Communicated by Gail Carpenter

Convergence Properties of Learning in ART1

Michael Georgiopoulos
Department of Electrical Engineering,
University of Central Florida, Orlando, FL 32816 USA

Gregory L. Heileman
Department of Electrical and Computer Engineering,
University of New Mexico, Albuquerque, NM 87131 USA

Juxin Huang
Department of Electrical Engineering,
University of Central Florida, Orlando, FL 32816 USA

We consider the ART1 neural network architecture. It is shown that in the fast learning case, an ART1 network that is repeatedly presented with an arbitrary list of binary input patterns, self-stabilizes the recognition code of every size-l pattern in at most l list presentations.

1 Introduction

A neural network architecture for the learning of recognition categories was derived by Carpenter and Grossberg (1987). This architecture was termed ART1 in reference to the *adaptive resonance theory* introduced by Grossberg (1976). It was shown in Carpenter and Grossberg (1987) that ART1 self-organizes and self-stabilizes its recognition codes in response to arbitrary orderings of arbitrarily many and arbitrarily complex binary input patterns.

In this paper, only the fast learning case is considered. We show that if ART1 is repeatedly presented with a list of binary input patterns it self-stabilizes the recognition code of every size-l pattern in at most l list presentations. (A size-l input pattern is a binary vector containing l components with value one and the remaining components with value zero.) An immediate consequence of this result is that if the input patterns in the input list can be represented by binary vectors of dimensionality M, with the size-0 and size-M vectors excluded from the list (one of our modeling assumptions in Section 2), then ART1 learns and recognizes the list in at most $M - 1$ list presentations. This result is valid independent of the ordering with which the input patterns are presented within the list.

Neural Computation **2, 502–509** (1990)

In short, this paper provides useful upper bounds on the number of list presentations required to learn a list of input patterns presented repeatedly to ART1. The modeling assumptions are presented in Section 2. In the same section the tightness of the upper bounds is exploited by examining two extreme examples. In Section 3, the results are stated and proven. Concluding remarks are contained in Section 4.

2 Model — Preliminaries

A complete description of ART1 and the theorems that give insight into its operation are provided in Carpenter and Grossberg (1987). An ART1 network consists of two layers of neurons (nodes), called the F_1 and F_2 layers. Input patterns are presented at the F_1 layer. Every node in the F_1 layer is connected via bottom-up traces to all of the nodes in the F_2 layer. Every node in the F_2 layer is likewise connected to all of the nodes in the F_1 layer via top-down traces. The results of this paper are proven under the following assumptions:.

A1: All hypotheses of section 18 in Carpenter and Grossberg (1987) hold (one of these hypotheses is that fast learning occurs)

A2: $L - 1 \leq |I|^{-1}$

A3: $1 \leq |I| \leq M - 1$

A4: F_2 has enough nodes to code all the patterns at every presentation of the input list

where $|I|$ is the size of an arbitrary pattern I in the input list, M is the number of nodes in the F_1 layer, and L is a parameter associated with the adaptation of bottom-up and top-down traces in the ART1 neural network architecture.

The top-down traces that emanate from a node in the F_2 layer are called templates. Assume that a pattern I which belongs to a list of binary input patterns is presented to ART1. Furthermore, assume that at the nth presentation of the list, pattern I activates some node k in the F_2 layer and furthermore k codes I. We denote by $V_{I,n}^{+}$ the template that corresponds to node k after k has learned I. We say that I is coded by $V_{I,n}^{+}$ or that $V_{I,n}^{+}$ has coded the pattern I; $V_{I,n}^{+}$ is referred to as a learned template. To prove our results, the templates of ART1 need to be considered either prior to a pattern's presentation, or after a template has coded a pattern. For the purposes of the results discussed in this paper, the ART1 templates can always be thought of as binary vectors. Actually, when the top-down trace of a template is taken as either zero or one it means that the trace is either small enough, or large enough to satisfy the 2/3 Rule of the ART1 network (for more details see Carpenter and Grossberg 1987).

Consider a pattern I in the list and a template V corresponding to an F_2 node. There is a one-to-one correspondence between the components of the binary vectors I and V. A component of I corresponds to a component of V if both of them activate the same F_1 node. We define, as in Carpenter and Grossberg (1987), three types of learned templates with respect to an input pattern I: *subset* templates, *superset* templates, and *mixed* templates. The components of a subset template V satisfy $V \subseteq I$. They are one only at a subset of the corresponding I components. The components of a superset template V satisfy $V \supset I$. They are one at all the corresponding components of I that are one, as well as at some components of I that are zero. The components of a mixed template V are one at some, but not all of the corresponding I components, as well as at some of the components of I that are zero. In this case, the set of the V components that are one is neither a subset nor a superset of the set of the I components that are one. Sometimes it is convenient to refer to a pattern I as being a subset, superset or mixed pattern with respect to a template V if $I \subseteq V$, $I \supset V$, or V is a mixed template with respect to I. Besides the learned templates described above, we also define a template V to be an *uncommitted* template if it corresponds to a node that has not coded any pattern yet. We assume that the components of an uncommitted template consist of all ones.

Since an input pattern I is a binary vector and a template V can be thought of as a binary vector, we define by $|I|$ and $|V|$ the *size* of the binary vectors I and V, respectively. We also define a template V to be a *stable template* if and only if, after its creation, it cannot be destroyed by future pattern presentations. We say, as in Carpenter and Grossberg (1987), that a pattern I has *direct access* to template V if presentation of I leads at once to activation of the F_2 node with corresponding template V, and this template codes I on that trial. Finally, if I is a pattern of the input list and V is a template of ART1, we define $I \cap V$ as the binary vector with ones only at components where both the I and V components are one, and zeros at all other components.

Let us now present two examples that are extreme cases and demonstrate clearly the tightness of the bounds mentioned in Section 1. To follow these examples the reader needs to be aware of Theorems 1 and 7 in Carpenter and Grossberg (1987).

In the first example, ART1, with a vigilance parameter $\rho = 1$ is repeatedly presented with a nested list of input patterns in order of decreasing size. In particular, the input list, $\{I_1, I_2, \ldots, I_{M-1}\}$, is such that $I_1 \subset I_2 \subset \ldots \subset I_{M-1}$ with $|I_k| = k$, and it is presented in order $I_{M-1}, I_{M-2}, \ldots, I_1, I_{M-1}, I_{M-2}, \ldots, I_1$, etc. Then, in the first list presentation only template $V_1 = I_1$ is created (see Theorem 7 of Carpenter and Grossberg 1987). Template V_1 cannot be destroyed thereafter because all patterns are supersets or equal to template V_1. Hence, template V_1 is a stable template. In list presentations ≥ 2 pattern I_1 will have direct access to template V_1 (see Theorem 1 in Carpenter and Grossberg 1987). As a result, the recognition code of pattern I_1 (i.e., V_1) self-stabilizes in exactly one list presentation. In the second list presentation only template

$V_2 = I_2$ is created. Template V_2 cannot be destroyed thereafter because all patterns other than pattern I_1 are supersets or equal with V_2, and pattern I_1 is coded by the stable template V_1. In list presentations ≥ 3 pattern I_2 will have direct access to template V_2. Hence, the recognition code of pattern I_2 (i.e., V_2) self-stabilizes in exactly two list presentations. Working similarly for the rest of the input patterns we can prove that ART1 self-stabilizes the code of a size-l ($3 \leq l \leq M-1$) pattern in exactly l list presentations. This example corresponds to the extreme case where the upper bound on the number of list presentations required by ART1 to self-stabilize the recognition codes of size-l patterns is attained.

In the second example, ART1, with a vigilance parameter $\rho = 1$, is presented with a nested list of input patterns in order of increasing size. The input list, $\{I_1, I_2, \ldots, I_{M-1}\}$, is such that $I_1 \subset I_2 \subset \ldots \subset I_{M-1}$ with $|I_k| = k$, and it is presented in order $I_1, I_2, \ldots, I_{M-1}, I_1, I_2, \ldots, I_{M-1}$, etc. Then, in the first list presentation the templates $V_l = I_l$, $1 \leq l \leq M-1$ are created. In the second list presentation pattern I_l, $1 \leq l \leq M-1$, will have direct access to template V_l, $1 \leq l \leq M-1$. As a result, ART1 self-stabilizes the code of every pattern in the input list in exactly one list presentation. This example is another extreme case where the number of list presentations required by ART1 to self-stabilize the recognition codes of size-l patterns attains its lowest possible value (i.e., one list presentation).

Carpenter and Grossberg (1987) made the following conjecture: Under their hypotheses of section 18, if F_2 has at least N nodes, then each member of a list of N input patterns that is presented cyclically to ART1 will have direct access to an F_2 node after at most N list presentations. In this paper, under assumptions A1 through A4 we prove a much stronger result, at least for most cases of interest. The result states that the size of the pattern determines the upper bound on the number of pattern presentations required by ART1 to learn the pattern. In particular, a size-l ($1 \leq l \leq M-1$) pattern requires at most l list presentations. One of the cases where the conjecture is stronger corresponds to the situation where the input list contains N patterns with $N < M-1$. Considering though that N is an integer between 2 and $2^M - 2$ (patterns of size-0 or size-M are excluded), the result of this paper is stronger than the conjecture for most cases of interest.

3 Results

We first state two Lemmas that are going to be useful for the proof of our results. Lemma 1 is valid under assumptions A1–A3.

Lemma 1. *Suppose that I is an arbitrary pattern from the input list. Learned subset templates with respect to I are searched first in order of decreasing size (i.e., the closest learned subset template to I is searched first, and if it is reset, the next closest subset template to I is searched and so on). If all learned* *subset templates are reset, then superset and mixed learned templates, as well as uncommitted templates are searched, not necessarily in that order.*

Lemma 1 is a shortened restatement of Carpenter and Grossberg's (1987) Theorem 7 and its proof can be found there. Let us now assume that an input pattern I is presented at F_1. The activity at F_1 changes from 0 to I. Let us also assume that a node in F_2 with template V_1 is searched first. The activity at F_1 changes to $I \cap V_1$. If $|I \cap V_1||I|^{-1} \geq \rho$ then template V_1 codes pattern I. If $|I \cap V_1||I|^{-1} < \rho$, then the node with template V_1 is reset and another node in F_2 is searched. The parameter ρ, called vigilance, determines whether the top-down template of an F_2 node is a good match of the input pattern I. It is obvious by the description of this reset mechanism, that if a template V_1 is searched first and reset (i.e., $|I \cap V_1||I|^{-1} < \rho$) then any other template V_2 that is searched later will be reset if $|I \cap V_2||I|^{-1} \leq |I \cap V_1||I|^{-1}$.

Lemma 2 is an immediate consequence of Lemma 1 and the above discussion. Lemma 2 is valid under assumptions A1–A3.

Lemma 2. *Suppose that I is an arbitrary pattern from the input list, V_1 is a learned subset template (with respect to I), and V_2 is an arbitrary mixed learned template (with respect to I), prior to I's presentation. Then, if V_1 is reset and V_2 is searched, V_2 will be reset if*

$$\frac{|I \cap V_2|}{|I|} \leq \frac{|I \cap V_1|}{|I|}$$

Our results are now presented in a form of a theorem. The theorem is valid under assumptions A1–A4.

Theorem 1. *Consider an arbitrary list of binary input patterns that is repeatedly presented to ART1. Then, in list presentations $\geq x$, where $x \geq 2$:*

T1: *A pattern I of size $\geq x$ cannot be coded by a mixed template V, such that $|I \cap V| \leq x - 1$.*

T2: *A pattern I of size $\leq x - 1$ will have direct access to a stable template that has been created in list presentations $\leq x - 1$.*

T1 is obviously true for $x \geq M$, because according to the assumptions of the theorem there are no patterns of size $\geq M$ in the input list. Hence, if we prove T1 for $2 \leq x \leq M-1$ we have proven T1 for all x. Furthermore, it is easy to see that if we prove T2 for $2 \leq x \leq M$ we have proven T2 for all x. We will prove T1 for $2 \leq x \leq M - 1$ and T2 for $2 \leq x \leq M$ in two steps. In step 1, we prove that T1 and T2 are valid for $x = 2$. In step 2, we will show that for every n, $3 \leq n \leq M$, the assumption that T1 and T2 are valid for all $2 \leq x \leq n - 1$ implies their validity for $x = n$. This iterative procedure guarantees the validity of T1 and T2 for all x, such that $2 \leq x \leq M$, and consequently the validity of the theorem for all $x \geq 2$.

Step 1. Prove that T1 and T2 are valid for $x = 2$.

Consider a pattern I of size ≥ 2. At all times, prior to I's appearance in list presentations ≥ 2, there exists a learned subset template V of I. Hence, according to Lemmas 1 and 2, I cannot be coded by a mixed template of size ≤ 1. This proves T1 at $x = 2$.

Now assume that a pattern I of size-1 has been coded by the template $V_{I,1}^{+}$ in the first list presentation. $V_{I,1}^{+}$ cannot be destroyed thereafter; hence, $V_{I,1}^{+}$ is a stable template. Furthermore, after the creation of $V_{I,1}^{+}$ no other template equal to $V_{I,1}^{+}$ can be created (see Lemmas 1 and 2). As a result, in list presentations ≥ 2, the size-1 pattern I will have direct access to its equal $V_{I,1}^{+}$ template (see Lemma 1). The stable template $V_{I,1}^{+}$ was created in the first list presentation. This proves T2 at $x = 2$.

Step 2. Pick an n such that $3 \leq n \leq M$ and assume that T1 and T2 are valid for every x, such that $2 \leq x \leq n - 1$. It will now be shown that T1 and T2 are true for $x = n$.

Proof of T1 at $x = n$**.** Consider a pattern I of size $|I| \geq n$. Assume that I was coded by $V_{I,n-1}^{+}$ in list presentation $n - 1$ and $|V_{I,n-1}^{+}| = l$. Two cases are distinguished:

(a) $l \leq n - 1$. The template $V_{I,n-1}^{+}$ can be destroyed by either (1) a size-k ($k < l$) pattern, or by (2) a mixed pattern $\hat{I}$ that is coded by $V_{I,n-1}^{+}$, where $|\hat{I} \cap V_{I,n-1}^{+}| = k$ ($k < l$). All size-k ($k < l \leq n - 1$) patterns have direct access to stable templates that have been created by the end of list presentation $n - 2$. This is due to the validity of T2 for all x such that $2 \leq x \leq n-1$; hence, (1) cannot happen. Furthermore, in list presentations $\geq n - 1$, (2) cannot happen either, due to the validity of T1 for all x such that $2 \leq x \leq n - 1$. As a result, the $V_{I,n-1}^{+}$ template of size $l \leq n - 1$ is stable, and pattern I will be coded in list presentations $\geq n$ by the subset template $V_{I,n-1}^{+}$, or by some other subset template of size larger than or equal to the size of $V_{I,n-1}^{+}$ (see Lemma 1). In short, I cannot be coded by any mixed template.

(b) $l \geq n$. The template $V_{I,n-1}^{+}$ can be refined to a size-k template, $k \geq n - 1$, prior to I's appearance in future list presentations; k cannot be smaller than $n - 1$ due to the validity of T1 and T2 at all s such that $2 \leq s \leq n - 1$. So, in list presentations $\geq n$, I will always have access to a subset template of size at least $n - 1$. Hence, in list presentations $\geq n$, the pattern I cannot be coded by a mixed template V, such that $|I \cap V| \leq n - 1$ (see Lemma 2). The above arguments prove T1 at $x = n$.

Proof of T2 at $x = n$**.** The result is obvious for a pattern I of size $< n - 1$ due to the validity of T2 for all x such that $2 \leq x \leq n - 1$. Let us now take a pattern I of size $|I| = n - 1$. Suppose, once more, that I was coded by $V_{I,n-1}^{+}$ in list presentation $n - 1$ and $|V_{I,n-1}^{+}| = l$. We distinguish two cases:

(a) $l = n - 1$. Due to the discussion in the proof of T1 for $x = n$, case (a), we conclude that the template $V_{I,n-1}^{+}$ is stable. Furthermore, after the creation of the template $V^+_{I,n-1} = I$, no other template equal to $V^+_{I,n-1}$ can be created (see Lemmas 1 and 2). As a result, in list presentations $\geq n$ the size-$(n-1)$ pattern, I, has direct access to its equal $V^+_{I,n-1}$ template (see Lemma 1). The stable $V^+_{I,n-1}$ template was created in a list presentation $\leq n-1$.

(b) $l < n-1$. Note that in this case, $V^+_{I,n-1}$ can code I. The template $V^+_{I,n-1}$ is stable, and new size-k templates ($l \leq k \leq n-2$) cannot be created in list presentations $\geq n-1$, due to the validity of T1 and T2 for all s such that $2 \leq s \leq n-1$. A template equal to I can be created prior to the end of list presentation $n-1$. After the end of list presentation $n-1$, knowing that I can be coded by the stable subset template $V^+_{I,n-1}$, a template equal to I can be created only if a pattern $\hat{I}$ is coded by a mixed template V such that $\hat{I} \cap V = I$; but this is impossible due to the validity of T1 at $x = n$ as proved above. If a template equal to I is created prior to the end of list presentation $n-1$, then this template is stable [see the discussion in the proof of T1 at $x = n$, case (a)]. No other template equal to I will be created thereafter (see Lemmas 1 and 2). Hence, in list presentations $\geq n$ either the stable template I or the stable template $V^+_{I,n-1}$ will code pattern I. In both cases, the stable template that codes I is created in list presentations $\leq n-1$. The proof of T2 at $x = n$ is now complete. Consequently, the theorem is true □.

Note: As mentioned before, the proof of T1 at $x = M$ is obvious because the assumptions of the theorem exclude patterns I of size greater than or equal to M. As a result, for the proof of T1 at $x = M$, it is not necessary to go through the arguments presented in the proof of T1 for $x < M$.

In the following, the conclusions of the theorem, as well as certain important byproducts of its proof, are presented as properties of learning in the ART1 network. Once more, it is assumed that an arbitrary list of binary input patterns is repeatedly presented to ART1. In list presentations $\geq x$, where $x \geq 2$, learning in ART1 has the properties:

P1: A pattern I of size $\geq x$ cannot be coded by a mixed template V, such that $|I \cap V| \leq x-1$.

P2: A pattern I of size $\leq x-1$ will have direct access to a stable template, that was created in list presentations $\leq x-1$.

P3: Size-$(x-1)$ templates cannot be created.

P4: Size-x templates cannot be destroyed.

The basic result of this work is that if an ART1 network is presented repeatedly with an arbitrary list of binary input patterns it self-stabilizes the recognition codes (templates) of size-l patterns in at most l list presentations.

This basic result is an immediate consequence of property P2. It is worth observing that properties P1–P4 are valid independent of the order in which the input patterns are presented within the list. In addition, the ordering of the patterns within the list can change from one list presentation to the next without affecting the validity of these properties. Finally, the basic result implies that if the input patterns can be represented by M input nodes, ART1 learns and recognizes the list after at most $M-1$ list presentations (size-0 and size-M patterns have been excluded from the input list).

4 Conclusion

An important self-organizing neural network, ART1, introduced and analyzed by Carpenter and Grossberg (1987) was considered. The convergence properties of any neural network model is an issue of fundamental importance. Carpenter and Grossberg have proven a multitude of ART1 properties, including certain of its convergence characteristics. In this work, we concentrated only on the convergence properties of ART1 in the fast learning case. In particular, under the modeling assumptions of Section 2, a size-l pattern from a list of binary input patterns presented repeatedly to ART1 has direct access to a stable code in at most l list presentations. Hence, each member of a list of binary input patterns presented repeatedly to ART1 will have direct access to a stable code after at most $M-1$ list presentations (size-0 and size-M patterns are excluded from the input list). Other useful properties associated with learning in the ART1 network were also shown.

Acknowledgments

This research was supported by a grant from the Florida High Technology and Industry Council with matching support from Martin Marietta Electronic Systems Division.

References

Carpenter, G. A., and Grossberg, S. 1987. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Comput. Vision, Graphics, Image Process.* **37**, 54–115.

Grossberg, S. 1976. Adaptive pattern recognition and universal recoding. II: Feedback, expectation, olfaction, and illusions. *Biol. Cybernet.* **23**, 187–202.

Received 18 April 90; accepted 6 August 90.