Journal of Computer Science 5 (11): 843-848, 2009
ISSN 1549-3636

New Constructive Neural Network Architecture for Pattern Classification

[1]S.S. Sridhar and [2]M. Ponnavaikko
[1]Department of Computer Science and Engineering,
Sri Ramaswamy Memorial University, Kattankulathur, Chennai-603203, India
[2]Bharathidasan University, Trichy, India

Abstract: Problem statement: Constructive neural network learning algorithms provide optimal ways to determine the architecture of a multi layer perceptron network along with learning algorithms for determining appropriate weights for pattern classification problems. These algorithms initially start with small network and dynamically allow the network to grow by adding and training neurons as needed until a satisfactory solution is found. The constructive neural network training is performed via feed forward paradigm under supervised training considerations. These supervised methods often make the network size grow exponentially, or, the network lacks generalization. To address these problems a new method for learning in constructive neural networks is necessary. **Approach:** To address these issues a new Multicategory Tiling architecture was chosen for its simple topology and an improved adaptive resonance theory unsupervised training algorithm was used with proper weight setting to train the constructive networks on binary sequence patterns. The results and performance of the new algorithm was compared with existing constructive neural network architectures and tabulated. **Results:** The new architecture with improved training algorithm offer faster convergence in learning, the nodes required for storage are less and the generalization of pattern classification was achieved in comparison with existing algorithms. **Conclusion:** Constructive neural networks could be trained using unsupervised algorithm to achieve better performance in comparison with existing supervised algorithms.

Key words: Adaptive resonance theory, constructive neural networks, multicategory tiling architecture, pattern classification

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. Supervised learning refer to the presence of inputs and desired outputs for training. Unsupervised learning refer 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 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. Constructive Neural Networks provide an optimal way to construct minimal networks for pattern classification. They are based on simple threshold logic units, which implement 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. The choice of network topology is dynamically determined during training. Some of the advantages of CoNN over conventional networks are they provide guaranteed convergence to zero classification errors on non contradictory finite data sets. 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.

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

Corresponding Author: S.S. Sridhar, Department of Computer Science and Engineering,
Sri Ramaswamy Memorial University, Kattankulathur, Chennai-603203, India Tel: 91-044-9840517356

Tiling algorithm[1] 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[2] 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[2] constructs a network similar to the tower algorithm, except that each newly added neurons receive input from each of the N input neurons as well as outputs of all neurons in each of the preceding layer.

Upstart algorithm[3] 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[4] 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[5] is similar to upstart algorithm except the daughter neurons receive input from each of input neurons and from each of previously added daughters.

The improved version of the above 6 algorithms to include real valued multi-categories like the MTower, MPyramid, MTiling, MSequential, MPerceptron cascade and MUpstart appear in literature which are proved to converge to zero classification error[6].

Oil-spot algorithm is[7] 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[8] 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[3].

Dynamic node creation algorithm 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. Training starts with a single node in a hidden layer, if the error is not minimized new hidden nodes are added and trained. This procedure is continued until the error is minimized[9].

Algorithms for training individual Threshold Logic Units in constructive networks appear in literature like Pocket algorithm with Ratchet Modification (PRM)[2] 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.

Thermal Perceptron Algorithm (TPA)[10] which 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. For separable problems, it finds separating weights much more quickly than the usual rules[10].

Barycentric Correction Procedure (BCP)[11] 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[11]. So, algorithms for constructing the architecture, training the TLU’s are quite different, which exist in literatures above.

In this research study Multicategory Tiling architecture (MTiling) is preferred over all other Constructive Neural Networks because of the following reasons:

- The input patterns need not be projected, normalized or quantized for guaranteed convergence as the network itself is a vector quantizer
- It ensures a faithful representation of training set, which is a necessary condition for convergence (Faithfulness: No two examples belonging to different classes produce identical output at any given layer[6])
- MTiling networks are strictly layered networks of TLUs, with each layer maintaining a set of master

neurons and ancillary neurons, if any, are trained progressively on smaller subsets

- Training TLUs using winner-take-all strategy is preferred as it makes the hidden layer competitive
- Faster than other constructive 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 MTiling algorithm which will be addressed in this research study:

- 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 to next layer
- 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 like improved adaptive resonance theory algorithm with proper weight setting is proposed
- As the network size grows generalization capability decreases, so techniques to suitably modify the existing training algorithms for reducing the size and increasing the generalization capability will be addressed

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[12]. Some of the issues in ART which will be addressed in this research work are the following:

- Proper weight setting for bottom up weights will ensure good classification
- 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
- Fixing the top down and bottom up weights initially and training them only once without modification will reduce the training time
- Vigilance test for ancillary neurons are not as they are already misclassified patterns and are assigned to the existing ancillary neurons by modified ART algorithm

MATERIALS AND METHODS

The proposed new MTiling constructive neural network learning architecture as shown in Fig. 1 constructs a layered network of threshold neurons through MTiling algorithm as given in Fig. 2. Salient features of the new architecture are given below:

- The input layer neurons receive N inputs and acts initially as comparison layer. The next layer and subsequent layers receive inputs from those layers immediately below itself
- There are two different types of weights (connections) between the layers as proposed in I-ART algorithm, namely top down and bottom up weights
- Each layer except the output layer has a single gain control signal, also, each layer except the input layer has a single reset signal and bias input

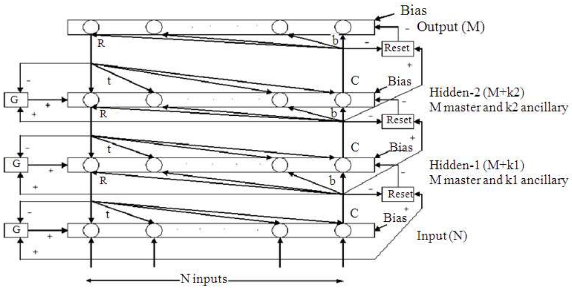
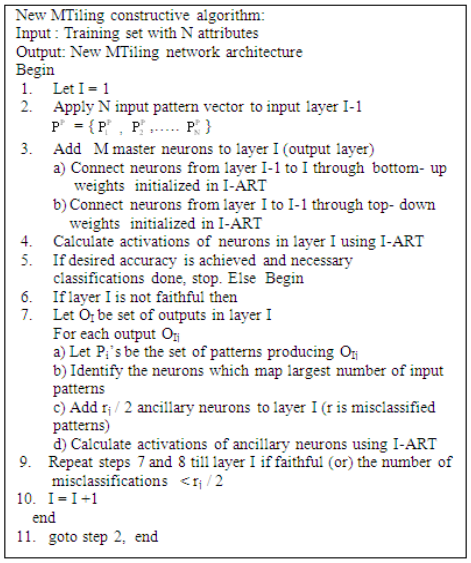


Fig. 1: New MTiling I-ART constructive neural network architecture



```
New MTiling constructive algorithm:
Input : Training set with N attributes
Output: New MTiling network architecture
Begin
 1.   Let I = 1
 2.   Apply N input pattern vector to input layer I-1
      $P^p = \{ P_1^p , P_2^p ,\ldots P_N^p \}$
 3.   Add  M master neurons to layer I (output layer)
      a) Connect neurons from layer I-1 to I through bottom- up
         weights initialized in I-ART
      b) Connect neurons from layer I to I-1 through top- down
         weights initialized in I-ART
 4.   Calculate activations of neurons in layer I using I-ART
 5.   If desired accuracy is achieved and necessary
      classifications done, stop. Else Begin
 6.   If layer I is not faithful then
 7.   Let $O_I$ be set of outputs in layer I
      For each output $O_{Ij}$
      a) Let $P_i$'s be the set of patterns producing $O_{Ij}$
      b) Identify the neurons which map largest number of input
      patterns
      c) Add $r_i$ / 2 ancillary neurons to layer I (r is misclassified
      patterns)
      d) Calculate activations of ancillary neurons using I-ART
 9.  Repeat steps 7 and 8 till layer I if faithful (or) the number of
     misclassifications  $< r_i / 2$
10.  I = I +1
   end
11.  goto step 2,  end
```

Fig. 2: New MTiling algorithm

- Each hidden layer (or) recognition layer and output layer maintains a set of master neurons which correctly classify more patterns, if master neurons does not classify, then half the required ancillary neurons are added to reduce misclassifications, which defers the classification to next layer thereby reducing the number of connections
- The top down and bottom up weights are initialized by I-ART algorithm as given below for master neurons and ancillary neurons, to ensure faithful representation of training set which also reduces misclassifications

The N input patterns are applied to input layer I-1, then M master neurons are added to 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 through I-ART algorithm. 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. Every time for 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 rather to defer it to the next layer. This may increase the complexity of the architecture but certainly decreases the number of connections needed.

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

Network size: The number of nodes and layers depend on the complexity of the input pattern. So developing a new modified MTiling learning algorithm for topology construction to get zero classification error is proposed[13,14].

Generalization: The CoNN algorithms generate a network with zero classification errors. If the network size is small, it leads to over fitting and the network start memorizing the misclassified patterns. So a compromise between network size and classification achieves better generalization[15].

Improved Adaptive Resonance Theory algorithm (I-ART) for training master and ancillary neurons: This algorithm produces necessary master neurons for a particular layer and each of the master neuron is trained using proper weight setting along the bottom-up and top-down weights as given below. In case of ancillary neurons, the algorithm considers only the misclassified patterns and trains only half of the necessary ancillary neurons.

Bottom-up weights:

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

$$b_{I-1_i,I_j} = A_{I-1} \quad \text{(A is number of neurons)}$$

$$b_0,_{I_j} = \frac{2C_j^p}{\sum_{L=1}^{M} (P_L^p)^2} \quad \text{(bias input)}$$

Top-down weights: $t_{I_j}, I-1_i = 1$, vigilance: $0<\rho<1$

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

Experimental setup: The dataset considered as inputs and adopted in the proposed strategy correspond to binary pattern vectors of fixed number of bits so as to classify them according to different categories in the dataset. The various existing constructive architectures like tower, pyramid, tiling were implemented in C language and the new MTiling constructive architecture was also implemented in the same language. The binary dataset attributes P1 to P6 as given in Table 1 were used as training inputs to the new architecture along with target categories.

The input patterns P7-P11 as given in Table 2, were used as testing dataset in all the four algorithms namely Tower, Pyramid, Tiling and new MTiling algorithms. On testing datasets, the performance of each algorithm was analyzed and discussed below. The output category assigned to the pattern along with desirable category is also given in boldface in Table 2.

Table1: Patterns used for training

Pattern-Id	Input pattern	Category
P1	000111	1
P2	111000	1
P3	000011	2
P4	110000	2
P5	001111	3
P6	111100	3

Table 2: Patterns used for testing

		Tower		Pyramid		Tiling		New MTiling	
Pattern-Id	Input pattern	Hidden neurons	Category	Hidden neurons	Category	Ancillary neurons	Category	Ancillary neurons	Category
P7	011111	3	3	3	3	2	3	-	3
P8	111110	3	3	3	**1(3)**	2	3	-	3
P9	000001	3	**3(2)**	3	**3(2)**	2	2	-	2
P10	100000	3	**1(2)**	3	**1(2)**	2	2	-	2
P11	011110	3	**2(3)**	3	**1(3)**	-	3	1	3

Category in boldface are misclassifications along with correct category in brackets

Table 3: Performance of four algorithms

Algorithm	Hidden no. of Layers	No. of Neurons	No. of connections	Total neurons	Generalization
Tower	5	15	63	24	Classifies P7, P8 Misclassifies P9, P10, P11
Pyramid	5	15	63	24	Classifies P7 Misclassifies P8, P9, P10, P11
Tiling	1	5	45	14	Classify all patterns
MTiling	1	4	36	13	Classify all patterns

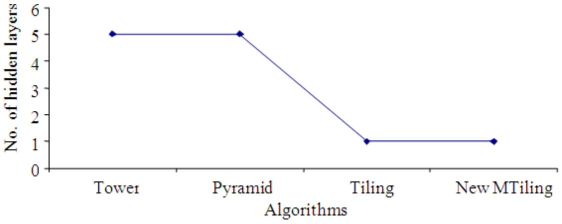


Fig. 3: Hidden layers

RESULTS

The dataset used for training (i.e.) P1 to P6 along with their categories were learned by all the four algorithms. The performance of four algorithms as given in Table 3 is discussed here.

The Tower algorithm introduced a hidden layer with 3 neurons for every new pattern in testing dataset, which exponentially makes the network very complex with 5 hidden layers as given in Fig. 3. It also misclassifies patterns P9-P11 thereby not able to generalize. Further the tower algorithm produced an architecture which had 24 neurons, which increases the training time.

The Pyramid algorithm introduced a hidden layer with 3 neurons like tower algorithm. For every new pattern in testing dataset, the network is very complex with 5 hidden layers as given in Fig. 3. It also misclassifies patterns P8 to P11, thereby not able to generalize. Further the pyramid algorithm produced an architecture which had 24 neurons, which increases the training time.

The Tiling algorithm does not introduce new hidden layers; rather it introduces ancillary neurons for patterns P7-P10. For pattern P11 it does not introduce ancillary neurons, since it belongs to category 3. This algorithm classifies all patterns as given in Table 2 so there is no misclassifications. The new MTiling architecture introduced 1 hidden layer with 3 master neurons and 1 ancillary neuron. The total number of neurons produced in the network was 13 and 36 which reduces the training time. This network also generalizes for the given input pattern.

DISCUSSION

Performance of algorithm: The selective binary patterns when applied as input to existing constructive neural network algorithms as well as with the proposed new algorithm and performance was analyzed. The performance of Tower, Pyramid, Tiling and New MTiling algorithm in terms of number of hidden layers, number of connections (weights) and generalization capability was analyzed and plotted. Figure 3 shows better performance by Tiling and new MTiling algorithms for number of hidden layers. Figure 4 shows better performance by new MTiling algorithm for number of weight connections. Figure 5 shows better performance by Tiling and new MTiling algorithms for generalization capability. So the New MTiling algorithm out performed all existing constructive neural network algorithms for the limited set of binary datasets thereby ensuring faster convergence.

Pattern classification problems: The above study is limited to binary datasets only. Application of real datasets as found in UCI Machine learning repository on our architecture for pattern classification is reserved as future research.

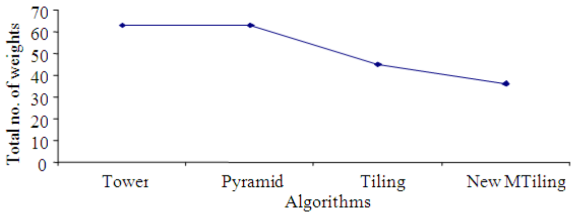


Fig. 4: Comparison of weight connections

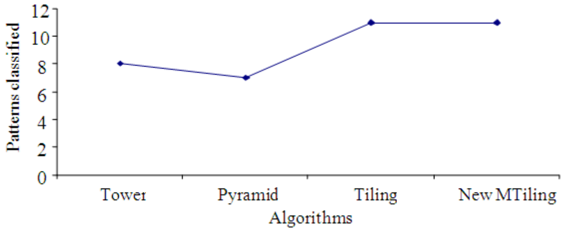


Fig. 5: Generalization capability

CONCLUSION

Constructive neural networks offer an attractive framework for pattern classifications problems. Which provide an optimal way to determine the architecture of a multilayer perceptron network trainable with supervised learning algorithms. In this study we proposed a new MTiling architecture with unsupervised learning strategy on binary pattern datasets for achieving better performance in terms of generalization capability, faster convergence and less connections thereby less storage requirement. This architecture could also be applied to other datasets which is an open research problem.

REFERENCES

1. Mézard, M. and J. Nadal, 1989. Learning feedforward networks: The tiling algorithm. J. Phys. A: Math Gen., 22: 2191-2203. DOI: 10.1088/0305-4470/22/12/019
2. Gallant, S.I., 1990. Perceptron based learning algorithms. IEEE Trans. Neural Networks, 1: 179-191. DOI: 10.1109/72.80230
3. Frean, M., 1990. The upstart algorithm: A method for constructing and training feedforward networks. Neural Comput., 2: 198-209. DOI: 10.1162/neco.1990.2.2.198
4. Marchand, M., M. Golea and P. Rujan, 1990. A convergence theorem for sequential learning in two-layer perceptrons. Europhys. Lett., 11: 487-492. DOI: 10.1209/0295-5075/11/6/001
5. Burgess, N., 1994. A constructive algorithm that converges for real-valued input patterns. Int. J. Neural Syst., 5: 59-66. DOI: 10.1142/S0129065794000074
6. Parekh, R., J. Yang and V. Honavar, 1997. Constructive neural-network learning algorithms for multicategory real-valued pattern classification. Technical Report ISU-CS-TR97-06. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.1374
7. Mascioli, F.M.F. and G. Martinelli, 1995. A constructive algorithm for binary neural networks: The oil-spot algorithm. IEEE Trans. Neural Networks, 6: 794-797. http://cat.inist.fr/?aModele=afficheN&cpsidt=3508167
8. Yang, J., R. Parekh and V. Honavar, 1999. DistAl: An inter-pattern distance Based constructive learning algorithm. Intel. Data Anal., 3: 53-73. http://tw.rpi.edu/wiki/DistAl:_An_inter-pattern_distance-based_constructive_learning_algorithm
9. Ash, T., 1989. Dynamic node creation in back propagation networks. Connect. Sci., 1: 365-375. DOI: 10.1080/09540098908915647
10. Frean, M., 1992. A thermal preceptron learning rule. Neural Comput., 4: 946-957. http://portal.acm.org/citation.cfm?id=159794
11. Poulard, H. and S. Labreche, 1995. A new threshold unit learning algorithm. Technical Report 95504. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.53.5691
12. Carpenter, G.A. and S. Grossberg, 1988. The ART of adaptive pattern recognition by a self-organizing neural network. IEEE Comput. Soc., 21: 77-88. DOI: 10.1109/2.33
13. Fang, W. and R.C. Lacher, 1994. Network complexity and learning efficiency of constructive learning algorithms. Proceeding of the IEEE International Conference on Neural Networks, IEEE World Congress on Computational Intelligence, June 27-July 2, IEEE Xplore Press, Orlando, FL., USA., pp: 366-369. DOI: 10.1109/ICNN.1994.374191
14. Kwok, T.Y. and D.Y. Yeng, 1993. Experimental analysis of input weight freezing in constructive neural networks. Proceedings of the IEEE International Conference on Neural Networks, Mar. 28-Apr. 1, IEEE Xplore Press, San Francisco, CA., USA., pp: 511-516. DOI: 10.1109/ICNN.1993.298610
15. Islam, M. M. Shajahan, and K. Murase, 2001. Exploring constructive algorithms with stopping criteria to produce accurate and diverse individual neural networks in an ensemble. Proceeding of the IEEE International Conference on Systems, Man, and Cybernetics, Oct. 7-10, IEEE Xplore Press, Tucson, AZ., USA., pp: 1526-1531. DOI: 10.1109/ICSMC.2001.973500