The N–N–N Conjecture in ART1

M. Georgiopoulos *, G. L. Heileman **, J. Huang *

* Electrical Engineerring Department, Univ. of Central Florida
** Elec. and Comp. Engineering Department, Univ. of New Mexico

Abstract

In this paper we consider the ART1 neural network architecture introduced by Carpenter and Grossberg. In their original paper, Carpenter and Grossberg made the following conjecture: 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 of ART1 will have direct access to an F_2 layer node after at most N list presentations. In this paper, we demonstrate that the conjecture is not valid for certain large L values, where L is a network parameter associated with the adaptation of the bottom-up traces in ART1. It is worth noting that previous work has shown the conjecture to be true for small L values.

1 Introduction

A neural network architecture that can be used to learn recognition categories was derived and analyzed by Carpenter and Grossberg [1]. This architecture was termed ART1 in reference to the adaptive resonance theory introduced by Grossberg [2]. The ART1 architecture self-organizes and self-stabilizes its recognition codes in response to presentations with arbitrary ordering of arbitrarily many and arbitrarily complex binary (0,1) input patterns. We assume that the reader is familiar with the ART1 architecture introduced by Carpenter and Grossberg (1987), as well as the terminology and theorems contained therein.

Carpenter and Grossberg [1] specified in the form of a conjecture the number of list presentations and F_2 layer nodes required by ART1 to learn and recognize a list of binary input patterns through direct access. Carpenter [3] refers to this conjecture as the *N–N–N Conjecture*. It is formally stated below.

The N–N–N Conjecture:

In the fast learning case, if the F_2 layer of ART1 has at least N nodes, each member of a list of N input patterns, which is cyclically presented at the F_1 layer of ART1, will have direct access to an F_2 layer node after at most N list presentations.

Georgiopoulos et al [4] proved the following result when the L parameter is small. Note that L is a parameter associated with the adaptation of the bottom-up traces in ART1.

Result 1:

In the fast learning case, if L is small and if the F_2 layer of ART1 has at least N nodes, each member of a list of N binary input patterns, which is cyclically presented at the F_1 layer of ART1, 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; the size of a pattern is defined to be the number of its components that are one.

The smallness of the parameter L that makes Result 1 valid is dictated by the largest size pattern in the input list. In particular, L has to be less than or equal to $1 + |I_{\max}|^{-1}$, where $|I_{\max}|$ denotes the size of the largest pattern in the input list. The validity of Result 1 for small L values implies the validity of the N–N–N Conjecture for small L values. Note though, that the N–N–N Conjecture does not impose any constraints on the range of the values for the parameter L beyond the ones imposed in Section 18 (i.e., $L > 1$) of Carpenter and Grossberg [1]. In this paper we demonstrate, by presenting two carefully chosen examples, that there exist large L values (i.e., $L \in (1 + |I_{\max}|^{-1}, \infty)$) for which the N–N–N Conjecture is not true. Obviously, for these L values, Result 1 is not true either.

2 Preliminaries

We denote a node in the F_1 layer of ART1 by v_i $(i = 1, 2, \ldots, M)$ and a node in the F_2 layer of ART1 by v_j $(j = M+1, M+2, \ldots)$. Every node in the F_1 layer is connected via *bottom up traces*, denoted by z_{ij}, to all of the nodes in the F_2 layer. Furthermore, every node in the F_2 layer is connected via *top-down traces*, denoted by z_{ji}, to all of the nodes in the F_1 layer. Input patterns are presented at the F_1 layer of ART1. Initial values of the bottom-up and top-down traces, denoted by $z_{ij}(0)$ and $z_{ji}(0)$, correspond to the values of the bottom-up and top-down traces prior to the presentation of any input pattern at the F_1 layer of ART1. The $z_{ij}(0)$'s are chosen according to the rules specified in Section 18 of Carpenter and Grossberg ([1]), while the $z_{ji}(0)$'s can be chosen, without loss of generality, to be equal to one.

The vector whose components are the top down traces emanating from a node v_j in the F_2 layer and converging to the nodes v_i $(i = 1, 2, \ldots M)$ in the F_1 layer is called *template* V_j (i.e., $V_j = (z_{j1}, z_{j2}, \ldots z_{jM})$). Since this paper concerns the fast learning case, and since we take all the initial values of the top-down traces to be equal to one, every template V_j $(j = M+1, M+2, \ldots)$ can be thought of as a binary (0,1) vector. We define $|I|$ and $|V_j|$ to be the *size* of the binary input pattern I and the binary template V_j, respectively. As mentioned in the introduction, 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_j 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.

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. 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 the template V_j is read out at the F_1 layer. *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 a parameter in the ART1 network, called *vigilance*.

A node in the F_2 layer is called *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*. 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 The N–N–N Conjecture

The parameters that we have at our disposal to construct examples that violate the N–N–N Conjecture are: N, M, ρ, L, and M_u. N denotes the number of patterns in the input list, M denotes the number of nodes at the F_1 layer, ρ is the vigilance parameter, and L is a network parameter associated with the adaptation of bottom-up traces. The parameter M_u, in conjunction with L, restricts the range of the initial values for the bottom-up traces. In particular,

$$0 < z_{ij}(0) < \frac{L}{L - 1 + M_u}, \tag{1}$$

where M_u is a real number greater than or equal to M. The above inequality when $M_u = M$ is referred to by Carpenter and Grossberg ([1]) as the *direct access inequality*.

The key parameter that allows us to construct examples that violate the N–N–N Conjecture is L. This is obvious from the discussion in Section 1, where we stated that the N–N–N Conjecture is valid for small L values (see the discussion after Result 1). In this section we will show that the N–N–N Conjecture is not true for large L values. First let us state two additional conjectures, that are related with the N–N–N Conjecture.

The N–N–∞ Conjecture:

In the fast learning case, if the F_2 layer of ART1 has at least N nodes, each member of a list of N input patterns, which is presented cyclically at the F_1 layer of ART1, will have direct access to an F_2 layer node after finitely many ($< \infty$) list presentations.

The N–∞–N Conjecture:

In the fast learning case, if the F_2 layer of ART1 has infinitely many nodes, each member of a list of N input patterns, which is presented cyclically at the F_1 layer of ART1, will have direct access to an F_2 layer node after at most N list presentations.

The N–N–∞ and the N–∞–N Conjectures are weaker than the N–N–N Conjecture in the sense that if either one of them is not true, then the N–N–N Conjecture is not true. Examples 1 and 2, presented below, violate the N–N–∞ and the N–∞–N Conjectures, respectively; consequently they violate the N–N–N Conjecture. In these examples we assume that the initial bottom-up traces are chosen to satisfy inequality (1). For ease of exposition, we also assume that the competition amongst uncommitted nodes in the F_2 layer always results in the node labeled with the smallest index being chosen as the winner.

3.1 Example 1

The input list in this example consists of the following patterns:

$$I_1 = 00000111$$
$$I_2 = 00001100$$
$$I_3 = 11111100.$$

The above input patterns are presented in the order $I_1 I_2 I_3$. The order of pattern presentation is kept fixed from list presentation to list presentation. The number of nodes M in the F_1 layer, the vigilance parameter ρ, the parameter L, and the parameter M_u are equal to 8, 0.4, 5.5 and 12.0, respectively. In this example, the F_2 layer has three nodes, denoted v_9, v_{10} and v_{11}. Note that Example 1 satisfies the assumptions of the N–N–∞ Conjecture for $N = 3$. In Figure 1, the template formation corresponding to this example is depicted for the first two list presentations. In the same figure, the input patterns and templates formed after the first two list presentations are depicted as sequences of open circles ($\circ$) and full circles ($\bullet$). An open circle designates a zero, while a full circle designates a one. In Figure 1 only the templates corresponding to committed F_2 nodes (i.e., learned templates) are shown. The template formation shown in Figure 1 can be easily verified for the input patterns and the network parameters (i.e., M, ρ, L, and M_u) chosen in Example 1. Observe that in Example 1, learning self-stabilizes in two list presentations. After the second list presentation (i.e., in list presentations $3, 4, 5, \ldots$) pattern I_1 has direct access to node v_{11}, pattern I_2 has direct access to node v_{10}, while pattern I_3 does not have direct access to any node in the F_2 layer. Actually, after the second list presentation, pattern I_3 activates nodes v_{10}, v_9 and v_{11} in that order, and it resets each one of them.

Example 1 demonstrates that the N–N–∞ Conjecture is not valid. As a result, the N–N–N Conjecture is not valid either. The N–N–∞ and N–N–N Conjectures fail in Example 1 because there are not enough nodes in the F_2 layer to code every input pattern. Note that if the F_2 layer in this example had one more node (i.e., node v_{12}), then pattern I_3 would have activated this node in the second list presentation and would have been coded by this node. Furthermore, in the case where the F_2 layer has four nodes, learning self-stabilizes in two list presentations, and after the second list presentation each one of the input patterns I_1, I_2 and I_3 has direct access to a node in the F_2 layer (i.e., pattern I_1 to node v_{11}, pattern I_2 to node v_{10} and pattern I_3 to node v_{12}).

It seems from Example 1 that if we have enough nodes in the F_2 layer of ART1, then at most N list presentations are enough to guarantee direct access to an F_2 node for each of the N patterns in the input list. Example 2, by violating the N–∞–N Conjecture, disputes the validity of the above statement, and consequently the validity of the N–N–N Conjecture.

3.2 Example 2

The input list in this example consists of the following patterns:

$$I_1 = 11111111111100$$
$$I_2 = 00000000010111$$
$$I_3 = 00000000101100.$$

The input patterns are presented in the order $I_1 I_2 I_3$. The order of pattern presentation is kept fixed from list presentation to list presentation. The number of nodes M in the F_1 layer, the vigilance parameter ρ, the parameter L, and the parameter M_u are equal to 14, 0.3, 6 and 30.0, respectively. In this example, the F_2 layer of ART1 has infinitely many nodes. These nodes are denoted v_{15}, v_{16}, v_{17}, v_{18}, ... Note that Example 2 satisfies the assumptions of the N–∞–N Conjecture for $N = 3$. In Figure 2, the template formation corresponding to this example is depicted for the first four list presentations. In the same figure, the input patterns and templates formed after the first four list presentations are again depicted as sequences of open and full circles, and only the templates corresponding to committed F_2 nodes (i.e., learned templates) are shown. The template formation shown in Figure 2 can be easily verified for the input patterns and the network parameters (i.e., M, ρ, L, and M_u) chosen in Example 2.

We see, by observing Figure 2, that pattern I_1 in the fourth list presentation does not have direct access to any node in the F_2 layer. In particular, pattern I_1 in the fourth list presentation initially activates and resets node v_{17}, then it might activate and reset nodes v_{16} and v_{15} and finally it activates and is coded by node v_{18}.

Example 2 demonstrates that the N–∞–N Conjecture is not valid. Consequently, the N–N–N Conjecture is not valid either. The reason that the N–∞–N Conjecture fails is that more than N list presentations are required before all patterns have direct access to an F_2 node. The reason that the N–N–N Conjecture fails is that more than N list presentations, and more than N nodes in the F_2 layer are required for all patterns to have direct access to an F_2 node. It is worth noting that in Example 2 learning self-stabilizes in four list presentations, and after the fourth list presentation each one of the input patterns has direct access to a node in the F_2 layer (i.e., pattern I_1 to node v_{18}, pattern I_2 to node v_{16} and pattern I_3 to node v_{17}).

4 Conclusions

In the previous sections we presented Examples 1 and 2 that violated the N–N–N Conjecture for large L values. In particular, Example 1 violated the N–N–∞ and N–N–N Conjectures, while Example 2 violated the N–∞–N, the N–N–∞, and the N–N–N Conjectures. It is worth pointing out that we could not devise an example that violates the N–∞–N conjecture without violating at the same time the N–N–∞ Conjecture. The purpose of of Example 1 was to demonstrate that there are cases in which the N–N–∞

Conjecture is violated without violating the N-∞-N Conjecture. Let us now present some final conclusions regarding properties of learning in ART1 which are immediate byproducts of Examples 1 and 2.

Conclusion 1:
In the fast learning case and for large L values, 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.

Conclusion 2:
In the fast learning case and for large L values, if ART1 is cyclically presented with an arbitrary list of binary input patterns, then after learning has self-stabilized the number of learned templates may be greater than the number of patterns in the input list.

Conclusion 1 is true for Examples 1 and 2 (in Example 1 the committed node v_9, after learning has self-stabilized, is not directly accessed by any pattern in the input list). Conclusion 2 is true for Example 2 (in Example 2, after learning has self-stabilized, there are four learned templates and only three input patterns). It is worth noting that Conclusion 1 is also true for small L values (see [4]). On the contrary, Conclusion 2 is not true for small L values . For small L values, after learning has self-stabilized, the number of learned templates is always smaller than or equal to the number of patterns in the input list (for a proof see [4]).

Acknowledgements: This research was supported in part by a grant from the Florida High Technology and Industry Council, and in part by a grant from Sandia National Laboratories.

References

[1] 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, 1987, pp. 54-115.

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

[3] G. A. Carpenter, "Neural Networks: From Foundations to Applications," *Short Course*, Wang Institute of Boston, Tyngsboro, MA, May 1991.

[4] M. Georgiopoulos, G. L. Heileman and J. Huang, "Properties of learning related to pattern diversity in ART1," *Neural Networks*, 4, 1991, pp. 751-757.

Pattern	First List		Second List		
	V_9	V_{10}	V_9	V_{10}	V_{11}
I_1	I_1		$I_1 \cap I_2$	I_3	I_1
I_2	$I_1 \cap I_2$		$I_1 \cap I_2$	$I_3 \cap I_2$	I_1
I_3	$I_1 \cap I_2$	I_3	$I_1 \cap I_2$	$I_3 \cap I_2$	I_1

(a)

Pattern or Template	Pictorial Representation
I_1	○ ○ ○ ○ ○ ● ● ●
I_2	○ ○ ○ ○ ● ● ○ ○
I_3	● ● ● ● ● ● ○ ○
V_9	○ ○ ○ ○ ○ ● ○ ○
V_{10}	○ ○ ○ ○ ● ● ○ ○
V_{11}	○ ○ ○ ○ ○ ● ● ●

(b)

Figure 1

Pattern	First List	Second List	
	V_{15}	V_{15}	V_{16}
I_1	I_1	$I_1 \cap I_2 \cap I_3$	I_1
I_2	$I_1 \cap I_2$	$I_1 \cap I_2 \cap I_3$	$I_1 \cap I_2$
I_3	$I_1 \cap I_2 \cap I_3$	$I_1 \cap I_2 \cap I_3$	$I_1 \cap I_2$

Pattern	Third List			Fourth List			
	V_{15}	V_{16}	V_{17}	V_{15}	V_{16}	V_{17}	V_{18}
I_1	$I_1 \cap I_2 \cap I_3$	$I_1 \cap I_2$	I_1	$I_1 \cap I_2 \cap I_3$	$I_1 \cap I_2$	$I_1 \cap I_3$	I_1
I_2	$I_1 \cap I_2 \cap I_3$	$I_1 \cap I_2$	I_1	$I_1 \cap I_2 \cap I_3$	$I_1 \cap I_2$	$I_1 \cap I_3$	I_1
I_3	$I_1 \cap I_2 \cap I_3$	$I_1 \cap I_2$	$I_1 \cap I_3$	$I_1 \cap I_2 \cap I_3$	$I_1 \cap I_2$	$I_1 \cap I_3$	I_1

(a)

Pattern or Template	Pictorial Representation
I_1	● ● ● ● ● ● ● ● ● ● ● ● ○ ○
I_2	○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○ ● ● ●
I_3	○ ○ ○ ○ ○ ○ ○ ○ ● ○ ● ● ○ ○
V_{15}	○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○ ○
V_{16}	○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○ ● ○ ○
V_{17}	○ ○ ○ ○ ○ ○ ○ ○ ● ○ ● ● ○ ○
V_{18}	● ● ● ● ● ● ● ● ● ● ● ● ○ ○

(b)

Figure 2