MTiling - A Constructive Neural Network Learning Algorithm for Multi-Category Pattern Classification

J. Yang, R. G. Parekh & V. Honavar *
Artificial Intelligence Research Group
Department of Computer Science
Iowa State University
Ames, IA 50011. U.S.A.

Abstract

Constructive learning algorithms offer an approach for incremental construction of potentially near-minimal neural network architectures for pattern classification tasks. Such algorithms help overcome the need for ad-hoc and often inappropriate choice of network topology in the use of algorithms that search for a suitable weight setting in an otherwise a-priori fixed network architecture. Several such algorithms proposed in the literature have been shown to converge to zero classification errors (under certain assumptions) on a finite, non-contradictory training set in a 2-category classification problem. This paper presents MTiling, a multi-category extension of *Tiling* algorithm [Mézard & Nadal, 89]. We establish the convergence of MTiling to zero classification error on a multi-category pattern classification task. Results of experiments with non linearly separable multi-category data sets demonstrate the feasibility of this approach to multi-category pattern classification and also suggest several interesting directions for future research.

1 Introduction

Multi-layer networks of threshold logic units (TLU) or multi-layer perceptrons (MLP) offer a particularly attractive framework for the design of pattern classification and inductive knowledge acquisition systems for a number of reasons including: potential for parallelism and fault tolerance; significant representational and computational efficiency that they offer over disjunctive normal form (DNF) functions and decision trees [Gallant, 93]; and simpler digital hardware realizations than their continuous counterparts.

A single TLU, also known as *perceptron*, can be trained to classify a set of input patterns into one of two classes. A TLU is an elementary processing unit that computes a function of the weighted sum of its inputs. Assuming that the patterns are drawn from an N-dimensional Euclidean space, the output O^p, of a TLU with weight vector $\mathbf{W}$, in response to a pattern $\mathbf{X}^p$, is a bipolar hardlimiting function of $\mathbf{W} \cdot \mathbf{X}^p$, i.e. $O^p = 1$ if $\mathbf{W} \cdot \mathbf{X}^p > 0$ and 0 otherwise. Such a TLU or threshold neuron implements a $(N-1)$-dimensional hyperplane given by $\mathbf{W} \cdot \mathbf{X} = 0$ which partitions the N-dimensional Euclidean pattern space defined by the coordinates $x_1 \cdots x_N$ into two regions (or two classes). Given a set of *examples* $S = S_+ \cup S_-$ where $S_+ = \{(\mathbf{X}^p, C^p) \mid C^p = 1\}$ and $S_- = \{(\mathbf{X}^p, C^p) \mid C^p = 0\}$ (C^p is the desired output of the pattern classifier for the input pattern $\mathbf{X}^p$), it is the goal of a *perceptron training* algorithm to attempt find a weight vector $\hat{\mathbf{W}}$ such that $\forall \mathbf{X}^p \in S_+$, $\hat{\mathbf{W}} \cdot \mathbf{X}^p > 0$ and $\forall \mathbf{X}^p \in S_-$,$\hat{\mathbf{W}} \cdot \mathbf{X}^p \leq 0$. If such a weight vector ($\hat{\mathbf{W}}$) exists for the pattern set S then S is said to be *linearly separable*. Several iterative algorithms are available for finding such a $\hat{\mathbf{W}}$ if one exists [Nilsson, 65; Duda & Hart, 73] or a reasonably good weight vector that correctly classifies a large fraction of the training set if S is not linearly separable (e.g., *pocket algorithm* [Gallant, 93], *thermal perceptron* [Frean, 90], *barycentric correction procedure* [Poulard, 95]). For a detailed comparison of the single TLU training algorithms see [Yang *et al.*, 96].

When S is not linearly separable, a multi-layer network of TLUs is needed to learn a complex decision boundary that correctly classifies all the training examples. The focus of this paper is on *constructive*

*This research was partially supported by the National Science Foundation grant IRI-9409580 to Vasant Honavar.

or *generative* learning algorithms that incrementally construct networks of threshold neurons to correctly classify a given (typically non linearly separable) training set. Some of the motivations for studying such algorithms [Honavar, 90; Honavar & Uhr, 93; Parekh, *et al.*, 95] include: Limitations of learning by weight modification alone within an otherwise a-priori fixed network topology; The need to discover near-minimal networks whose *complexity* (as measured by number of nodes, links, etc.) matches the intrinsic complexity of the classification task (implicitly specified by the training data) (for efficient hardware implementations and improved generalization); Their potential to provide useful empirical estimates of the complexity of neural circuits needed for hard pattern classification tasks; Their potential for trading off among performance measures (e.g., learning time, network size, generalization).

A number of constructive algorithms that incrementally construct networks of threshold neurons for 2-category pattern classification tasks have been proposed in the literature. These include the *tower*, *pyramid* [Gallant, 90], *tiling* [Mézard & Nadal, 89], *upstart* [Frean, 90], and *perceptron cascade* [Burgess, 94]. They are all based on the idea of transforming the hard task of determining the necessary network topology and weights to two subtasks: Incremental addition of one or more threshold neurons to the network when the existing network topology fails to achieve the desired classification accuracy on the training set; and training the added threshold neuron(s) using some variant of the perceptron training algorithm (e.g., the pocket algorithm). These algorithms differ in terms of the topological and connectivity constraints as well as training strategies used for individual neurons. The interested reader is referred to [Chen *et al.*, 95] for an analysis (in geometrical terms) of the decision boundaries generated by some of these constructive learning algorithms. Each of these algorithms can be shown to converge to networks which yield zero classification errors on any given training set in the 2-category case. The convergence proof in each case is based on the ability of the variant of the perceptron training algorithm to find a weight setting for each newly added neuron or neurons such that the number of pattern misclassifications is reduced by at least one each time a unit (or a set of units) is added and trained. We will refer to such a variant of the perceptron algorithm as L_W. In practice, the performance of the constructive algorithm depends partly on the choice of L_W and its ability to find weight settings that reduce the total number of misclassifications each time a new unit is added to the network and trained. Some possible choices for L_W are the *pocket algorithm*, the *thermal perceptron*, and other variants of the perceptron algorithm for non linearly separable data sets.

Pattern classification tasks that arise in practice often require assigning patterns to one of M ($M > 2$) classes. Although in principle, an M-category classification task can be reduced to an equivalent set of M 2-category classification tasks (each with its own training set constructed from the given M-category training set), a better approach might be one that takes into account the inter-relationships between the M output classes. For instance, the knowledge of membership of a pattern $\mathbf{X}^p$ in category Ψ_i can be used by the learning algorithm to effectively rule out its membership in a different category Ψ_j ($j \neq i$) and any internal representations learned in inducing the structure of Ψ_i can therefore be exploited in inducing the structure of a category Ψ_j ($j \neq i$). Thus, extensions of 2-category constructive learning algorithms to deal with multi-category classification tasks are clearly of interest. However, in most cases, such extensions have not been explored while in other cases, only some preliminary ideas (not supported by detailed theoretical or experimental analysis) for possible multi-category extensions of 2-category algorithms are available in the literature. Against this background, this paper develops a provably convergent modification of *tiling* algorithm [Mézard & Nadal, 89] for construction of networks of threshold neurons for pattern classification. Preliminary experiments on two classification tasks (an artificial task involving random boolean mappings, and a real-world task of classifying the *iris* data set) suggest the feasibility of the proposed approach.

2 MTiling - A Multi-Category Constructive Learning Algorithm

The two-category tiling algorithm [Mézard & Nadal, 89] constructs a strictly layered network of threshold neurons. The bottom-most layer of neurons receives inputs from each of the N input neurons. The neurons in each subsequent layer receive inputs from the neurons in the layer immediatedly below itself. Each layer maintains a *master neuron*. The network construction procedure ensures that the master neuron in a given layer correctly classifies more patterns than the master neuron of the previous layer. Ancillary units may be added to layers and trained to ensure a *faithful representation* of the training set. The *faithfulness* criterion simply ensures that no two training examples belonging to different classes produce identical output at any given layer. Faithfulness is clearly a necessary condition for convergence in strictly layered networks [Mézard & Nadal, 89].

The proposed extension to multiple output classes involves constructing layers with M master neurons (one for each of the output classes). Sets of one or more ancillary neurons are trained at a time in an attempt to make the current layer faithful. Fig. 1 shows the construction of a tiling network.

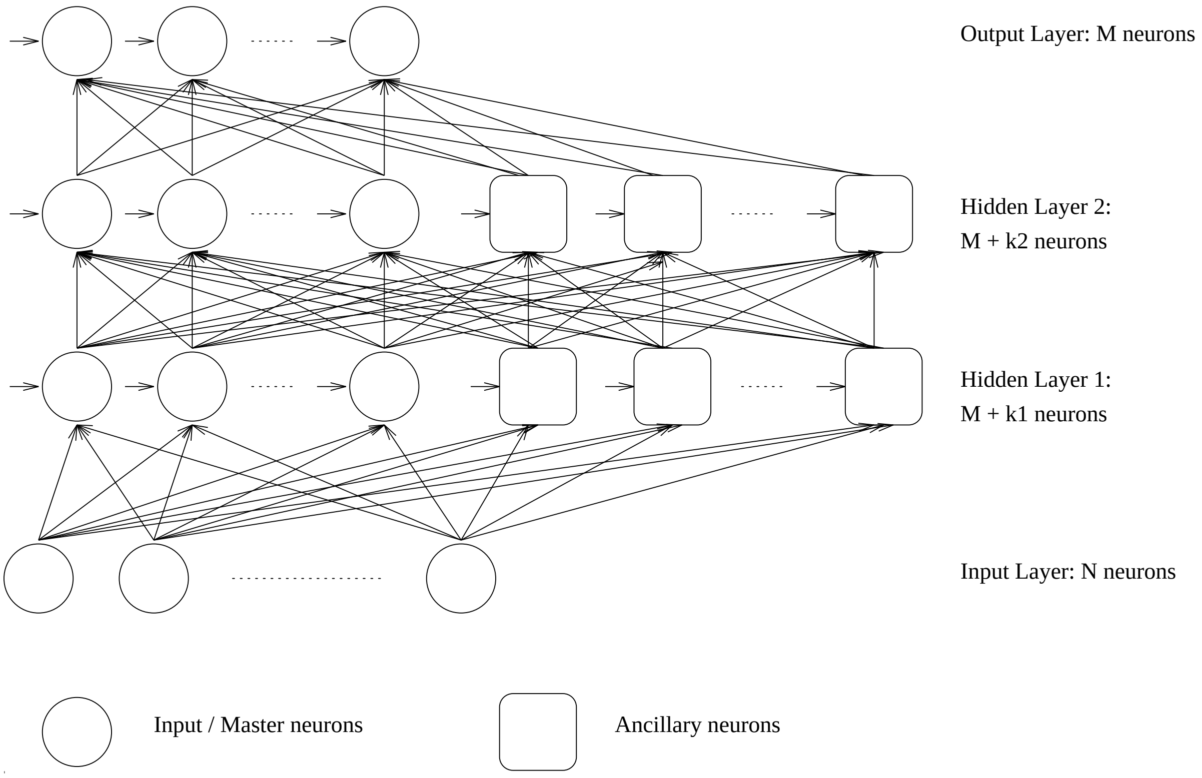

Figure 1: Tiling Network

2.1 Multi-Category Tiling Algorithm

1. Train a layer of M master neurons. Each master neuron is connected to the N inputs.
2. If the master neurons of the current layer can achieve the desired classification accuracy then stop.
3. Otherwise, if the current layer is not faithful, add ancillary neurons to the current layer to make it faithful as follows, else go to step 4.
   (a) Among all the unfaithful output vectors at the current output layer, identify the one that the largest number of input patterns map to. (An output vector is said to be unfaithful if it is generated by input patterns belonging to different classes).
   (b) Determine the set of patterns that generate the output vector identified in step 3(a) above. This set of patterns will form the training set for ancillary neurons.
   (c) Add a set of k ($1 \leq k \leq M$) ancillary units where k is the number of target classes represented in the set of patterns identified in the above step and train them.
   (d) Repeat these last three steps (of adding and training ancillary units) till the output layer representation of the patterns is faithful.
4. Train a new layer of M master neurons that are connected to each neuron in the previous layer and go to step 2.

2.2 Convergence Proof

Let N and M be the number of input and output neurons respectively; the number of units in layer A be U_A; the weight vector of neuron j be $\mathbf{W}_j$; the net input of neuron j of layer A in response to pattern $\mathbf{X}^p$ be $n^p_{A_j}$; the threshold (or bias) for unit i of layer A be $W_{A_i,0}$; the weight between unit i of layer A and unit j of layer B be W_{A_i,B_j}. The neurons in layer A are indexed by $A_1, A_2, \ldots, A_{U_A}$. Let $\mathbf{X}^p = < X^p_0, X^p_1, \ldots, X^p_N >$, $X^p_0 = 1$ for all p be the augmented pattern vector, the corresponding target output $\mathbf{C}^p = < C^p_1, C^p_2, \ldots, C^p_M >$ (where $C^p_i = 1$ if $\mathbf{X}^p$ belongs to category i and $C^p_i = -1$ otherwise), and the observed output $\mathbf{O}^p_A = < O^p_{A_1}, O^p_{A_2}, \ldots, O^p_{A_k} >$ where $U_A = k$. A pattern is said to be correctly classified at layer A when $\mathbf{C}^p = \mathbf{O}^p_A$. Let the number of patterns wrongly classified at layer A be e_A. Define $sgn(x) = -1$ if $x < 0$ and $sgn(x) = 1$ if $x \geq 0$.

In the tiling algorithm each hidden layer contains M master units plus several ancillary units to achieve a faithful representation of the patterns in the layer. Let $\tau^p =< \tau^p_1, \tau^p_2, \ldots, \tau^p_{M+K} >$ (also called a prototype) be the representation of a subset of patterns that have the same output in a layer (say A) with $U_A = M$ (master) + K (ancillary) units. $\tau^p_i = \pm 1$ for all $i = 1 \ldots (M+K)$.

Theorem:
Suppose that all classes in layer $L-1$ are faithful and that the number of errors of the master units (e_{L-1}) is non-zero. There exists a weight setting for the master units of the newly added layer (L) such that $e_L < e_{L-1}$.

Proof:
Consider a prototype τ^p for which the master units at layer $L-1$ do not yield the correct output. i.e., $< \tau^p_1, \tau^p_2, \ldots \tau^p_M > \neq < C^p_1, C^p_2, \ldots C^p_M >$. The following weight setting for the master unit j ($j = 1 \ldots M$) in layer L results in correct output for prototype τ^p at layer L. Also, this weight setting ensures that the outputs of all other prototypes, τ^q, for which the master units at layer $L-1$ produce correct outputs (i.e. $< \tau^q_1, \tau^q_2, \ldots \tau^q_M > = < C^q_1, C^q_2, \ldots C^q_M >$), are unchanged. Thus,

$$W_{L_j,0} = 2C^p_j; \quad W_{L_j,L-1_k} = C^p_j \tau^p_k \text{ for } k = 1 \ldots U_{L-1}, k \neq j; \quad \text{and } W_{L_j,L-1_j} = U_{L-1}$$

Figure 2 shows the weight setting.

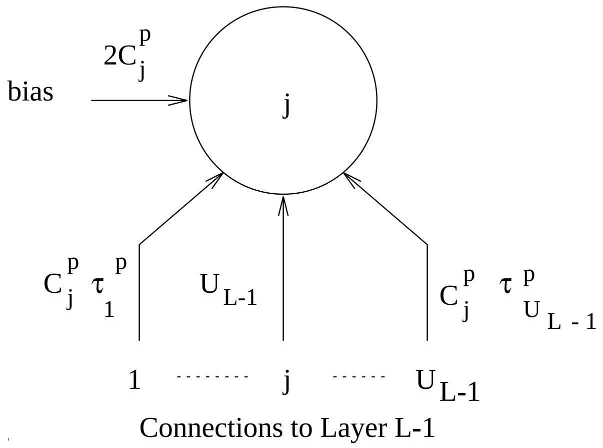

Figure 2: Weight Setting for the jth output neuron in the Tiling Network

For prototype τ^p we have:

$$n^p_{L_j} = W_{L_j,0} + \sum_{k=1}^{U_{L-1}} W_{L_j,L-1_k} \tau^p_k = 2C^p_j + U_{L-1}\tau^p_j + (U_{L-1} - 1)C^p_j = U_{L-1}\tau^p_j + (U_{L-1} + 1)C^p_j$$

and hence $O^p_{L_j} = sgn(n^p_{L_j}) = C^p_j$. For prototype τ^q (see above) $\neq \tau^p$, we have:

$$n^q_{L_j} = W_{L_j,0} + \sum_{k=1}^{U_{L-1}} W_{L_j,L-1_k} \tau^q_k = 2C^p_j + U_{L-1}\tau^q_j + \sum_{k=1,k\neq j}^{U_{L-1}} W_{L_j,L-1_k} \tau^q_k = 2C^p_j + U_{L-1}\tau^q_j + \sum_{k=1,k\neq j}^{U_{L-1}} C^p_j \tau^p_k \tau^q_k$$

CASE I: $\tau_j^q \neq \tau_j^p$ and $\tau_k^q = \tau_k^p$ for $1 \leq k \leq U_{L-1}, k \neq j$
Since τ^q is correctly classified at layer $L-1$ whereas τ^p is not, $\tau_j^q = C_j^p$ since $\tau_j^q = -\tau_j^p$ and $C_j^p = -\tau_j^p$.

$$n_{L_j}^q = 2C_j^p + U_{L-1}\tau_j^q + \sum_{k=1,k\neq j}^{U_{L-1}} C_j^p \tau_k^p \tau_k^q = 2C_j^p + U_{L-1}\tau_j^q + (U_{L-1}-1)C_j^p = (2U_{L-1}+1)\tau_j^q \text{ since } \tau_j^q = C_j^p$$

Thus, $O_{L_j}^q = sgn(n_{L_j}^q) = \tau_j^q$.

CASE II: $\tau_k^q \neq \tau_k^p$ for some k, $1 \leq k \leq U_{L-1}, k \neq j$
In this case, $\sum_{k=1,k\neq j}^{U_{L-1}} C_j^p \tau_k^p \tau_k^q \leq (U_{L-1}-3)C_j^p$. Hence,

$$n_{L_j}^q = 2C_j^p + U_{L-1}\tau_j^q + \sum_{k=1,k\neq j}^{U_{L-1}} C_j^p \tau_k^p \tau_k^q \leq 2C_j^p + U_{L-1}\tau_j^q + (U_{L-1}-3)C_j^p = (U_{L-1}-1)C_j^p + U_{L-1}\tau_j^q$$

Thus, $O_{L_j}^q = sgn(n_{L_j}^q) = \tau_j^q$ since $U_{L-1}\tau_j^q$ dominates $(U_{L-1}-1)C_j^p$. We rely on algorithm L_W to find an appropriate weight setting. With the above weights the previously incorrectly classified prototype, τ^p, would be corrected and all other prototypes that were correctly classified would be unaffected. This reduces the number of incorrect prototypes by one (i.e. $e_L < e_{L-1}$). Since the training set is finite, the number of prototypes must be finite, and with a sufficient number of layers the tiling algorithm would eventually converge to zero classification errors. □

3 Summary and Discussion

Constructive neural network learning algorithms offer a potentially powerful approach to inductive learning for pattern classification applications. This paper has developed MTiling, a provably convergent extension of 2-category tiling algorithm [Mézard & Nadal, 89] to handle multi-category classification.

The convergence of MTiling to zero classification errors was established by showing that each modification of the network topology guarantees the existence of a weight setting that would yield a classification error that is less than that provided by the network before such modification and assuming a weight modification algorithm L_W that would find such a weight setting. We do not have a rigorous proof that any of the graceful variants of perceptron learning algorithms that are currently available can in practice, satisfy the requirements imposed on L_W, let alone find an *optimal* set of weights (in some suitable well-defined sense of the term - e.g., so as to yield minimal networks). The design of suitable threshold neuron training algorithms that (with a high probability) satisfy the requirements imposed on L_W and are at least approximately optimal remains an open research problem. Detailed theoretical and experimental analysis of the performance of single threshold neuron training algorithms is in progress [Yang *et al.*, 96].

We have conducted several experiments with MTiling on artificial as well as real-world multi-category data sets. Space does not permit a detailed discussion of the experiments. The interested reader is refered to [Parekh *et al.*, 95] for details. When a Gallant's ratchet algorithm is used for neurons, convergence to zero classification error was achieved on several non linearly separable boolean functions as well as a quantized version of iris data. The number of neurons that were added during training were far fewer than the number of training patterns in the data set in each case (suggesting that the algorithm can potentially find relatively compact networks). However, much more systematic experiments (using a broad range of data sets and other variants of single neuron training algorithms - preferably those that are guaranteed to satisfy the requirements imposed on L_W) are needed to more completely characterize the behavior of MTiling on difficult classification problems.

Since our primary focus in this paper was on provably convergent multi-category constructive learning algorithms for pattern classification, we have not addressed a number of important issues in the preceding discussion. Each constructive algorithm (including MTiling) has its own set of inductive and representational biases implicit in the design choices that determine when and where a new neuron is added and how it is trained. A systematic characterization of this bias would be quite useful in guiding the design of better constructive algorithms. Comparative analysis of performance of various constructive algorithms on a broad range of real-world data sets is currently in progress. Generalization ability of constructive learning algorithms also deserves systematic investigation. Extensions of constructive algorithms to work with multi-valued or real-valued inputs are of interest as well.

References

Burgess, N. (1994). A Constructive Algorithm That Converges for Real-Valued Input Patterns. *International Journal of Neural Systems.* Vol. 5, No 1. pp 59-66.

Chen, C-H., Parekh, R.G., Yang, J., Balakrishnan, K. and Honavar, V.G. (1995). Analysis of Decision Boundaries Generated by Constructive Neural Network Learning Algorithms. *Proceedings of the World Congress on Neural Networks.* Washington D.C. July 1995, Vol. I. pp 628-635.

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis.* New York: Wiley.

Fahlman, S. E. and Lebiere, C. (1990). The Cascade Correlation Learning Architecture. In *Neural Information Processing Systems 2.* Touretzky, D. S. (ed). Morgan-Kauffman, 1990. pp 524-532.

Frean, M. (1990). Small nets and short paths: Optimizing neural computation. Ph.D. Thesis. Center for Cognitive Science. University of Edinburgh, UK.

Gallant, S.I. (1990). Perceptron Based Learning Algorithms. *IEEE Transactions on Neural Networks.* Vol. I, No. 2, June 1990. pp 179-191.

Gallant, S. I. (1993). *Neural Network Learning and Expert Systems.* Cambridge, MA: MIT Press.

Honavar, V. (1990). Generative Learning Structures and Processes for Generalized Connectionist Networks. Ph.D. Thesis. University of Wisconsin, Madison, U.S.A.

Honavar, V. and Uhr, L. (1993). Generative Learning Structures and Processes for Connectionist Networks. *Information Sciences* 70, 75-108.

Mézard, M. and Nadal, J. (1989). Learning in feed-forward networks: The tiling algorithm. *J. Phys. A: Math. and Gen.* 22. 2191-2203.

Nilsson, N. (1965). *Learning Machines.* New York: McGraw-Hill.

Parekh, R.G., Yang, J. and Honavar, V. (1995). Multi-category Constructive Neural Network Learning Algorithms for Pattern Classification. Tech. Rep. ISU-CS-TR 95-15a. Department of Computer Science, Iowa State University, Ames, Iowa.

Poulard, H. (1995). Barycentric Correction Procedure: A Fast Method of Learning Threshold Units. In *Proceedings of the World Congress on Neural Networks 95.* Washington D.C. July 95, Vol. I. pp 710-713.

Yang, J., Parekh, R.G. and Honavar, V. (1996). Empirical Comparison of Graceful Variants of the Perceptron Learning Algorithm on Non-Separable Data Sets. In preparation.