hidden neuron numbers to get the faster convergence and better accuracy with minimal statistical error. Architecture of the proposed ELMAN network model to selecting the hidden neuron numbers is shown in Fig.1. Fig.1 inferred input and output target vector pair defined as follows.

$(X_1, X_2, X_3, X_4, X_5 : Y)$ = (Temperature, Wind direction, Relative Humidity, Precipitation of water content and Wind speed: Predicted wind speed).

$$\left(X_1, X_2, X_3, X_4, X_5 : Y\right) = \left(TD_w, WD_w, RH_w, PW_w, N_w : N_{pw}\right) \quad (4)$$

where, N_{pw} is the predicted wind speed, W_c be the weight between context layer and input layer, W_1 be the weight between input and hidden layer, W_2 be the weight between hidden and recurrent link layer, $h(\cdot)$ is hyperbolic tangent sigmoid activation function, $f(\cdot)$ is purelin activation function.

The layer performs the independent computations on received data and the computed results passed to the next layer and lastly determine the network output this information is inferred from the Fig.1. The response to the current input depends on the previous inputs. The input $U(K-1)$ passed through the hidden layer that multiplies weight (W_1) using hyperbolic tangent sigmoid activation function. The current input $W_1U(K-1)$ plus previous state output $W_cX_c(K)$ helps the network to learn the function. The value of $X(K)$ passed through output layer that multiplied with W_2 using purelin activation function. In network training, the previous state information reflects to the ELMAN neural network. The novel criteria used to select the hidden neuron numbers in network design and the proposed approach adopted for wind speed prediction application.

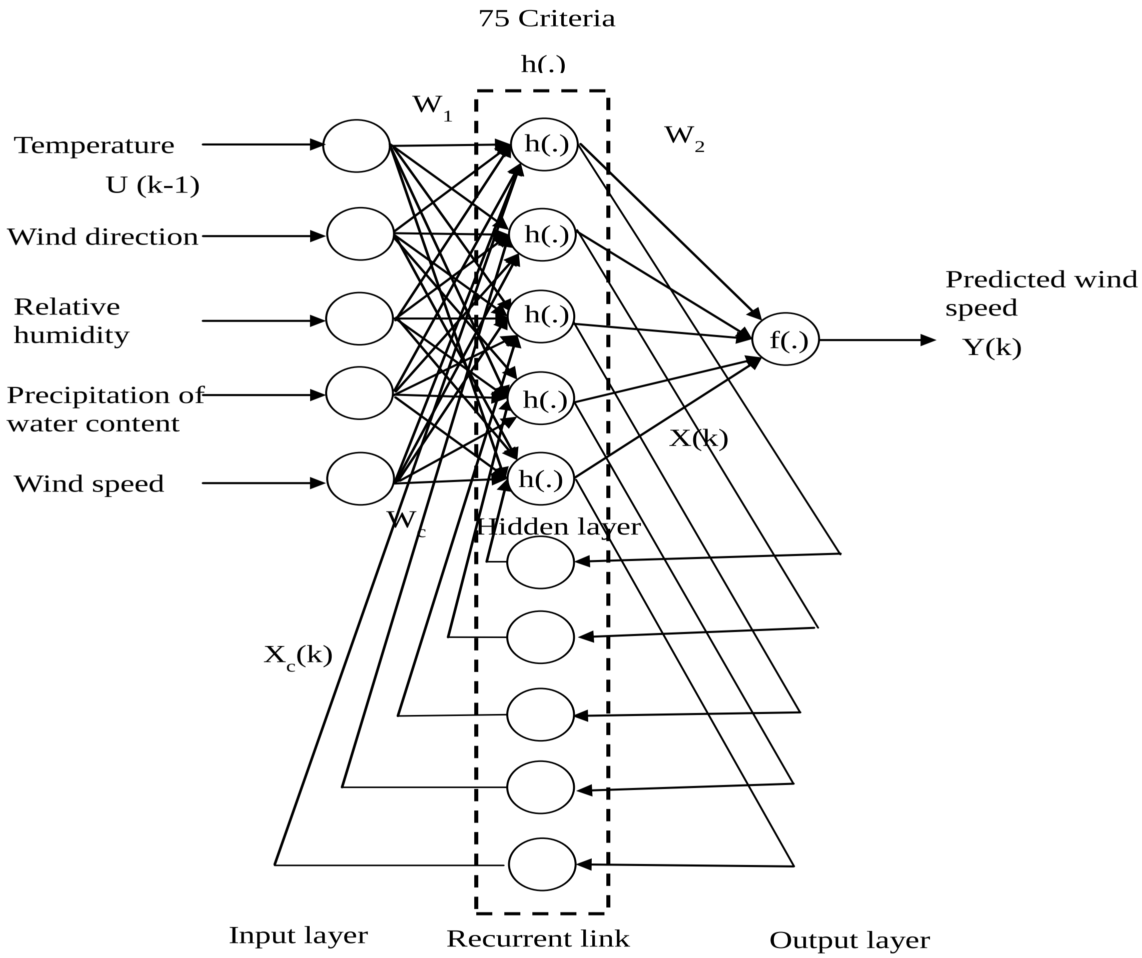

Fig.1. Architecture of the Proposed ELMAN network model to select the hidden neuron numbers.

Input vector, $X = \lfloor TD_w, WD_w, RH_w, PW_w, N_w \rfloor$ (5)

Output vector, $Y = \lfloor N_{pw} \rfloor$ (6)

Weight vectors of input to the hidden vector,

$$W_1 = \begin{bmatrix} W_{11}, W_{12}, .., W_{1n}, W_{21}, W_{22}, .., W_{2n}, W_{31}, W_{32}, \\ ., W_{3n}, W_{41}, W_{42}, .., W_{4n}, W_{51}, W_{52}, .., W_{5n} \end{bmatrix} \quad (7)$$

Weight vectors of recurrent link layer vector, $W_2 = [W_{21}, W_{22}, ..., W_{2n}]$ (8)

Weight vectors of recurrent link layer to input vector,

$$W_c = \begin{bmatrix} W_{c11}, W_{c12}, .., W_{c1n}, W_{c21}, W_{c22}, .., W_{c2n}, W_{c31}, W_{c32}, \\ ., W_{c3n}, W_{c41}, W_{c42}, .., W_{c4n}, W_{c51}, W_{c52}, .., W_{c5n} \end{bmatrix} \quad (9)$$

Input, $X(K) = h\left(W_c X_c(K) + W_1 U(K-1)\right)$ (10)

Output, $Y(K) = f\left(W_2 X(K)\right)$ (11)

Input of recurrent link layer,

$$X_c(K) = X(K-1) \quad (12)$$

4.2 Proposed Approach

The proposed approach fix the suitable hidden neuron numbers in ELMAN network for prediction of wind speed with better accuracy and faster convergence speed. The wind speed prediction using neural network involves the modeling, training and testing process. Exact neural network model design is complex and a challenging task. The input parameters used for the ELMAN neural network Temperature, Wind direction, Relative humidity, Precipitation of water content and Wind speed. The real-time data used as inputs so normalization process performed due to overcome training process problem of large valued input data tend to cut the effect of smaller value input data. Hence, the min-max scaling technique used to normalize the variable range of real-time data into the range of 0 to 1. The normalization process helps improve the numerical computational accuracy, so the ELMAN network model accuracy is also enhanced.

Selection of hidden neuron numbers in neural network is a challenging task, random selection of hidden neuron number cause the over fitting or under fitting problem. Therefore, propose the new 75 criteria function of input neurons to fix the hidden neuron numbers in neural network. Considered all 75 criteria verified using convergence theorem. The proper hidden neuron numbers in hidden layer selected based on the best lowest minimal error. Furthermore, the ELMAN neural network output (i.e.) wind speed predicted.

4.2.1. Data Collection

The real-time data collected from National Oceanic and Atmospheric Administration, United States from January 1948 to December 2013. The Temperature ($^{\circ}C$), Wind direction (Degree), Relative Humidity (%), Precipitation of water content (%) and Wind speed (m/s) inputs to the proposed ELMAN neural network and network output predicted wind speed (m/s). The proposed ELMAN model developed using $80{,}000$ numbers of samples. Table 1 shows the neural network model used input parameters. Table 2 shows the collected real-time data sample inputs.

4.2.2. Data Normalization

Scaling (or) Normalization is very important for dealing with real-time data; the real-time data has different range and different units. Hence, the normalization used to scale the real-time data within the range of 0 to 1. The normalization process helps make better correct numeric computation and improve the output accuracy. The proposed approach uses the min-max normalization technique. Following transformation equation used for normalization of the real-time data.

Scaled input,

$$X_i^{'} = \left(\frac{X_i - X_{\min}}{X_{\max} - X_{\min}}\right)\left(X_{\max}^{'} - X_{\min}^{'}\right) + X_{\min}^{'} \qquad (13)$$

where, X_i is actual input data, $X_{\min}$ is minimum input data, $X_{\max}$ is maximum input data, $X_{\min}^{'}$ is minimum target value, $X_{\max}^{'}$ is maximum target value.

4.2.3. Network Design

The proposed ELMAN network designed parameter includes dimensions and epochs shown in Table 3. The dimensions such as input neuron numbers, hidden neuron numbers, output neuron numbers defined in the network design. The presented network design has five input neuron (Temperature, Wind direction, Relative Humidity, Precipitation of water content and Wind speed), one hidden layer, one output neuron (Predicted wind speed) and the hidden neuron numbers in hidden layer selected from the proposed 75 various criteria based on the lowest minimal error. The inputs transferred to the hidden layer that multiplies weight W_1 using hyperbolic tangent sigmoid activation function and output from the hidden layer transferred to the output layer that multiplies with weight W_2 using purelin activation function. Training learned from the normalized data. As result of network training, earlier information reflected to ELMAN network. Test for the stopping condition, the error coming to a very negligible value.

4.2.4. Selection of Hidden Neurons

The considered 75 different criteria function of input neurons and it is verified on convergence theorem. The infinite sequence changed into finite sequence called convergence. The considered 75 different criteria met the convergence theorem requirement. All chosen criteria examined in ELMAN network to estimate the network training process and statistical errors.

The development of the proposed model initiated by adopting the chosen criteria to the ELMAN network. After the development process, train the Elman network and calculate the statistical errors. The selection of the proper hidden neuron number fixed based on the minimal statistical error. The equation (1-3) represents the statistical error formulas.

4.2.5. Training and Testing of the Network Performance

The wind speed prediction model developed based on training data while the performance of the proposed model evaluated by using the testing data. The collected $80{,}000$ real-time data classified in to training and testing. The collected 70% of data ($56{,}000$) used for training phase and 30% of the collected data ($24{,}000$) used for testing phase of the network. Considered 75 various criteria applied to ELMAN neural network one by one and verify the performance accepted (or) not. Network performance calculated based on the statistical error criteria such as MSE, MAE and MRE. The criteria with the lowest minimal statistical error are the best

Table 1. Proposed model input parameters.

S. No	**Input Parameter**	**Units**	**Range of Parameter**
1	Temperature	Degree. Celsius	23-30
2	Wind Direction	Degree	0.1-360
3	Relative Humidity	Percentage	53-94
4	Precipitation of water content	Percentage	19-51
5	Wind Speed	m/s	1-8

Table 2. Collected real-time inputs data sample.

Temperature ($°C$)	**Wind direction (Degree)**	**Relative Humidity ($\%$)**	**Precipitation of water content ($\%$)**	**Wind speed (m/s)**
26.9490	193.3204	82.3222	41.4056	5.7405
25.9666	191.9673	82.425	43.4056	7.4599
26.9539	357.9629	54.1871	23.1142	3.4599
23.51	193.5835	91.16	46.6103	5.9100
28.6538	235.2403	58.1148	34.0674	2.3899
25.4973	190.8898	85.8516	44.3883	4.75
28.6087	258.5871	61.9775	32.9267	2.4100
25.8013	190.0067	85.9016	47.0499	6.2099
28.5903	292.6652	61.1471	35.4674	1.8600
24.6519	173.7271	86.9213	45.0380	5

Table 3. Proposed ELMAN network designed parameters.

ELMAN Network	
Input neuron	= 5 $[TD_w, WD_w, RH_w, PW_w, N_w]$
Number of hidden layer	= 1
Output neuron	= 1 N_{pw}
Number of epochs	= 2000
Threshold	= 1

Table 4. ELMAN network Statistical analysis of different criteria for selecting hidden neuron numbers.

Hidden neuron numbers	Considered Criteria to select hidden neuron numbers	MSE	MAE	MRE
23	$(9n+1)/(n-3)$	0.0023	0.0347	0.0101
74	$(3n^2-1)/(n^2-24)$	0.0025	0.0345	0.0100
8	$(5(n^2+2)+1)/(n^2-8)$	0.0025	0.0386	0.0112
50	$10n/(n-4)$	0.0013	0.0275	0.0080
62	$(2(n^2+4)+4)/(n^2-24)$	7.1338e-04	0.0202	0.0059
17	$(9n+6)/(n-2)$	0.0012	0.0244	0.0071
45	$9n/(n-4)$	0.0011	0.0234	0.0068
53	$(10n+3)/(n-4)$	0.0013	0.0244	0.0071
68	$(2(n^2+6)+6)/(n^2-24)$	0.0023	0.0349	0.0102
5	$4n/(n-1)$	8.3664e-04	0.0210	0.0061
20	$4n/(n-4)$	0.0019	0.0329	0.0096
36	$(8n-4)/(n-4)$	0.0019	0.0290	0.0084
27	$(5n+2)/(n-4)$	0.0038	0.0413	0.0120
59	$(2(n^2+3)+3)/(n^2-24)$	7.5446e-04	0.0203	0.0059
14	$(8n+2)/(n-2)$	0.0011	0.0239	0.0070
72	$(2(n^2+9)+4)/(n^2-24)$	0.0020	0.0312	0.0091
29	$(5n+4)/(n-4)$	6.5705e-04	0.0179	0.0052
33	$(6n+3)/(n-4)$	0.0018	0.0300	0.0087
42	$(9n-3)/(n-4)$	9.5151e-04	0.0235	0.0068
3	$(3n-3)/(n-1)$	0.0012	0.0266	0.0078
65	$(2(n^2+7)+1)/(n^2-24)$	0.0034	0.0405	0.0118
38	$(8n-2)/(n-4)$	0.0016	0.0289	0.0084
55	$(10n+5)/(n-4)$	9.8635e-04	0.0206	0.0060
11	$(4n+2)/(n-3)$	0.0011	0.0243	0.0071
47	$(9n+2)/(n-4)$	0.0015	0.0282	0.0082

Table 4. Continued

Hidden neuron numbers	Considered Criteria to select hidden neuron numbers	MSE	MAE	MRE
63	$\left(2\left(n^2+6\right)+1\right)/\left(n^2-24\right)$	0.0017	0.0274	0.0080
49	$(9n+4)/(n-4)$	9.5936e-04	0.0247	0.0072
13	$(5n+1)/(n-3)$	0.0023	0.0332	0.0097
56	$\left(2\left(n^2+2\right)+2\right)/\left(n^2-24\right)$	0.0010	0.0236	0.0069
25	$5n/(n-4)$	0.0020	0.0302	0.0088
1	$n/(n+1)$	9.8456e-04	0.0242	0.0071
69	$\left(2\left(n^2+7\right)+5\right)/\left(n^2-24\right)$	0.0011	0.0217	0.0063
32	$(6n+2)/(n-4)$	0.0012	0.0254	0.0074
19	$(3n+4)/(n-4)$	0.0024	0.0356	0.0104
75	$\left(3\left(n^2+1\right)-3\right)/\left(n^2-24\right)$	0.0014	0.0282	0.0082
44	$(9n-1)/(n-4)$	0.0011	0.0226	0.0066
60	$\left(2\left(n^2+4\right)+2\right)/\left(n^2-24\right)$	0.0013	0.0288	0.0084
7	$(4n+1)/(n-2)$	0.0016	0.0277	0.0081
35	$7n/(n-4)$	0.0019	0.0292	0.0085
52	$\left(2n^2+2\right)/\left(n^2-24\right)$	6.1873e-04	0.0181	0.0053
66	$\left(2\left(n^2+6\right)+4\right)/\left(n^2-24\right)$	0.0015	0.0278	0.0081
10	$(5n+5)/(n-2)$	0.0013	0.0264	0.0077
46	$(8n+6)/(n-4)$	0.0037	0.0317	0.0108
21	$(8n+2)/(n-3)$	0.0023	0.0317	0.0092
30	$6n/(n-4)$	0.0017	0.0304	0.0089
4	$(n+3)/(n-3)$	5.5878e-04	0.0176	0.0051
39	$(7n+4)/(n-4)$	0.0014	0.0251	0.0073
58	$\left(2\left(n^2+2\right)+4\right)/\left(n^2-24\right)$	0.0017	0.0313	0.0091
71	$\left(2\left(n^2+8\right)+5\right)/\left(n^2-24\right)$	0.0014	0.0278	0.0081
16	$(8n+8)/(n-2)$	0.0015	0.0303	0.0088

Table 4. Continued

Hidden neuron numbers	Considered Criteria to select hidden neuron numbers	MSE	MAE	MRE
15	$9n/(n-2)$	0.0015	0.0270	0.0079
73	$(2(n^2+8)+7)/(n^2-24)$	0.0012	0.0252	0.0073
2	$(2n-2)/(n-1)$	0.0012	0.0264	0.0077
40	$8n/(n-4)$	0.0011	0.0223	0.0065
31	$(5n+6)/(n-4)$	8.0590e-04	0.0206	0.0060
24	$(8n+8)/(n-3)$	0.0021	0.0329	0.0096
48	$(10n-2)/(n-4)$	6.4270e-04	0.0156	0.0045
54	$(2n^2+4)/(n^2-24)$	7.0377e-04	0.0195	0.0057
67	$(2(n^2+7)+3)/(n^2-24)$	0.0011	0.0230	0.0067
26	$(4n+6)/(n-4)$	0.0023	0.0329	0.0096
34	$(7n-1)/(n-4)$	0.0013	0.0247	0.0072
18	$(3n+3)/(n-4)$	8.2323e-04	0.0220	0.0064
51	$(10n+1)/(n-4)$	9.1362e-04	0.0208	0.0061
6	$(4n^2+2)/(n^2-8)$	0.0012	0.0238	0.0069
37	$(7n+2)/(n-4)$	0.0015	0.0282	0.0082
22	$(4n+2)/(n-4)$	0.0017	0.0277	0.0081
61	$(2(n^2+5)+1)/(n^2-24)$	0.0022	0.0328	0.0096
43	$(8n+3)/(n-4)$	**3.6994e-04**	**0.0149**	**0.0043**
70	$(2(n^2+9)+2)/(n^2-24)$	9.5941e-04	0.0214	0.0062
12	$(7(n^2+4)+1)/(n^2-8)$	8.5259e-04	0.0205	0.0060
28	$(6n-2)/(n-4)$	9.9541e-04	0.0228	0.0066
57	$(2(n^2+3)+1)/(n^2-24)$	0.0026	0.0315	0.0092
9	$(5(n^2+6)-2)/(n^2-8)$	9.9395e-04	0.0234	0.0068
41	$(7n+6)/(n-4)$	0.0023	0.0338	0.0098
64	$(2(n^2+5)+4)/(n^2-24)$	0.0014	0.0264	0.0077

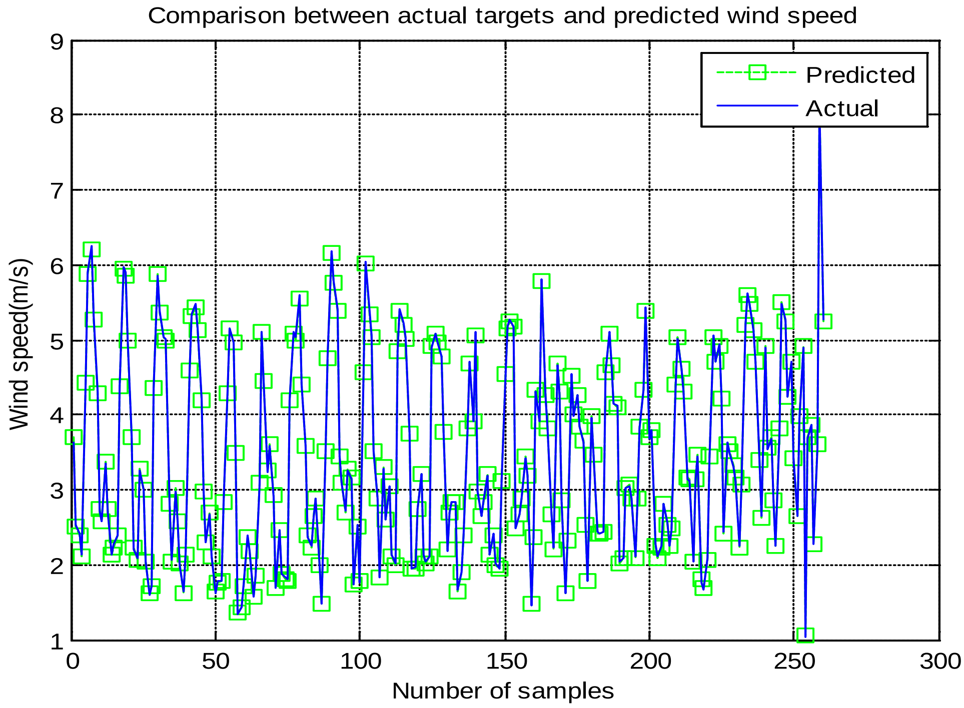

Fig.2. Comparison between actual and predicted wind speed.

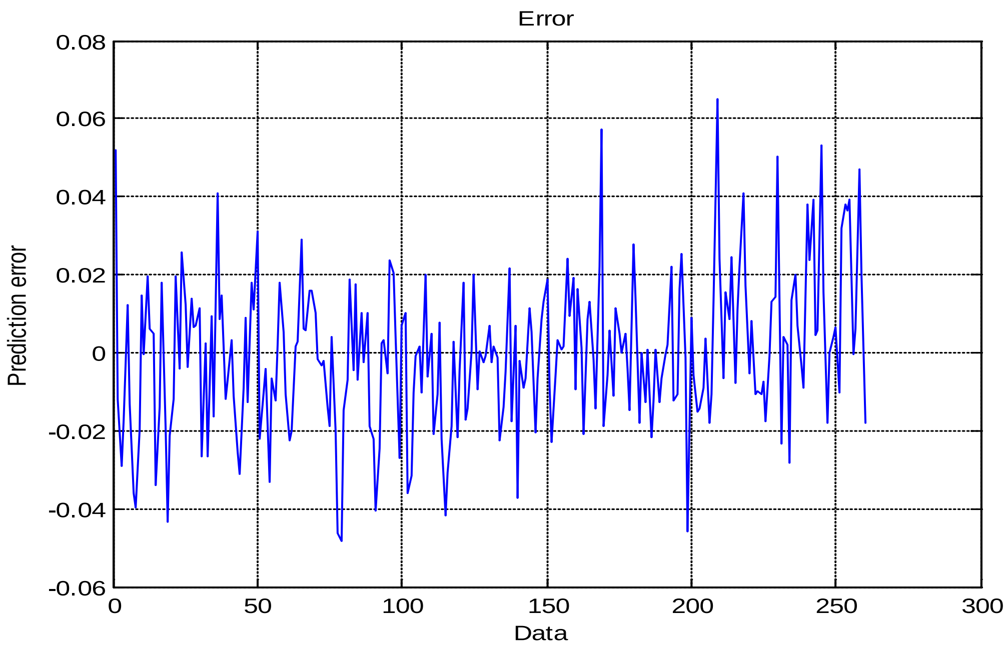

Fig.3. Prediction error.

criteria for selection of hidden neuron in ELMAN neural network.

4.3. Validation for the Selected Proposed Criteria

The proof for the fixation of hidden neuron numbers initiated based on the convergence theorem discussion in the Appendix.
Lemma 1 is confirms the convergence of the presented selection criteria.
Lemma 1: The sequence $u_n = [(8n+3)/(n-4)]$ converged and $u_n \geq 0$.
The sequence tends to have a finite limit l, if there exists constant $\varepsilon > 0$ such that $|u_n - l| < \varepsilon$, then $\lim_{n \to \infty} u_n = l$.
Proof:
Lemma 1 based proof defined as follows
Regarding to convergence theorem, the selected value (or) sequence converges to finite limit value.

$$u_n = \frac{(8n+3)}{n-4}, \tag{14}$$

$$\lim_{n \to \infty} \frac{(8n+3)}{n-4} = \lim_{n \to \infty} \frac{n\left[\left(8+\frac{3}{n}\right)\right]}{n\left(1-\frac{4}{n}\right)} = 8 \text{, finite limit value.} \tag{15}$$

Here 8 is limit value of the selected sequence as $\lim n \to \infty$.

Hence, above sequence convergent sequence because it has the finite limit. Where n is the number of inputs parameter.

5 Results and Discussion

Several researchers suggested a lot of strategies to select the hidden neuron numbers in neural network. The strategies classified in to pruning and constructive strategies. In the pruning strategy the network starts with oversize and then prunes the minimum relevant neuron and weights search the minimum size, while in the constructive strategy network starts with undersized network and then additional hidden neuron added to the network [11, 39]. The problems of proper hidden neuron numbers for a particular problem selected. The earlier approaches use trial and error rule to decide hidden neuron numbers in neural network. This starts the network with undersized hidden neuron numbers and neurons added to N_h. Demerits of the earlier methods are there are no guarantees of selecting the number of hidden neuron and it consumes much time. The proposed criteria considered in this paper used to build three layer neural networks. The proposed neural network design were run on an Acer laptop computer with Pentium (R) Dual Core processor running at 2.30GHZ with 2GB of RAM. Network performances are evaluated by statistical error calculation. The real-time data initially classified into the training and testing sets. Training set used in neural network learning, and testing set used to calculate the error.

5.1. Analysis of computed wind speed and error factors employing proposed method

The experimental results confirm with minimal error determined as the best solution for selecting hidden neuron numbers in neural network model. Considered 75 different criteria for selection of hidden neuron numbers in neural network based on the statistical error is established in Table. 4. Simulation results proved that the predicted wind speed is in the best agreement with the experimental measured values. The chosen criteria for neural network design is $u_n = [(8n+3)/(n-4)]$ made a minimal MSE of $3.6994e-04$, MAE of 0.0149 and MRE of 0.0043. It has noticed that the error values minimal compared to other criteria. Therefore, the proposed criteria improve the effectiveness and accuracy for prediction of wind speed. For the clarity of proposed model, the comparison between actual and predicted wind speed for 240 samples is shown in Fig.2. Prediction error is noticed from Fig.3. Errors vs. hidden neuron numbers are depicted in Fig.4. The merits of the presented approach very effective, minimal error and simple implementation for wind speed prediction.
The chosen criteria for neural network model wind speed prediction application is $(8n+3)/(n-4)$ with 43 hidden neuron numbers and achieved a minimal mean square error (MSE) value of $3.6994e-04$ in comparison with other criteria. Results revealed that the proposed approach meet the better results than that of the other existing approaches. The performance analysis of earlier approaches and proposed method is depicted in Table 5. Table 5. infers that compared to other existing model the suggested model make better minimal statistical error.

Table 5. Comparative analysis of different approaches performance in existing and proposed method.

S. No	Different Approaches	Year	Hidden Neuron Numbers	Statistical Error (MSE)
1	Arai Approach	1993	$N_h = 2^n / 3$	0.0011
2	Jin-Yan Li et al. Approach	1995	$N_h = \left(\sqrt{1+8n-1}\right)/2$	0.0012
3	Tamura & Tateishi, M Approach	1997	$N_h = N - 1$	5.5878e-04
4	Osamu Fujita Approach	1998	$N = K \log\left(\left\|P_v^{(0)} Z\right\| / C\right) / \log S$	0.0025
5	Zhaozhi Zhang et al. Approach	2003	$N_h = 2^n / (n+1)$	8.3994e-04
6	Jinchauan Ke & Xinzhe Lie Approach	2008	$N_h = \left(N_{in} + \sqrt{N_p} / L\right)$	0.0018
7	Shuxiang Xu & Ling Chen Approach	2008	$N_h = C_f \left(N / (d \log N)\right)^{1/2}$	0.0023
8	Stephen Trenn Approach	2008	$N_h = (n + n_0 - 1)/2$	0.0012
9	Katsunari Shibata & Yusuke Ikeda Approach	2009	$N_h = \sqrt{N^{(i)} N^{(0)}}$	0.0012
10	David Hunter et al .Approach	2012	$N_h = 2^n - 1$	8.0590e-04
11	Gnana Sheela, K & S, N, Deepa Approach	2013	$N_h = (4n^2 + 3)/(n^2 - 8)$	0.0012
12	Gue Qian & Hao Yong Approach	2013	$N_h = \sqrt{n + m + a}$	5.5878e-04
13	Proposed Approach		$N_h = (8n+3)/(n-4)$	**3.6994e-04**

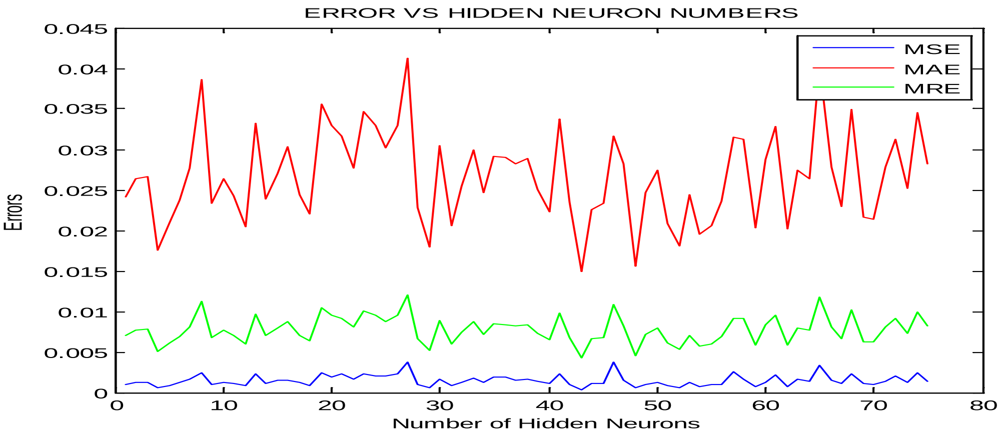

Fig 4. Errors vs. Hidden neuron numbers.

6 Conclusion

In this paper a comparative analysis has performed for the selection of hidden neuron numbers in neural networks design. The suggested approach adopted and tested with real-time wind data. The main goal of the proposed methodology to fix the exact hidden neuron numbers in ELMAN neural network for wind speed prediction application and met a better performance compared to other existing approaches. The statistical errors used to analyze the better performance. According to this paper, the following conclusions are inferred.

1) Analyzes various methods to select the hidden neuron numbers in neural network.

2) Proposed novel method for fixing the hidden neuron numbers thus achieving better framework for ELMAN neural network design and make accurate wind speed prediction with minimal error.

APPENDIX

Considers different criteria 'n' as input parameters number. All considered criteria satisfied based on the convergence theorem. Some explanations are given below. If the limit of sequence is finite, the sequence called convergent sequence. If the limit of a sequence does not tend to a finite number, the sequence called divergent [40].

The convergence theorem characteristics are given below.

1. Convergent sequence needed condition is that it has finite limit and bounded.

2. An oscillatory sequence does not tend to have a unique limit.

In a network there no change occurring in the state of the network regardless of the operation called the stable network. In the neural network model most important property is it is always converges to a stable state. In real-time optimization problem the convergence play a major role, risk of getting stuck at some local mini ma problem in a network prevented by the convergence. Convergence of sequence infinite has been established in convergence theorem because of the discontinuities in model. The real-time neural optimization solvers designed by the use of convergence properties.

Discuss convergence of the considered sequence as follows.

Taking the sequence $u_n = \frac{3n+3}{n-4}$ (A.1)

Apply convergence theorem

$$\lim_{n\to\infty} u_n = \lim_{n\to\infty} \frac{3n+3}{n-4} = \lim_{n\to\infty} \frac{n\left(3+3/n\right)}{n\left(1-4/n\right)} = 3 \neq 0$$

, it has finite value.

(A.2)

Hence, the terms of sequence has a finite limit value and bounded so the considered sequence convergent sequence.

Take the sequence $u_n = \frac{6n}{n-4}$ (A.3)

Apply convergence theorem

$$\lim_{n\to\infty} u_n = \lim_{n\to\infty} \frac{6n}{n-4} = \lim_{n\to\infty} \frac{6n}{n\left(1-4/n\right)} = 6 \neq 0$$

, it has finite value.

(A.4)

Hence, the terms of sequence has a finite limit value and bounded so the considered sequence convergent sequence.

Acknowledgment

The authors are thankful to National Oceanic and Atmospheric Administration, United States providing real-time observations, to do the research work.

References:

[1] S. N. Sivanandam, S. Sumathi, S. N. Deepa, *Introduction to Neural Networks using Matlab 6.0*, Tata McGraw Hill, 2008.

[2] Gaurang Panchal, Amit Ganatra, Y. P. Kosta and Devyani Panchal, Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers, *International*

Journal of Computer Theory and Engineering, Vol.3, No.2, 2011, pp. 332-337.

[3] David Hunter, Hao Yu, Michael. S. Pukish III, Janusz Kolbusz, Bogdon. M. Wilamowski, Selection of proper neural network sizes and architecture- A comparative study, *IEEE Transactions on Industrial Informatics*, Vol.8, No.2, 2012, pp. 228-240.

[4] Shuxiang Xu, Ling Chen, A novel approach for determining the optimal number of hidden layer neurons for FNN's and its application in data mining, *5th International Conference on Information Technology and Application (ICITA)*, 2008, pp. 683-686.

[5] Jinchuan Ke, Xinzhe Liu, Empirical analysis of optimal hidden neurons in neural network modeling for stock prediction, *Pacific, Asia workshop on Computational Intelligence and Industrial Application*, Vol.2, 2008, pp. 828-832.

[6] Saurabh Karsoliya, Approximating number of hidden layer neuron in multiple hidden layer BPNN architecture, *International Journal of Engineering Trends and Technology*, Vol.31, No.6, 2012, pp. 714-717.

[7] Shih-Chi Huang, Yih-Fang Huang, Bounds on the number of hidden neurons in multilayer perceptrons, *IEEE Transaction on Neural Network*, Vol.2, No.1, 1991, pp. 47- 55.

[8] Arai. M, Bounds on the number of hidden units in binary-valued three-layer neural networks, *Neural Networks*, Vol.6, 1993, pp. 855-860.

[9] Masafumi Hagiwara, A simple and effective method for removal of hidden units and weights, *Neuro Computing*, Vol. 6, No.2, 1994, pp. 207-218.

[10] Noboru Murata, Shuji Yoshizawa, Shun-Ichi Amari, Network Information Criterion determining the number of hidden units for an artificial neural network model, *IEEE Transaction on Neural Networks*, Vol.5, No.6, 1994, pp. 865-872.

[11] Jin-Yan Li, Chow. T. W. S, Ying-Lin Yu, The estimation theory and optimization algorithm for the number of hidden units in the higher-order feed forward neural network, *Proceeding IEEE International Conference on Neural Networks*, Vol.3, 1995, pp. 1229-1233.

[12] Onoda. T, Neural network information criterion for the optimal number of hidden units, *Proceeding IEEE International Conference on neural networks*, Vol.1, 1995, pp. 275-280.

[13] Tamura. S, Tateishi. M, Capabilities of a four-layered feed forward neural network: four layer versus three, *IEEE Transaction on neural network*, Vol. 8, No.2, 1997, pp. 251-255.

[14] Osamu Fujita, Statistical estimation of the number of hidden units for feed forward neural network, *Neural Network*, Vol.11, 1998, pp. 851-859.

[15] Keeni. K, Nakayama. K and Shimodaira. H, Estimation of initial weights and hidden units for fast learning of multilayer neural networks for pattern classification, *International Joint Conference on Neural Networks*, Vol.3, 1999, pp. 1652-1656.

[16] Guang-Bin Huang, Learning capability and storage capacity of two-hidden layer feed forward networks, *IEEE Transactions on Neural Networks*, Vol.14, No.2, 2003, pp. 274-281.

[17] H. C. Yuan, F. L. Xiong and X. Y. Huai, A method for estimating the number of hidden neurons in feed-forward neural networks based on information entropy, *Computers and Electronics in Agriculture*, Vol.40, 2003, pp. 57-64.

[18] Zhaozhi Zhang, Xiaomin Ma, Yixian Yang, Bounds on the number of hidden neurons in three-layer binary neural networks, *Neural Network*, Vol.16, 2003, pp. 995-1002.

[19] K. Z. Mao, Guang-Bin Huang, Neuron selection for RBF neural network classifier based on data structure preserving criterion, *IEEE Transactions on Neural Networks*, Vol.16, No.6, 2005, pp. 1531-1540.

[20] E. J. Teoh, K. C. Tan, C. Xiang, Estimating the number of hidden neurons in a feed forward network using the singular value decomposition, *IEEE Transactions on Neural Networks*, Vol.17, No.6, 2006, pp.1623-1629.

[21] Xiaoqin Zeng, Daniel S. Yeung, Hidden neuron pruning of multilayer perceptrons using a quantified sensitivity measure, *Neuro Computing*, Vol.69, 2006, pp. 825-837.

[22] Bumghi Choi, Ju-Hong Lee, Deok-Hwan Kim, Solving local minima problem with large number of hidden nodes on two layered feed forward artificial neural networks, *Neuro Computing*, Vol.71, No. 16-18, 2008, pp. 3640-3643.

[23] Min Han, Jia Yin, The hidden neurons selection of the wavelet networks using support vector machines and ridge regression, *Neuro Computing*, Vol.72, No.1-3, 2008, pp. 471-479.

[24] Nan Jiang, Zhaozhi Zhang, Xiaomin Ma, Jian Wang, The lower bound on the number of hidden neurons in multi-valued multi threshold neural networks, *Second International Symposium on Intelligent Information Technology Application*, Vol.1, 2008, pp.103-107.

[25] Stephen Trenn, Multilayer perceptrons: Approximation order and necessary number of hidden units, *IEEE Transactions on Neural Networks*, Vol. 19, No.5, 2008, pp. 836-844.

[26] Katsunari Shibata, Yusuke Ikeda, Effect of number of hidden neurons on learning in large-scale layered neural networks, *ICROS-SICE International Joint Conference*, 2009, pp. 5008-5013.

[27] Doukin. C. A,Dargham. J. A, Chekima. A, Finding the number of hidden neurons for an MLP neural network using coarse to fine search technique. *10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA),* 2010, pp. 606-609.

[28] Junfang Li, Buhan Zhang, Chengxiong Mao, Guang Long Xie, Yan Li, Jiming Lu, Wind speed prediction based on the Elman recursion neural networks, *International Conference on Modelling, Identiication and Control*, 2010, pp. 728-732.

[29] Jianye Sun, Learning algorithm and hidden node selection scheme for local coupled feed forward neural network classifier, *Neuro Computing*, Vol.79, 2012, pp. 158-163.

[30] Ramadevi. R, Sheela Rani. B, Prakash. V, Role of hidden neurons in an Elman recurrent neural network in classification of cavitation signals, *International Journal of Computer Application*, Vol.37, No.7, 2012, pp. 9-13.

[31] K. Gnana Sheela, S. N. Deepa, Review on methods to fix number of hidden neurons in neural networks, *Mathematical Problems in Engineering*, Vol.2013, 2013, pp.1-11.

[32] Guo Qian, Hao Yong, Forecasting the rural per capita living consumption based on Matlab BP neural network, *International Journal of Business and Social Science*, Vol.4, No.17, 2013, pp.131-137.

[33] KuldipVora, ShrutiYagnik, A new technique to solve local minima problem with large number of hidden nodes on feed forward neural network, *International Journal of Engineering Development and Research*, Vol.2, No.2, 2014, pp.1978-1981.

[34] Siddhaling Urolagin, K. V. Prema, N. V. Subba Reddy, Generalization capability of artificial neural network incorporated with pruning method, *Lecture Notes in Computer Science*, Vol.7135, 2012, pp. 171-178.

[35] Jeffrey L. Elman, Finding structure in time. *Cognitive Science*, Vol.14, 1990, pp. 179-211.

[36] Lin. F. J, Hung. Y. C, FPGA-based Elman neural network control system for linear ultrasonic motor, *IEEE Trans ultras on Ferro electr Freq Control*, Vol.56, No.1, 2009, pp. 101-113.

[37] Liu Hongmei, Wang Shaoping, Ouyang Pingchao, Fault diagnosis based on improved Elman neural network for a hydraulic servo system, *Proceeding International Conference on Robotics, Automation Mechatronics*, 2006, pp. 1-6.

[38] Xiang Li, Guanrong Chen, Zengqiang Chen, Zhuzhi Yuan, Chaotifying linear Elman networks, *IEEE Transaction neural network*, Vol.13, No.5, 2002, pp. 1193-1199.

[39] Morris. A. J, Zhang. J, A sequential learning approach for single hidden layer neural network, *Neural Networks*, Vol.11, No.1, 1998, pp. 65-80.

[40] H. K. Dass, *Advanced Engineering Mathematics*, S. CHAND & Company Ltd, First edition 1988, reprint 2009.