International Journal of Computer and Electrical Engineering, Vol.3, No.1, February, 2011
1793-8163

Improved Adaptive Learning Algorithm for Constructive Neural Networks

S.S.Sridhar, *Member, IACSIT & IEEE* and M.Ponnavaikko, *Senior Member IEEE*

Abstract—**Constructive Neural Network learning algorithms provide incremental ways to determine the near-minimal architecture of a multi layer perceptron network along with learning algorithms for determining appropriate weights for pattern classification problems. An improved version of adaptive learning algorithm in a structured multilayer networks is proposed in this research work. A proper weight setting for the constructive architecture to solve pattern classification problems is analyzed and tabulated.**

Index Terms— **Adaptive Resonance Theory, Constructive Neural Network, Pattern Classification.**

I. Introduction

Artificial Neural Networks (ANN) are biologically-inspired models of computation. They are networks with elementary processing units called neurons massively interconnected by trainable connections called weights. ANN algorithms involve training the connection weights through a systematic procedure. Learning in ANN refers to searching for an optimal network topology and weights so as to accomplish a given goal-dictated task. Learning can be categorized as *Supervised* or *Unsupervised*. The Supervised learning refers to the presence of inputs and desired outputs for training. Unsupervised learning refers to determining the output categories or correlation inherent in inputs for training. ANNs are capable of generalization, adaptation and performing computation in parallel resembling the human brain. A layer of neurons in ANN have common functionality. The choice of number of neurons in input, hidden and output layers and their functionality depend on the learning algorithm and task needed. The numbers of input neurons are usually same as the total number of attributes of the training patterns, and, the numbers of output neurons are same as the number of categories sought. Determination of number of hidden layers and neurons is dependent on the inherent complexity of the task and number of training examples available. A number of ANN architectures and algorithms have been proposed by researchers, of which Constructive Neural Networks (CoNN) offer an attractive framework for pattern classifications problems. CoNN algorithms provide an optimal way to determine the architecture of a Multi Layer Perceptron network trainable with learning algorithms to determine appropriate weights. These algorithms initially start with small network (usually a single neuron) and dynamically allow the network to grow by adding and training neurons as needed until a satisfactory solution is found[1].

Traditional neural network learning involves modification of interconnection weights between neurons on a pre-specified network. Determining the network architecture is a challenging problem in such situations, which currently requires an expensive trial-and-error process. Determining an appropriate neural network topology for classification problems has two opposing objectives. The network must be large enough to be able to adequately define the separating surface and should be small enough to generalize as well**.** Rather than learning on a pre-specified network topology, a constructive algorithm learns the topology in a manner specific to the problem. The advantage of such constructive learning is that it automatically fits network size to the data without overspecializing, but yields better generalization towards arbitrary pattern classification [2].

Constructive Neural Networks provide an optimal way to construct minimal networks for pattern classification. They are based on simple Threshold Logic Units (TLU), which implement a hard- limiting function. It starts with single TLU and additional TLUs are added if necessary. It also offers a compact network rendering simpler architecture implementation, easier extraction of knowledge rules and capability for generalization. Some of the advantages of CoNN over conventional networks are listed below.

- The choice of network topology is dynamically determined during training.
- They provide guaranteed convergence to zero classification errors on non contradictory finite datasets.
- Use of elementary threshold neurons for training
- By restricting its architectural size, it is less complex and easy to generalize.
- No extensive learning parameters needs to be used or fine tuned.
- Limitation of conventional networks in weight search as the initial weights are assigned random numbers and to be in the vicinity of inputs for convergence.

The rest of the paper is divided into five sections. Section-II presents a literature review of various existing algorithms for Constructive Neural Network architectures and for training the individual TLU's. Section-III discusses the issues in Constructive architectures and its learning algorithms. Section-IV proposes the methodology for improved adaptive learning algorithm. Section-V illustrates the practical applicability of the proposed learning algorithm in comparison with existing architectures on various datasets and analyzes their performances and Section-VI concludes and suggests topics for further research.

II. Literature Review

A. Constructive Neural Network Algorithms

A number of CoNN algorithms for constructing and training the threshold logic units appear in literatures which are discussed here.

Tiling algorithm [3], constructs a strictly layered network of threshold neurons. Each layer maintains a master neuron which classifies more patterns than master in previous layer. Ancillary neurons are added to ensure faithful representation, in which no two examples of different classes produce identical outputs.

Tower algorithm [4], constructs a tower of TLUs. The bottom most neuron receives inputs from each of N input neurons. The tower is built by successively adding neurons to the network and training them using any of the perceptron training algorithms until the desired classification accuracy is achieved. The newly added neurons receive input from each of the N input and output of neurons immediately below itself

Pyramid algorithm [5] constructs a network similar to the tower algorithm; except that each newly added neuron receive input from each of the N input neurons as well as outputs of all neurons in each of the preceding layer.

Upstart algorithm [6] constructs a binary tree of threshold neurons. First an output layer of M neurons is trained, if patterns are correctly classified, it terminates, else it finds a neuron that makes most number of errors , if it is wrongly-on or wrongly-off, daughter neurons are added to correct errors. The daughters are then connected to each neuron in output layer and trained.

Sequential algorithm [7], instead of training neurons to classify a maximal subset of patterns, it trains neurons to sequentially exclude patterns belonging to one class from other. When all patterns are excluded, the internal representation of patterns in hidden layer is linearly separable.

Perceptron Cascade algorithm [8] is similar to upstart algorithm except the daughter neurons receive input from each of input neurons and from each of previously added daughters. Similarly Cascade Correlation network is also a famous network with growing number of neurons[9].

The improved version of the above six algorithms to include real valued multi-categories like the MTower, MPyramid, MTiling, MSequential, MPerceptron cascade and MUpstart [J.Yang, R.G Parekh & V.Honavar, 1997] appear in literature which are proved to converge to zero classification error [10].

Oil-Spot algorithm [11] is based on the representation of the mapping of interest onto the binary hypercube of input space. It dynamically constructs a 2- layer network by binary examples, and in non-linear problems several vertices of N-dimensional hypercube, each representing a neuron is added until all vertices are enclosed in a positive cut .

DistAI algorithm [12] is based on inter-pattern distance which constructs a single hidden layer of spherical threshold neurons. Each neuron is designed to exclude a cluster of patterns belonging to same class. The weights are the inter pattern distances.

Dynamic Node Creation algorithm [13] adjusts the weights in a network by training the topology. It begins with minimal neural network, then trains and adds new hidden node one by one into a multilayer structure. It begins with a single node in a hidden layer, and then starts training. After some iteration, if the error is not minimized then it adds new hidden node and trains again. This procedure is continued until the error is minimized.

The following are some of the challenges and potential research areas in CoNN as proposed in [10].

1. The design of suitable TLU training algorithms that satisfy the requirements at least approximately optimal remains an open research problem.
2. A cross-validation based criteria for training with stopping condition when generalization begins to deteriorate after the addition of new neurons.
3. Hybrid network training schemes that dynamically select a network architecture, a TLU training algorithm and output strategy.
4. Using various pre-processing techniques for converting training data for better learning.
5. Pruning strategies for trained network to optimize the performance is an interesting research area.

The impact of parameters like network size, training time, generalization, convergence of the existing as well as improved algorithms is an open research problem.

B. Learning Algorithms for Constructive Neural Networks

Algorithms for training individual Threshold Logic Units in constructive networks appear in literature such as

Pocket algorithm with Ratchet Modification(PRM) in which the basic idea is to run perceptron learning algorithm while keeping an extra set of weights "in your pocket." Whenever the perceptron weights have a longest run of consecutive correct classifications of randomly selected training examples, these perceptron weights replace the pocket weights. The pocket weights are the outputs of the algorithm[4].

Thermal Perceptron Algorithm (TPA) finds stable weights for nonseparable problems as well as separable ones through a good initial setting for a pseudo artificial temperature parameter. It is proved that the thermal perceptron outperforms the Pocket algorithm and methods based on gradient descent. The learning rule stabilizes the weights over a fixed training period. [14].

Barycentric Correction Procedure (BCP) is an efficient TLU training algorithm that is not based on Perceptron, but on the geometrical concept of barycenter. The extension of the procedure deals with linearly non-seperable mapping as two versions, one is to minimize the number of misclassification patterns, and, other is to maximize the number of excluded patterns[15].

III. Issues in Constructive architecture and Learning Algorithms

The focus of this study will be limited to the famous Tiling algorithm as proposed in [9] with multi category input. The choice of Multi Category Tiling algorithm over all other Constructive Neural Network algorithms for the architecture construction is highlighted with the following reasons.

1. The input patterns need not be projected, normalized or quantized for guaranteed convergence as the network itself is a vector quantizer.
2. It ensures a faithful representation of training set, which is a necessary condition for convergence .It ensures hidden layers to be faithful.[16]. (Faithfulness: No two examples belonging to different classes produce identical output at any given layer)
3. Multi Category Tiling networks are strictly layered networks of neurons, with each layer maintaining a set of master neurons and ancillary neurons, if any, are trained progressively on smaller subsets
4. Training the neurons using winner-take-all strategy is preferred as it makes the hidden layer competitive.
5. Faster than other algorithms, as the neurons are trained only once, and the number of ancillary neurons progressively decreases as additional neurons are added.

The following are the issues in existing Multi Category Tiling algorithm which will be addressed in this research work

1. Network size grows as misclassifications occur, which reduces the performance of the network. This can be addressed by adding N/2 ancillary neurons to current layer, thereby deferring the correct classification in the current layer and moving it to the next layer.
2. As the network size grows generalization capability also decreases, so techniques to suitably modify the existing training algorithms for reducing the size of the network thereby increasing the generalization capability needs attention.
3. Choice of weight training algorithm decides the training time and accuracy. Performance of PRM, TPA and BCP are poor, so a proper competitive learning method with WTA strategy needs to be proposed.

On analyzing the literatures, it is evident that researchers have used supervised learning algorithms for Constructive Neural Networks with results showing a compromise between generalization and convergence parameters. Using an unsupervised learning algorithm like the Adaptive Resonance Theory for adaptive training strategy will be the theme of this study. Adaptive Resonance Theory refers to a class of self-organizing neural architecture that clusters the pattern space and produce appropriate weight vector templates. The potential advantages are, it addresses the famous stability-plasticity dilemma, thereby learning new patterns without affecting existing patterns[17]. Some of the issues in ART pertaining to constructive networks, which will be addressed in this proposed work, are the following

1. Proper weight setting for bottom up weights will ensure good classification.
2. Modification or removal of second gain control signal, as it merely performs an 'OR' function which is not required when used along with CoNN algorithms to reduce the training time.
3. Fixing the top down and bottom up weights initially and training them only once without modification will reduce the training time.
4. Vigilance test for ancillary neurons are not done, as they are already misclassified patterns so they are assigned to the existing ancillary neurons by modified ART algorithm.

IV. METHODOLOGY

The proposed Constructive Neural Network with Improved Adaptive Learning Algorithm has input layer nodes (N) to accept N input patterns. Master neurons (M) are added to the first output layer I. Each neuron in layer I-1 is connected to neurons in layer I through bottom up weights which are normalized. Each neuron in layer I is connected to neurons in layer I-1 through top down weights initialized to 1. Activations of neurons at layer I are calculated and winner node chosen by competitive learning. The output of layer I is presented to layer I-1 and activations of neurons at layer I-1 are calculated. A vigilance test is performed for misclassifications, if the test fails, the winner node is reset and network enters a search operation for other winner node. In case of misclassifications, only half the necessary ancillary neurons are added to the current layer, which is done to prevent the entire classification to be done at the current layer. This may increase the complexity of the network but certainly decreases the number of connections needed.

The improved adaptive learning algorithm given below, uses the improved version of traditional unsupervised ART learning technique. It trains the required master neurons with proper weight setting along the bottom-up weights as explained in the algorithm given below. The misclassified patterns are collected and necessary ancillary neurons are added to the current layer I and trained separately. This strategy ensures the convergence of the network. The impact of this algorithm on the parameters like network size and convergence rate and generalization capability is highlighted in the experimental results.

The following parameters will be used to compare the performance of existing and proposed architecture

1. Network Size: The number of nodes and layers depend on the complexity of the input pattern. So developing a new learning algorithm for topology construction to get zero classification error is proposed[18][19].
2. Convergence: When large training sets are used it takes much time for convergence. It is proved that tower, pyramid and tiling algorithms are faster than upstart, perceptron cascade and sequential algorithms. A proper weight setting will reduce the time for convergence[18][19].
3. Generalization: The CoNN algorithms generate a network with zero classification errors. If the network size is small or noise is present in training data, it leads to over fitting and the network start memorizing the misclassified patterns. So a compromise between network size and classification achieves better generalization[20].The above parameters will be analyzed on existing CoNN algorithms as well as on the proposed algorithm.

Improved Adaptive Learning Algorithm

Algorithm:

1. *Let I represent the current output layer with M master neurons and*
   Let I-1 represent the input layer with N input neurons
   Initialize I =1; Ancillary-neuron-count = 0
   For I < max-allowed-layers or I not faithful
   Begin
2. *Initialize the Bottom-up weights as*

(i) $$b_{I-1_i, I_j} = \frac{P_i^p C_j^p}{\sum_{L=1}^{M} (P_L^p)^2} \quad \text{for all neurons except } j$$

(ii) $$b_{I-1_i, I_j} = A_{I-1} \quad \text{for particular neuron } j$$

(iii) $$b_{0, I_j} = \frac{2C_j^p}{\sum_{L=1}^{M} (P_L^p)^2} \quad \text{for bias input}$$

3. *Initialize Top-down weights as*

$$t_{I_J, I-1_i} = 1$$

4. *Inputs presented and activations at layer I calculated as*

$$net_{I_J} = \sum b_{I-1_i, I_J} . P_{I-1_i}$$

$$out_J = \text{f(net)} = 1 \big/ 1 + e^{-net}$$

5. *Through WTA strategy elect the winner node in layer I as*

$$\max(net_{I_J}) = r_{I_J}$$

6. *Output at layer I are sent back to layer I-1 through top-down weights as*

$$net_{I-1_J} = \sum t_{I_j, I-1_i} . r_{I_J}$$

7. *Perform Vigilance test of whether to accept the winner neuron or*

If $$\frac{net_{I-1_J}}{P_{I-1_i}} < \rho \text{ (vigilance parameter) then}$$

Reset the winner neuron in layer I (case of Misclassification)
Increment the Ancillary-neuron-count
Collect the Input patterns which caused misclassification
Else
Update weights using pocket algorithm and Goto step 4

8. *I=I + 1*
   End
   { Training the Ancillary Neurons }
9. *Initialize the Bottom-up weights for ancillary neurons as*

(i) $$b_{I-1_i, I_j} = 0 \quad \text{for all neurons except } j \text{ and bias input}$$

(ii) $$b_{I-1_i, I_j} = \frac{P_i^p}{\sum_{I-1}^{N} (P_L^p)^2} \quad \text{for particular neuron } j$$

(iii) $$b_{0, I_j} = \text{m} \quad \text{for bias input}$$

Repeat steps 10 and 11
Begin

10. *Apply the misclassified input patterns and Calculate activations at layer I only for ancillary neurons as*

$$net_{I_J} = \sum b_{I-1_i, I_J} . P_{I-1_i} \quad \text{for ancillary neurons}$$

11. *Assign the input patterns to ancillary neurons and update weights using pocket algorithm*
    End Until there is no more misclassified input patterns

V. Experimental Results

The performance of the new Tiling network with improved adaptive learning algorithm and the existing constructive architectures like tower, pyramid, tiling are analyzed in this section. The real-world datasets **iris** and **pima Indian diabetic** are available at the UCI Machine Learning Repository [21] while the **B-pattern** dataset was artificially generated. The **B-pattern** dataset comprises of binary patterns belonging to classes 1,2, and 3 if their sum of bits vary in length.

Each dataset was divided into equal sized folds and a pattern sample size of 15 was chosen for the study for every trail on four algorithms namely Tower, Pyramid, Tiling and New Tiling.. The patterns in one fold were divided as training set and testing set. The weights of each neuron for tower, pyramid and tiling algorithms were initialized based on weight initialization strategy as given in [10]. The weights of each neuron in the new Tiling network were initialized based on improved adaptive learning algorithm as discussed above. The parameters like the number of hidden layers and neurons, number of weight connections, total weight connections in the network, generalization capability for each of the four algorithms and number of ancillary neurons generated for the last two algorithms are analyzed based on the results obtained.

Table II, III and IV summarizes the results of experiments designed to test the performance of the above said four learning algorithms on various datasets.

On applying Iris dataset as in Table-II, it was found that tower and pyramid algorithms behaved in a similar manner for number of misclassifications, but differ in number of hidden layers and neurons produced. The tiling and New Tiling algorithms behaved in a similar way for all the parameters used for the study. The performance graph is shown in Fig.1.

On applying Pima Indian diabetic dataset as given in Table III, it was found that tower and pyramid algorithms behaved in a similar manner for all the parameters. The behavior of tiling algorithm was very poor in terms of number of weight connections, total number of neurons and also the number of misclassifications. The performance of New Tiling algorithm was found to be very good in classifying the already misclassified data with less number of ancillary neurons. The performance graph is shown in Fig 2.

On applying B-Pattern dataset as given in Table IV, it was found that tower and pyramid algorithms behaved similarly. They produced 4 hidden layers for the 4 different data in the dataset. The tiling algorithm produced four ancillary neurons in 1 hidden layer and the number of weight connections was also more. The New Tiling algorithm added only half the necessary neurons (i.e one) in the hidden layer which led to less number of weight connections and there was no misclassifications. The performance graph is shown in Fig.3.

TABLE II PERFORMANCE ON IRIS DATASET

Algorithm	Hidden No. of Layers	Hidden No. of neurons	No. of weight connections	Total neurons	Generalization (misclassifications in 15)	No. of ancillary neurons
Tower	6	18	66	25	2 patterns	
Pyramid	5	15	57	22	2 patterns	
Tiling	1	3	21	10	1 pattern	none
New Tiling	1	3	21	10	1 pattern	none

TABLE III PERFORMANCE ON PIMA INDIAN DIABETIC DATASET

Algorithm	Hidden No. of Layers	Hidden No. of neurons	No. of weight connections	Total neurons	Generalization (misclassifications in 15)	No. of ancillary neurons
Tower	3	6	24	16	2 patterns	--
Pyramid	3	6	24	16	2 patterns	--
Tiling	1	9	90	19	7 patterns	7
New Tiling	1	3	30	13	none	1

TABLE IV PERFORMANCE ON B-PATTERN DATASET

Algorithm	Hidden No. of Layers	Hidden No. of neurons	No. of weight connections	Total neurons	Generalization (misclassifications in 15)	No. of ancillary neurons
Tower	4	12	54	21	4 patterns	--
Pyramid	4	12	54	21	4 patterns	--
Tiling	1	7	63	16	none	4
New Tiling	1	4	36	13	none	1

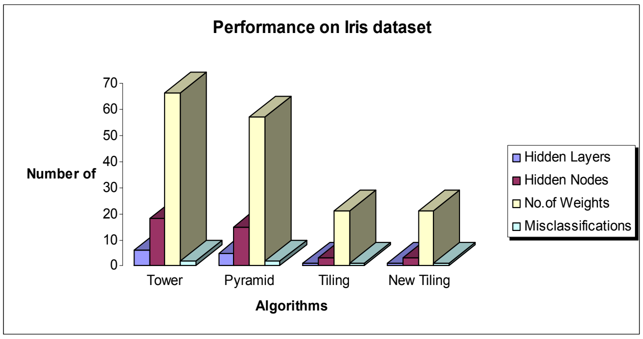

Fig.1. Performance of algorithms on IRIS dataset

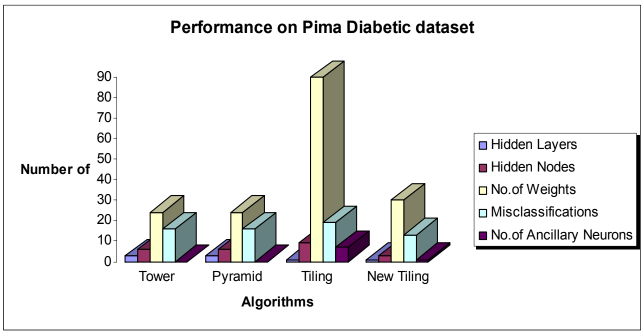

Fig.2. Performance of algorithms on Pima Indian Diabetic dataset

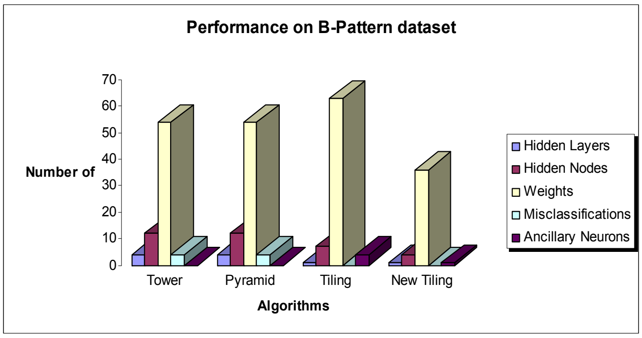

Fig.3. Performance of algorithms on B-Pattern dataset

VI. CONCLUSION

In this study an Improved Adaptive Learning Algorithm was proposed for MutiCategory Tiling Constructive Neural Networks for pattern classification problems. The performance of algorithms like Tower, Pyramid, Tiling and New Tiling networks were studied using Machine learning datasets namely the Iris and Pima Indian Diabetic and also on the generated B-Pattern dataset. It was found that the Multi Category Tiling network with improved adaptive learning algorithm outperformed other networks used for study in the parameters like the number of hidden layers and neurons generated, number of patterns classified for generalization and adoption of misclassified patterns forbetter convergence. The results show that the improved adaptive learning algorithm as proposed in this study could be useful in many pattern classification tasks involving constructive neural networks.

Some of the future research directions are

1. Various other training algorithms with proper weight setting shall be experimented for improving the performance of the New Tiling network.
2. The performance of the new learning algorithm in terms of parameters like the network size, training time, generalization capability, training methodology and convergence properties on various other datasets as found in UCI repository is a challenging research direction.
3. Various pre-processing techniques shall be applied to input data for better improvement in performance.

REFERENCES

[1] R. Parekh, J.Yang, V.Honovar, "Constructive Neural Network Learning Algorithms for Pattern Classification", IEEE Transactions on Neural Networks, Vol.11,No.2,pg. 436-451,March 2000
[2] Justin Fletcher and Zoran ObradoviC, "Constructively Learning a Near- Minimal Neural Network Architecture", *IEEE*, 1994
[3] M. Mezard and J.P. Nadal , "Learning in feedforward layered networks: the tiling algorithm", J. Phys. A: Math. Gen. 22, 2191-2203,1989
[4] S.Gallant, "Perceptron based learning algorithms," IEEE Transaction on Neural Networks, vol. 1, pp. 179–191, 1990.
[5] S.Gallant.," Neural Network learning and Expert Systems", Cambridge MA. , MIT Press, 1993
[6] M. Frean, "The Upstart Algorithm: A Method for Constructing and Training Feedforward Neural Networks", Vol. 2, No. 2, pg.98-209,1990
[7] M.Marchand et al, "A Convergence Theorem for Sequential Learning in Two-Layer Perceptrons", Europhys. Lett. 11 487-492,1990
[8] N. Burgess,A Constructive Algorithm that converges for Real-valued input patterns", International Journal of Neural Systems, Volume: 5, No.1, pp. 59-66, 1994.
[9] S. Fahlman, and C. Lebiere, "Thc Cascade Correction architecture," In Advances in Neural Infornation Processing Systems 2, Morgan Kaufman, San Mateo, CA, pp 524-532, 1990.
[10] R.Parekh, J. Yang, and V. Honavar, "Constructive Neural-Network learning algorithms for multicategory real-valued pattern classification " Dept. Computer. Science., Iowa State Univ., Tech. Rep. ISU-CS- TR97-06, 1997.
[11] F.M.Frattale-Mascioli. and G. Martinelli, "A constructive algorithm for binary neural networks: The Oil-Spot algorithm," *IEEE Transaction on Neural Networks*, vol. 6, pp. 794–797, 1995.
[12] J.Yang.J, R. Parekh and V. Honavar, "DistAl: An inter-pattern distance Based constructive learning algorithm," *Intelligent Data Analysis* 3, pp. 53-73, 1999
[13] T.Ash, "Dynamic node creation in back propagation networks," *Connection Science,1,*365−375, 1989.
[14] M.Frean,"A Thermal Preceptron learning rule", Neural Computation,MIT,pp.946-957,1992.
[15] H. Poulard and S. Labreche, "A New Threshold Unit Learning Algorithm", Technical Report 95504, LAAS, December 1995.
[16] M.C. Nicoletti and J.R.. Bertini Jr, "An empirical evaluation of constructive neural network algorithms in classification tasks",International Journal of Innovative computing and applications, Vol.1.No.1., 2007.
[17] G.A.Carpenter and S. Grossberg, "The ART of Adaptive Pattern Recognition by a self-organizing neural network",IEEE, March 1988.
[18] W.Fang and R.C.Lacher,"Network Complexity and Learning efficiency of Constructive learning algorithms", IEEE,1994
[19] Tin Yau Kwak and D.Y. Yeng," Experimental analysis of input weight freezing in Constructive Neural Networks", IEEE,1993.
[20] Md. Monirul Islam, Md.Shajahan and K.Murase,"Exploring Constructive algorithms with stopping criteria to produce accurate and diverse individual neural networks in an ensemble", IEEE, 2001.
[21] P.Murphy and D.Aha, "Repository of machine learning databases", Dept. Inform. Comput. Sci., Univ. California,Irvine,CA, http://www.ics.uci.edu/AI/ML/MLDBRepository.html,1994

S.S.Sridhar received his B.E.(Computer Engineering) from Madurai Kamaraj University,India in 1990 and M.S (Software Systems) from Birla Institute of Technology and Science, Pilani, India in 1995 and currently doing his Research from SRM University,Chennai,India in the field of Neural Network Architectures for Pattern Classification.
e-mail: sridhar.srm@gmail.com.

M.Ponnavaikko received his B.E.(Electrical Engg.) and M.Sc.(Engg.) from College of Engineering Guindy, India in 1969 and 1972 respectively and Ph.D in Optimal Distribution system planning from IIT Delhi in 1983. He has worked in various government organizations and universities and has got vast experience in various capacities starting from Engineer to Vice Chancellor. He is currently the Vice Chancellor of Bharathidhasan University, Trichy, India and he is a well known personality in the field of Computer Science in Southern India.
e-mail: ponnav@gmail.com