Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 5, May 2015, pg.358 – 364

RESEARCH ARTICLE

Optimizing Number of Hidden Nodes for Artificial Neural Network using Competitive Learning Approach

Foram Panchal[1], Mahesh Panchal[2]

[1]Department of Computer Science & Engineering, Kalol Institute of Technology, Gujarat, India

[2]Deputy Director at GTU, Gujarat, India

[1] forampanchal87@gmail.com, [2] mkhpanchal@gmail.com

Abstract – Artificial Neural Network is appropriate and most effective for pattern recognition, signal processing, and classification problems. For getting the proper result, ANN requires accurate data preprocessing, architecture selection, network training. The major problem in ANN is to select the number of hidden nodes. The random selection of number of hidden nodes leads towards the problem of Overfitting or Underfitting. The Competitive Learning Approach works on real world data to train the ANN and finds the optimized number of hidden layer. The proposed method calculates the number of hidden layer on the basis of input data also numbers of hidden nodes are dynamic as they are generated while training the ANN.

Keyword– Artificial Neural Network (ANN), Competitive learning approach, Hidden nodes

I. INTRODUCTION

Artificial Neural Network is appropriate and most effective for pattern recognition, signal processing, and classification problems. For proper result ANN requires correct preprocessing of data, selection of architecture and training for the network. But the major problem is to select the number of hidden nodes, because the random selection of number of hidden nodes may cause the problem of Overfitting and Underfitting.Sometimes the network matches the data narrowly and loses its generalization ability over the test data which give rise to overtraining. This paper gives information regarding the competitive learning approach of finding hidden nodes for artificial neural network. The advantage of proposed method is that it is not approximately calculating number of hidden nodes but based on similarity between input data. In proposed method number of hidden nodes are not predefined but generated during the

training time. The organization of this paper is, in section II various methods for finding number of hidden nodes are explain. Section III contains proposed method to finding number of hidden nodes. Section IV is about experiments and its result. Last section contains the conclusion about the study that is done so far and so forth future work.

II. VARIOUS METHODS TO FIND NUMBER OF HIDDEN NODES

A. Structured trial and error method[1]

The characteristics of this method is to do repeated attempts until it successes or until the last attempt of agent. This method is dividing in to two approaches.

1. Forward Approach

This method works with bottom to top approach. Initially we take lower number of hidden neurons to train and test the NN. Then we train and test NN by increasing number of hidden neurons step by step. This process is repeated until it gives improved result for training and testing the NN.

2. Backward Approach

This method works with top to bottom approach. Initially we take big number of hidden neurons to train and test the NN. Then we train and test NN by decreasing number of hidden neurons step by step. This process is repeated until it gives improved result for training and testing the NN.

B. Two phase method[1][5]

The characteristic of this method is to do repeated attempts until it successes or until the last attempt of agent, which is similar to above method but the approach is different. Total 4 groups are created from the dataset. In the first phase two of the groups are used to train the NN. During the second phase the third group is used to test the NN.Output values from the trained NN are determined by using the remaining fourth group. Leastpossible error term regarding number of neurons in the hidden layer is obtained by repeating the process.

C. Rule of Thumb Methods[2]

The number of neurons to be used in hidden layer is determined by this method as described below:

- The number of hidden neurons should not be greater than the size of the input layer and as well as be less than the output layer.

- Suppose the size of the input layer is 8 and the size of the output layer is 4 the hidden neurons are in range between 4 and 8.

- The number of hidden neurons should be summation of 2/3 of the input layer size and the size of the output layer.
  - If we have input neurons 9 and output neurons 2 then we have 8 hidden neurons.
- The number of hidden neurons should not be more than double the input layer size.
  - If we have input neurons 6 then we have hidden neurons less than 12.

D. Method based on some random criteria[3]

- Data collection :
  - Real time data was collected from Suzlon Energy India Ltd., India wind farm for a period from April 2011 to Dec. 2012.
- Data Normalization :
  - Normalization of data is essential as the variables of different units.
  - Data are scaled within 0 to 1.
  - Scaling will result in improved accuracy.
- Designing the network :
  - Set-up parameter includes epochs and dimensions.
  - The dimensions like number of input, hidden, and output neuron are to be designed.
  - The three input parameters are temperature, wind direction and wind speed.
- Selection of proposed criteria :
  - For the proposed model, 101 various criteria were examined to estimate training process and errors
- Selection of proposed criteria :
  - For the proposed model, 101 various criteria were examined to estimate training process and errors.

Considered criteria for fixing number of hidden neurons	no. of hidden neurons	MSE	MRE	MAE
$5n + 4/n - 2$	19	0.0771	0.0125	0.101
$5(n^2 + 2) + 4/n^2 - 8$	59	0.0422	0.0101	0.0815
$5(n^2 + 5)/n^2 - 8$	70	0.0366	0.0083	0.067
$4.5n/(n - 1)$	7	0.0727	0.0161	0.1301
$5n^2 + 3/n^2 - 8$	48	0.0183	0.0118	0.0958
$6(n^2 + 5) - 1/n^2 - 8$	83	0.0407	0.0092	0.0747
$5(n^2 + 6) - 2/n^2 - 8$	73	0.0493	0.0089	0.0721
$5n + 5/n - 2$	20	0.142	0.0157	0.1271
$7(n^2 + 4) + 2/n^2 - 8$	93	0.0133	0.0093	0.0753
$7(n^2 + 5) + 1/n^2 - 8$	99	0.0377	0.012	0.0971

Table 1 Criteria for fixing number of hidden nodes

- Divide the data into training and testing :
  - 7000 data was used for training, and 3000 data was used for testing data.

III. METHODOLOGY

In this section, methods used for proposed work is discussed in detail. Expected outcomes after applying proposed methods are included along with implementation strategies.

A. Competitive approach

Competitive approach results in win-lose consequences as it works similar to the process of distributive negotiation.

B. Flow of Work

1. Initialize number of input nodes.
2. Initialize similarity threshold.
3. Now read a sample from training data.
4. If it is first sample then add one hidden node and output node.
5. X = sqrt (sum (square(weights between input-hidden nodes)))
   Y = sqrt (sum (square (values of new sample)))
6. Now find the difference of X and Y.
7. If difference is greater than threshold value there is no need to create hidden node otherwise create hidden node.
8. Now assign weight to new hidden node from input-hidden nodes as input values.
9. Now check class of new sample if it is similar to network class then no need to create new output node otherwise create output node.

10. Now assign weight to new output node from hidden-output nodes as output values.
11. Repeat steps 3 to 10.
12. As a final result we get the number of hidden nodes necessary to train the NN.

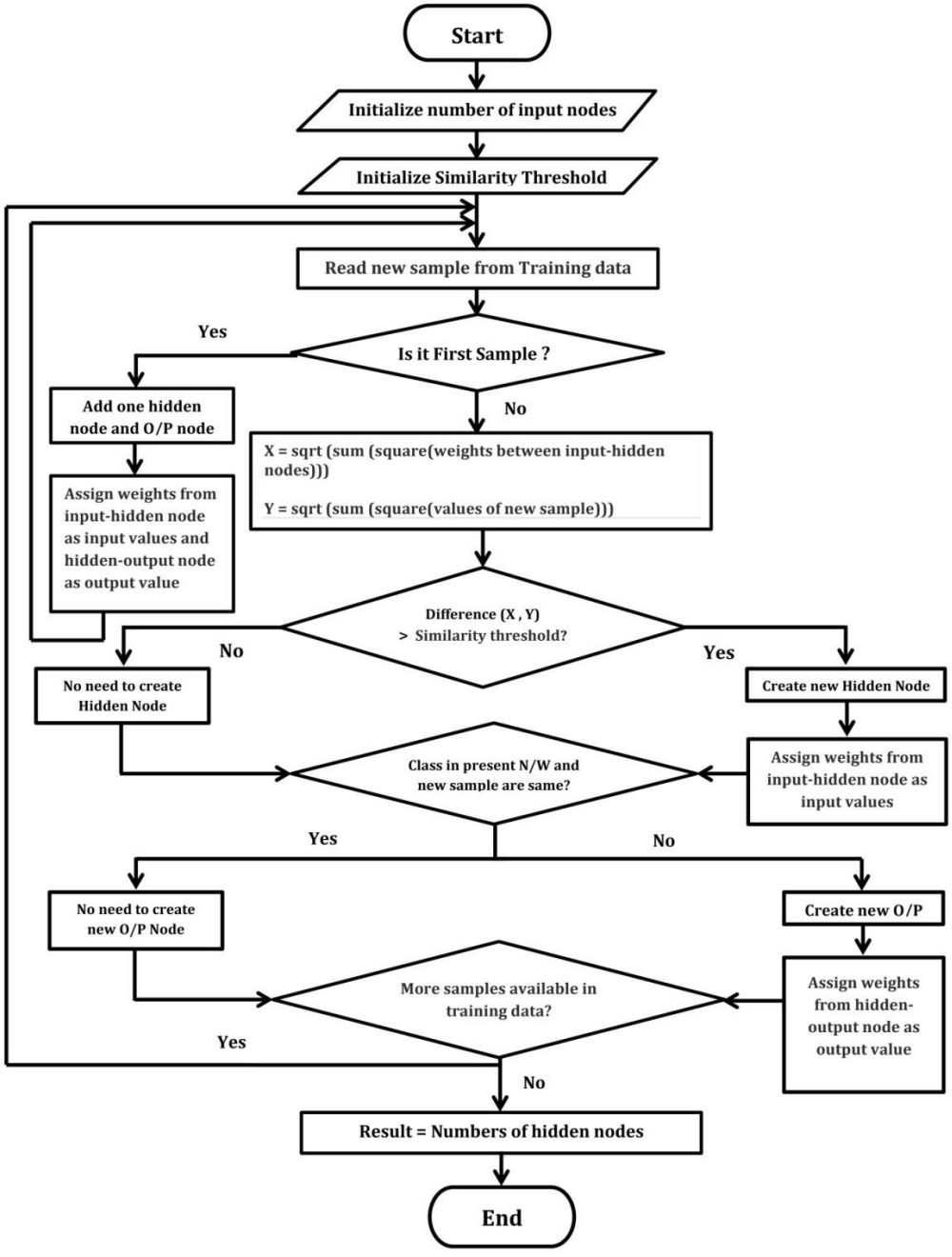
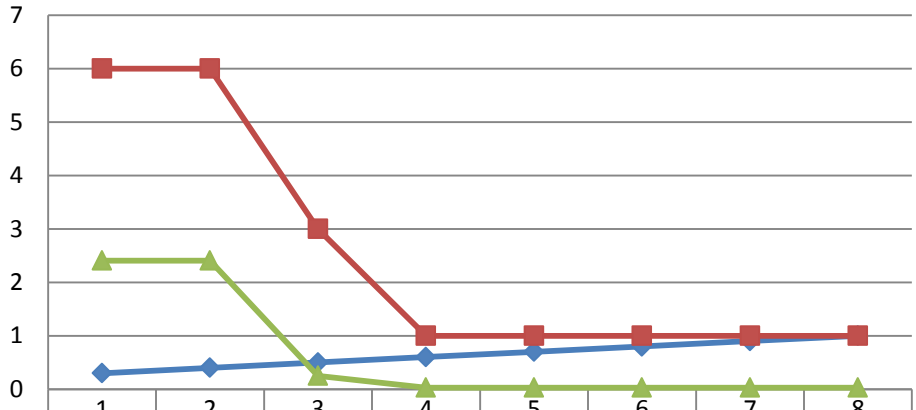


Figure 1 Flow Chart

Figure 1 Flow chart of Competitive Approach

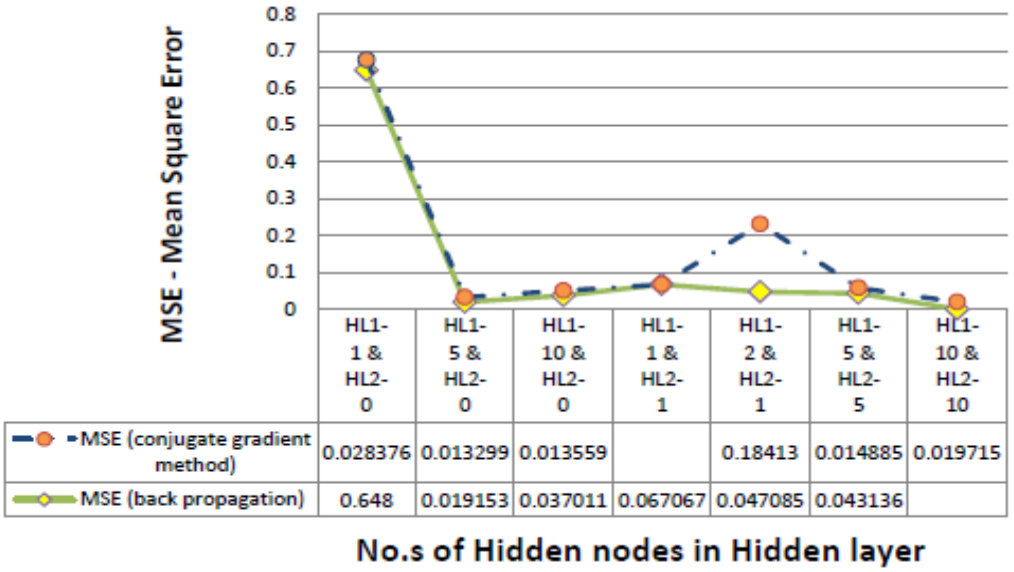
IV. EXPERIMENTS AND RESULT

For the experimental purpose Sales Forecasting data set with 6 inputs and 2 outputs has been taken. Proposed method (Competitive Learning Approach) has performed on this dataset also Backpropagation with momentum and Conjugate Gradient Method perform on the dataset for comparison purpose.



Similarity Threshold	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
No.of Hidden nodes	6	6	3	1	1	1	1	1
MSE	2.4076	2.4076	0.25143	0.03051	0.03051	0.03051	0.03051	0.03051

Figure - 2 Sales forecasting using Competitive Learning Approach



	HL1-1 & HL2-0	HL1-5 & HL2-0	HL1-10 & HL2-0	HL1-1 & HL2-1	HL1-2 & HL2-1	HL1-5 & HL2-5	HL1-10 & HL2-10
MSE (conjugate gradient method)	0.028376	0.013299	0.013559		0.18413	0.014885	0.019715
MSE (back propagation)	0.648	0.019153	0.037011	0.067067	0.047085	0.043136	

Figure - 3Sales forecasting using Backpropagation and Conjugate Gradient [4]

From the figure – 2 as we increased the similarity threshold value, number of hidden nodes decreased also the MSE is decreased according to the number of hidden nodes decreased and similarity threshold increased.

V. CONCLUSION

From the above experiment we conclude that as we increased similarity threshold, no of hidden node are also decreased and we get better MSE (min square error). We also conclude that all the previous methods to find number of hidden nodes are based on approximation, also we have to decide the number of hidden nodes before training the neural network. But proposed method is not based on approximation rather it gives exact solution. Also there is no need to decide number of hidden node before training of ANN. In proposed method number of hidden nodes is generated automatically during the training of ANN, based on the value of similarity threshold.

REFERENCES

[1] Saurabh Karsoliya, "Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture", *International Journal of Engineering Trends and Technology- Volume 3 Issue 6- 2012*

[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, April 2011, ISSN: 1793-8201*

[3] K. Ghana Sheila and S. N. Deepak, "Review on Methods to Fix Number of Hidden Neurons in Neural networks," *Mathematical Problems in Engineering, vol. 2013, Article ID 425740, 11 pages, 2013. doi:10.1155/2013/425740*

[4] Foram Panchal and Mahesh Panchal, "Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network,"*International Journal of Computer Science and Mobile Computing, Vol.3, Issue.11, Nov 2014, pages 455-464*

[5] Blum, A., 1992, Neural Networks in C++,NY:Wiley

