A Survey of Learning Results for ART1 Networks

Gregory L. Heileman[†] Michael Georgiopoulos[‡] Juxin Hwang[‡]

[†]ICS Group, Department of Electrical and Computer Engineering
University of New Mexico, Albuquerque, NM 87131

[‡]Department of Electrical and Computer Engineering
University of Central Florida, Orlando, 32816

Abstract

A collection of results related to learning in ART1 networks is presented. These results are concerned primarily with the complexity of the learning process, rather than with the quality of the learned concepts. These results provide numerous insights into the operation of ART1 networks, and detail the conditions under which such networks can learn efficiently.

1 Introduction

Adaptive Resonance Theory was introduced in 1976 as a means of studying the learning process in brain-like networks [9]. A specific neural network architecture, based on Adaptive Resonance Theory, was subsequently derived by Carpenter and Grossberg [2]. This architecture, termed ART1, has been employed in various capacities in a variety of systems that require autonomous learning capabilities [1, 3, 4, 10]. In this paper we present a number of results related to the learning process in ART1. These results do not consider the quality of the pattern clustering produced by this network, but rather the time and resource complexity of the learning process. Furthermore, we have restricted our results in this paper to the ART1 network. Learning results for ARTMAP and Fuzzy ART can be found in [8, 11].

In the following section we introduce some terminology and then provide a brief overview of the ART1 network. In Section 3 we present some specific results, along with a discussion of their implications.

2 Notation and Terminology

In this section we supply a brief overview of the notation and terminology we will use to describe the ART1 network. For a more detailed description of this model see [2]. An ART1 network is composed of two layers of nodes, denoted the F_1 and F_2 layers. A node in the F_2 layer is denoted by v_i ($i = 1, 2, \ldots, M$), and a node in the F_2 layer by v_j. Every node in the F_1 layer is connected via bottom-up long term memory (LTM) traces to all of the nodes in the F_2 layer. Furthermore, every node in the F_2 layer is connected via top-down LTM traces to all of the nodes in the F_1 layer. The bottom-down and top-down LTM traces between node v_i and v_j are denoted by z_{ij} and z_{ji}, respectively. Binary (0,1) input patterns are presented at the F_1 layer of ART1. An input list (or training set) is a collection of N input patterns that we wish to learn, and a list presentation involves one presentation of each pattern in the training set to the F_1 layer. Fast learning is said to occur if we allow the limiting value of all LTM traces to be reached on every pattern presentation. In the case of top-down LTM traces this implies a value of either 0 or 1.

The initial values of the bottom-up LTM traces are chosen according to the rules specified in Section 18 of Carpenter and Grossberg [2], while top-down LTM traces can be chosen, without loss of generality, to be equal to one. The vector whose components are the top-down LTM traces emanating from a node in the F_2 layer and converging to the nodes in the F_1 layer is called a template. Since the results we only consider concern the fast learning case, and since we take all the initial values of the top-down LTM traces to be equal to one, every template can be thought of as a binary vector. We define $|I|$ and $|V_j|$ to be the size of the binary input pattern I and the binary template V_j associated with node v_j in the F_2 layer. The size of a binary vector is equal to the number of its components that have value one. Furthermore, if I is a pattern in the input list and V is a template in the F_2 layer, we define $I \cap V_j$ to be the binary vector with ones only at components where both the I and V_j components are one, and zeroes at all the other components. The reset of an active F_2 node v_j, during the presentation of an input pattern I, occurs if $|I \cap V_j| \cdot |I|^{-1} < \rho$, where ρ is the vigilance parameter in the ART1 network. An active F_2 node v_j is said to code an input pattern I on a given trial if no reset of v_j occurs after V_j is read out at the F_1 layer. In this case, the bottom-up LTM traces are adjusted according to

$$z_{ij} = \begin{cases} \frac{L}{L-1+|I \cap V_j|} & \text{if } i \in I \cap V_j \\ 0 & \text{if } i \notin I \cap V_j. \end{cases}$$

An input pattern I is said to have direct access to an F_2 node v_j if presentation of I leads at once to activation of v_j, and v_j codes I on that trial.

A node in the F_2 layer is said to be committed if it has already coded a pattern from the input list; otherwise it is called uncommitted. The templates corresponding to committed F_2 nodes are called learned templates, while the templates corresponding to uncommitted F_2 nodes are called uncommitted templates. Subset templates, with respect to an input pattern I, are templates which have value 1 only at a subset of the corresponding I components that have value 1. We say that learning in ART1 self-stabilizes in n list presentations, if subsequent presentations of the input list (i.e., list presentations $n+1, n+2, n+3, \ldots$) can neither modify already existing learned templates, nor create new learned templates by committing uncommitted templates.

3 Learning Results

The following results are valid under the ART1 conditions stated in Section 18 of [2]. One of these conditions is that fast learning occurs. In addition, we make the assumption that $1 \leq |I| \leq M - 1$.

The first result we present is important because it demonstrates that the learning process self-stabilizes in response to an arbitrary list of binary input patterns.

RESULT 1 (Carpenter & Grossberg [2]):
In response to an arbitrary list of binary input patterns, all LTM traces approach limits after a finite number of learning trials, or equivalently learning self-stabilizes in a finite number of learning trials.

The next result demonstrates that once learning has self-stabilized, there will be rapid access to learned patterns.

RESULT 2 (Carpenter & Grossberg [2]):
After learning has self-stabilized in response to an arbitrary list of binary input patterns, each input pattern either has direct access to a node in the F_2 layer that possesses the largest subset template with respect to the input pattern, or the input pattern cannot be coded by any node in the F_2 layer. In the latter case, the F_2 layer contains no uncommitted nodes.

Note that there is the possibility that an input pattern cannot be coded by any node in the F_2 layer. This is certainly possible if there are more patterns than F_2 layer nodes, but leaves open the question of whether this may occur if there are as many F_2 layer nodes as patterns.

We now present three important properties of ART1 templates:

1. As a consequence of each pattern presentation, any given template will either stay the same size, or will be modified so that it has a smaller size.

2. Equal templates (i.e., identical binary vectors) cannot be created.

3. If no templates are modified during any given list presentation, then learning has self-stabilized.

The following results follow immediately from these properties.

RESULT 3 (Moore [12]):
In response to an arbitrary list of binary input patterns, the maximum number of learned templates created is equal to 2^M.

This result is derived by noting that since each template is a subset of at least one of the input patterns, and each must be distinct (template property 2), the maximum number of templates is bounded by the maximum number of binary input patterns, i.e., 2^M.

> **RESULT 4** (Moore [12]):
> In response to an arbitrary list of binary input patterns, learning self-stabilizes (i.e., new templates cannot be created and the already existing learned templates cannot be modified) after at most $M2^M$ list presentations.

This results also follows via a simple counting argument: If no learning on a given list presentation occurs, then learning has stabilized (template property 3). If, on the other hand, some template is altered on a given list presentation, then it is reduced in size (template property 1). Any template can be reduced a maximum of M times, and since there are 2^M possible templates (Result 3), the maximum number of list presentations is $M2^M$.

If the learning parameter L is chosen appropriately, then a much better bound than the one given in Result 4 can be obtained. This leads to Result 5.

> **RESULT 5** (Georgiopoulos, Heileman, & Hwang [6]):
> If L is small (i.e., $\leq |I|^{-1} + 1$) and the F_2 layer has at least N nodes, then each member of a list of N binary input patterns, which is presented at the F_1 layer, will have direct access to an F_2 layer node after at most m list presentations, where m is the number of distinct size patterns in the input list.

It is interesting to note that Results 1–5 do not require the input list to be presented cyclically, i.e., the patterns need not remain in the same order on each list presentation. A related result (which also makes the assumption that L is small) states that any pattern I of size $\leq x-1$ will have direct access to a stable template that was created in list presentations $\leq x - 1$ [5]. Result 5, however, is stronger since it only depends on the relative sizes of the of the patterns in the inputs list. More precisely, it was shown in [6] that if the members of the input list are grouped according to their size into the sets $S_1, S_2, \ldots, S_m$, with S_1 containing the smallest size patterns and S_m the largest, then in list presentations $\geq x$ $(2 \leq x \leq m+1)$:

- Every $I \in S_{x-1}$ has access to a subset learned template V that can code I, and V is created in list presentations $\leq x - 1$.
- The presentation of $I \in S_{x-1}$ can neither create new templates, nor modify already existing learned templates.

An immediate consequence of Result 5 follows from the fact that we typically know N the number of patterns in the input list, as well as M the number of nodes in the F_1 layer. In these cases, we can say that learning in ART1 stabilizes in at most $\min(N, M-1)$ list presentations. Even more interesting is the fact that if the input patterns are complement coded, then this result guarantees that all patterns can be learned in a single list presentation. Because this bound is so much better than the one provided in Result 4, the question of the feasibility of small L values becomes important. The parameter L must be chosen in this case less than or equal to $|I|^{-1}+1$. The only difficulty arises when the pattern size is extremely large. This would lead to a very small value for $|I|^{-1}$, which in turn may lead to difficulties when trying to accurately represent L on a digital computer.

The following two results were derived after studying the N–N–N conjecture. This conjecture, posed by Carpenter and Grossberg [2], states that in the fast learning case, if the F_2 layer in ART1 has at least N nodes, then each member of a list of N input patterns presented cyclically at the F_1 layer will have direct access to an F_2 layer node after at most N list presentations.

The next result demonstrates that the N–N–N conjecture is not valid for certain large L values.

> **RESULT 6** (Georgiopoulos, Heileman, & Hwang [7]):
> For large L values $(> |I|^{-1}+1)$, if ART1 is presented with an arbitrary list of binary input patterns, then after learning has self-stabilized: (i) the number of learned templates may be greater than the number of patterns in the input list, and (ii) the number of pattern presentations may have exceeded N.

This result is valid independent of whether the list is presented cyclically. Thus, it is easy to see that this result violates the N–N–N conjecture since in case (i) more than N F_2 layer nodes may be required to represent the learned templates, and in case (ii) more than N list presentations may be required for learning to self-stabilize. We emphasize that Result 6 does not apply when L assumes small values. Furthermore, Result 5 can be used to show that the N–N–N conjecture is valid for small L values. Since the maximum number of distinct-size binary patterns cannot exceed N, the number of patterns in the list, the maximum number of list presentations in this case will be N. It is worth noting the even if L assumes large values, it is difficult to come up with examples that violate the N–N–N conjecture, unless they are constructed specifically with that purpose in mind. Nevertheless, the best worst-case bound available in this case is $M2^M$ list presentations as discussed in Result 4.

The final result we present is independent of the size of L.

> **RESULT 7** (Georgiopoulos, Heileman, & Hwang [6]):
> If ART1 is cyclically presented with an arbitrary list of binary input patterns, then after learning has self-stabilized, there may exist committed nodes that are not directly accessed by any pattern in the input list.

Thus, there is the possibility of wasted resources in the ART1 network. The question of how many such "wasted" committed nodes are possible is still an open problem.

Acknowledgments: This research was partially supported by grants from Boeing Computer Services under Contract W-300445, and the Florida High Technology and Industry Council.

References

[1] A. A. Baloch and A. M. Waxman. Visual learning, adaptive expectations, and learning behavioral conditioning of the mobil robot MAVIN. *Neural Networks*, 4(3):271–302, 1991.

[2] G. A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54–115, 1987.

[3] G. A. Carpenter, S. Grossberg, and J. H. Reynolds. ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, 4(5):565–588, 1991.

[4] T. Caudell, S. D. Smith, R. Escobedo, and G. C. Johnson. A neural database system for reusable engineering. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, Singapore, 1991.

[5] M. Georgiopoulos, G. L. Heileman, and J. Huang. Convergence properties of learning in ART1. *Neural Computation*, 2(4):502–509, 1990.

[6] M. Georgiopoulos, G. L. Heileman, and J. Huang. Properties of learning related to pattern diversity in ART1. *Neural Networks*, 4(6):751–757, 1991.

[7] M. Georgiopoulos, G. L. Heileman, and J. Huang. The N–N–N conjecture in ART1. *Neural Networks*, 5(5):745–753, 1992.

[8] M. Georgiopoulos, J. Huang, and G. L. Heileman. Properties of learning in ARTMAP. to appear in *Neural Networks*.

[9] S. Grossberg. Adaptive pattern recognition and universal recoding II: Feedback, expectation, olfaction, and illusions. *Biological Cybernetics*, 23:187–202, 1976.

[10] M. J. Healy, T. P. Caudell, and S. D. G. Smith. A neural architecture for pattern sequence verification through inferencing. *IEEE Transactions on Neural Networks*, 4(1):9–20, 1993.

[11] J. Huang, M. Georgiopoulos, and G. L. Heileman. Fuzzy ART properties. submitted to *Neural Networks*.

[12] B. Moore. ART1 and pattern clustering. In D. S. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proceedings of the 1988 Connectionist Summer School*, pages 174–185, San Mateo, CA, 1989. Morgan Kaufmann.