The Augmented ART1 Neural Network

Gregory L. Heileman
Department of Electrical and Computer Engineering,
University of New Mexico, Albuquerque, NM 87131 USA
Michael Georgiopoulos
Department of Electrical Engineering,
University of Central Florida, Orlando, FL 32816 USA

Abstract

A set of nonlinear differential equations that describe the dynamics of the ART1 model are presented, along with the motivation for their use. These equations are extensions of those developed in [2]. It is shown how these differential equations facilitate the real-time implementation of the ART1 model in its full generality. We specifically use the term real-time to refer to a neural network model whose description requires no *external* control features. That is, the dynamics of the model are *completely* determined by the set of differential equations that comprise the model. In this paper we interpret this definition of real-time in its strictest sense. This involves the removal of all algorithmic components from an implementation of the ART1 model.

1 Introduction

This paper discusses issues involved in the real-time implementation of ART1 architectures [2]. Specifically, we present a modified version of the ART1 neural network model that facilitates its real-time implementation. We denote this model as the *augmented* ART1 (AART1) neural network. A detailed analysis of the nonlinear differential equations that describe the AART1 model can be found in [3]. This analysis demonstrates how network parameters can be chosen in order to guarantee that the AART1 model behaves in the same manner as the ART1 model.

The term *real-time*, in this case, is used to refer to neural network models that require no external control (i.e., supervision) of system dynamics [1]. Thus, by definition, a real-time network cannot be placed in a *learn mode* during training and then later switched by some external control activity to a *performance mode* when training is complete. This, for example, is not the case with back-propagation learning in perceptron networks—separate externally controlled learning and performance modes do exist in this model.

Nonlinear differential equations are usually used to describe the dynamics of real-time neural network models. These equations must simultaneously describe both the performance dynamics of the network, as well as the network's learning dynamics. The description of neural network models in this manner is inherently nonalgorithmic. Typically, however, implementations of the ART1 model exhibit algorithmic components. This ranges, on the one extreme, to a popular implementation that is completely algorithmic in nature [4]. It is the *behavior* of the set of differential equations describing the ART1 network that is implemented by the algorithm presented in [4]. Although this algorithm has proven quite useful, it fails to capture the full generality of the ART1 model. A more subtle introduction of algorithmic components occurs in ART1 simulations in which the differential equations describing the system dynamics are numerically implemented; however, the resetting of network nodes, and the reset mechanism itself are handled through external control. This approach typically involves iterating the network equations through a number of time steps, and then stopping to test if the reset mechanism needs to be employed.

In this paper, we introduce modifications to the differential equations describing ART1 that allow all of these so-called algorithmic components to be removed from ART1 implementations. It should be emphasized that these modifications allow the ART1 model to be implemented *solely* as a set of concurrently executing nonlinear differential equations. At the same time, the locality constraints that must be observed by a neural network model are preserved. The significance of these modifications, from a theoretical point of view, is that they demonstrate the inherent capability of the ART1 network to operate in a totally unsupervised manner. From a practical point of view, these modifications

0-7803-0164-1/91/0000-0467$01.00©1991 IEEE

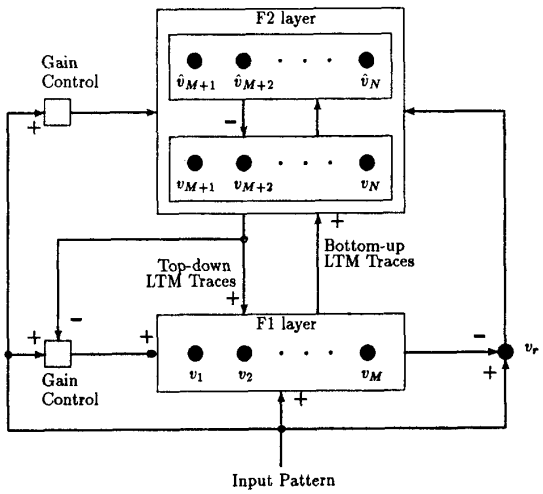

Figure 1: The architecture of the augmented ART1 neural network model.

may facilitate the implementation of ART1 networks. For example, if an engineer wishes to design a parallel version of the ART1 network using analog circuitry, then the designer must only consider how to implement each of the differential equations described here in parallel—not some algorithmic description of the model. Therefore, the designer does not have to create special circuitry for keeping track of which nodes have been reset during a particular pattern presentation, or circuitry for performing the actual resetting of network nodes—this functionality is embodied in the differential equations themselves.

2 The Augmented ART1 Neural Network

A number of implementation issues that are not directly addressed in [2] are considered here. These include: **i)** the manner in which the mismatch-mediated reset wave can be generated; **ii)** the approach taken to ensure that an F2 layer node remains inactive, once it is reset, until a new input pattern is presented; and **iii)** A way of automatically driving the activity of every node in the network to its resting value of zero whenever an input pattern is removed from the network. The resolution of these issues will involve the addition of nodes in the ART1 neural network architecture, and minor modifications to the original ART1 neural network equations presented in [2]. The resulting model is termed the AART1 neural network. The major components of the AART1 neural network are shown in Figure 1. Section 3 demonstrates that the AART1 network is capable of providing a real-time implementation of the ART1 model.

Let $|I|$ denote the number of input pathways which receive positive inputs when the input pattern I is presented. Also, let $|X|$ denote the number of nodes in the F1 layer that are supraliminally active. In the ART1 model, each of the $|I|$ input pathways sends an excitatory signal of fixed size P to the orienting subsystem, and each of the $|X|$ supraliminally active nodes in the F1 layer generates an inhibitory signal of fixed size Q that also impinges on the orienting subsystem. Furthermore, the orienting subsystem in the ART1 model generates a nonspecific reset wave whenever

$$\frac{|X|}{|I|} < \rho = \frac{P}{Q}, \tag{1}$$

where ρ is called the *vigilance parameter.*

The generation of the reset wave by the orienting subsystem can be accomplished within the framework of a real-time implementation through the introduction of a *reset node* v_r whose activity,

x_r, satisfies the following differential equation:

$$\epsilon_r \frac{d}{dt} x_r = -A_r x_r + U[P \sum_{i=1}^{M} I_i - Q \sum_{i=1}^{M} f_1(x_i)], \tag{2}$$

where U is the unit step function. Note that the activity of the reset node becomes positive whenever $\frac{|X|}{|I|} < \rho$ and decays exponentially to zero whenever $\frac{|X|}{|I|} \geq \rho$. The output of the reset node, $f_r(x_r)$, which corresponds to the non-specific rest wave, satisfies

$$f_r(x_r) = \begin{cases} 1, & \text{if } x_r > \delta_r; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The introduction of this reset node provides a mechanism for the generation of the reset wave required by the ART1 model whenever there is a sufficient mismatch between the input pattern I and the activity pattern X across the F1 layer. This resolves issue **i.**

One of the properties of the ART1 model is that the reset wave selectively and enduringly inhibits active F2 layer nodes until the input pattern is removed. This can be accomplished within the framework of a real-time ART1 implementation by augmenting the F2 layer with a set of inhibitory nodes. That is, every node v_j in the F2 layer is assigned an inhibitory node $\hat{v}_j$ whose activity, $\hat{x}_j$, satisfies the following differential equation:

$$\epsilon_2 \frac{d}{dt} \hat{x}_j = -[1 - g(I)]\hat{x}_j + g(I) f_r(x_r) f_2(x_j), \tag{4}$$

where

$$g(I) = \begin{cases} 1, & \text{if } \sum_{i=1}^{M} I_i \neq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The activity of an F2 layer inhibitory node can only become positive when the following actions are satisfied simultaneously: an input pattern is being presented to the network, a reset wave is being emitted by the reset node, and the corresponding node in the F2 layer is supraliminally active. Once the activity of an F2 layer inhibitory node has become positive, its activity decays exponentially to zero only when the input pattern is removed. In conjunction with a modification to the differential equation characterizing the activity of F2 layer nodes, this mechanism will allow the implementation of the selective and enduring inhibition of F2 layer nodes after a reset event, and as long as the input pattern is present. Specifically, the total inhibitory input to the node v_j in the F2 layer of the ART1 model is modified as follows:

$$J_j^- = \sum_{k \neq j} f_2(x_k) + \hat{f}_2(\hat{x}_j), \tag{6}$$

where $\hat{f}_2(\hat{x}_j)$ is the output of the F2 layer inhibitory node $\hat{v}_j$. The output of an F2 layer inhibitory node obeys

$$\hat{f}_2(\hat{x}_j) = \begin{cases} 1, & \text{if } \hat{x}_j > \hat{\delta}_2; \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

This resolves issue **ii.**

A modification to the equation describing the total excitatory input to an F2 layer node must also be made to allow the ART1 implementation to operate in a completely real-time manner. This will allow the F1 and F2 layer nodes to be reset to zero whenever an input pattern is removed from the network. This can be accomplished in the following manner. When an input pattern is removed from the network, it must be followed by the presentation of the null pattern (a pattern of size zero). This will rapidly drive the activity of nodes in the F1 and F2 layer to zero if we modify the J_j^+ quantity in the ART1 model as follows:

$$J_j^+ = f_2(x_j) g(I) + D_2 \sum_i f_1(x_i) z_{ij}. \tag{8}$$

Conceptually, this presents no problems as it represents an absence of stimuli at the network inputs. That is, instead of a constant bombardment of stimuli, the learning system is allowed a brief "rest period" between each stimulus presentation. This resolves issue **iii.**

3 Computer Simulation

In this section we demonstrate the capabilities of the real-time AART1 model for both the fast and slow learning cases. It is assumed that in a fast learning environment, the input pattern is presented long enough for the bottom-up and top-down LTM traces to reach their limiting values; while a slow learning environment corresponds to the case where the input pattern is presented long enough for the network to choose the correct node to code the input pattern, but not necessarily long enough for the bottom-up and top-down LTM traces to reach their limiting values.

The sample network considered here contains 4 nodes in the F1 layer (v_1–v_4), a reset node (v_r), and 8 nodes in the F2 layer (v_5–v_8, and $\hat{v}_5$–$\hat{v}_8$). The node differential equations were numerically approximated using the fourth order Runge Kutta method with a step size of 10^{-6}. Three patterns were presented to the network: $I^1 = 1000$, $I^2 = 0000$, and $I^3 = 1100$. Note that I^2 is the null pattern used between presentation of other "interesting" patterns.

The parameters chosen for the simulation of the sample network in the fast and slow learning cases are shown below. These parameters were chosen so as to satisfy the constraints presented in [3].

$A_1 = 1$	$B_1 = 0.5$	$C_1 = 100$	$D_1 = 1$	$\epsilon_1 = 0.001$	$\delta_1 = 0.01$	
$A_r = 2$	$\epsilon_r = 0.001$	$\delta_r = 0.02$	$\rho = 1$			
$A_2 = 0.3$	$B_2 = 10000$	$C_2 = 10000$	$D_2 = 1.25$	$\epsilon_2 = 0.01$	$\delta_2 = 0.01$	$\hat{\delta}_2 = 0.0001$
$K = 1$	$L = 1.01$	$\epsilon_z = 1$				

The LTM traces for these simulations were selected so that initially $z_{ji} = 1$, and $0 < z_{ij} < \frac{L}{L-1+M}$ for all i, j. In addition, the bottom-up LTM traces were chosen so that when I^1 is initially presented, v_5 receives the largest bottom-up input. Furthermore, when I^3 is initially presented, v_5 receives the largest bottom-up input, and v_6 receives the next largest bottom-up input.

The fast learning case is examined first. Pattern I^1 is presented to the network at time $t = 0$ (see Figure 2A). After I^1 is presented, the activity of v_1 increases from zero to a positive value above δ_1—point a in the figure. (Note that points a and b in each of the figures below correspond to the thresholds δ_1 and δ_r, respectively.) Node v_5 becomes supraliminally active before any other node in the F2 layer. At this point, v_1 is receiving both bottom-up input, and strong top-down input from v_5. This causes x_1 to decrease and subsequently reach a limiting value that is above δ_1. Once v_5 becomes supraliminally active, it will inhibit the other F2 layer nodes, forcing them to remain subliminally active as long as it remains supraliminally active. The activity of v_r in Figure 2A should also be noted. Immediately after the presentation of I^1, x_r increases due to the mismatch between the output activity across the F1 layer, which equals zero, and I^1. Notice that the output activity across the F1 layer becomes equal to I^1 before x_r exceeds δ_r. After the activation of v_5, x_r continually decreases due to the fact that the mismatch at the F1 layer no longer exists. Pattern I^1 is presented until time $t = 3.0$. This allows the bottom-up and top-down LTM traces to approximately reach their limiting values. Hence, v_5 codes I^1.

At time $t = 3.0$, I^2 is presented to the network (see Figure 2B). Initially x_1 is above δ_1, but it drops to a level below δ_1 almost instantaneously. This results from v_1 receiving only top-down input. After the deactivation of v_1, x_1 and x_2 stay at a constant level until v_5 is deactivated. Once v_5 becomes subliminally active, x_1 and x_2 decrease to zero because they are no longer receiving top-down input. The activity of v_6 starts increasing from a negative value towards zero immediately after v_5 becomes subliminally active. Pattern I^2 is held at the network input until time $t = 3.2$.

At time $t = 3.2$, I^3 is presented (see Figure 3A). After the presentation of I^3, v_5 becomes supraliminally active before any other node in the F2 layer because it receives the largest bottom-up input. Once v_5 becomes supraliminally active, x_1 and x_2 begin to decrease. Notice that x_1 remains above δ_1, while x_2 decreases to a level below δ_1. This is a consequence of v_1 receiving strong top-down input, while v_2 receives weak top-down input. When v_2 becomes subliminally active, x_r starts increasing due to the mismatch that is now occurring at the F1 layer. When v_r becomes supraliminally active it

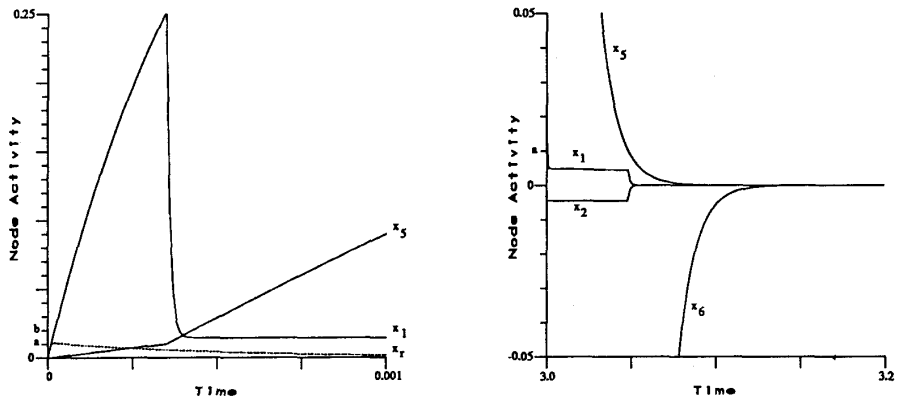

Figure 2: (A) Node activities during the presentation of pattern I^1. (B) Node activities during the presentation of pattern I^2.

generates a reset wave that deactivates v_5. After v_5 becomes subliminally active, v_1 and v_2 receive only bottom-up input, and their activities increase (see Figure 3B). Node v_6 will become supraliminally active next since it is the node in the F2 layer that receives the next largest bottom-up input. When v_6 becomes supraliminally active, x_1 and x_2 begin to decrease; but they remain above δ_1. This is a consequence of both v_1 and v_2 receiving bottom-up input and strong top-down input. Notice also that x_r continues to decrease after the deactivation of v_5. Hence, v_6 codes I^3.

We now consider the slow learning case. First, I^1 is presented at time $t = 0$, and the network exhibits the behavior depicted in Figure 2A. However, in this case, soon after v_5 wins the competition in the F2 layer, I^1 is removed from the network inputs. Thus, the bottom-up and top-down LTM traces are not allowed to converge to their limiting values. Pattern I^1 is presented until time $t = 0.1$, and then I^2 is presented. By time $t = 0.3$, all node activities have converged to their resting values of zero. The behavior of the network during the presentation of I^2 is similar to that shown in Figure 2B. The major difference between the fast and slow learning cases demonstrated in these simulations occurs when I^3 is presented to the network at time $t = 0.3$ (see Figure 4). Node v_5 receives the largest bottom-up input, and it is activated prior to any other F2 layer node. This activation forces x_1 and x_2 to decrease to limiting values that remain above δ_1. In the slow learning case, the fact that x_2 remains above δ_1 while I^3 is presented is a consequence of not allowing the top-down traces leading to v_5 to reach their limiting values during the presentation of I^1. As a result, when v_5 becomes supraliminally active, v_2, as well as v_1, receive bottom-up input and strong top-down input. Thus, since both v_1 and v_2 stay supraliminally active, v_5 is not reset. Therefore, v_5 codes I^3.

4 Conclusions

A real-time neural network, AART1, that describes the dynamics of the ART1 model was presented. This involved modifying the architecture of the ART1 network through the addition of a reset node and a set of inhibitory nodes in the F2 layer. Along with modifications to the differential equations describing the F1 and F2 layer nodes, theses changes embody the functionality of the orienting subsystem in the ART1 model. In addition, the modifications described here allow the ART1 model to be implemented solely using a set of concurrently executing nonlinear differential equations. Thus, the AART1 network requires no external control features. Computer simulation results demonstrate the efficacy of this network. In [3] we provide an analysis of the AART1 network that proves it is capable of behaving in the same manner as the ART1 model. The modified equations presented in Section 2 have been successfully simulated for a variety of networks using a wide range of parameter values.

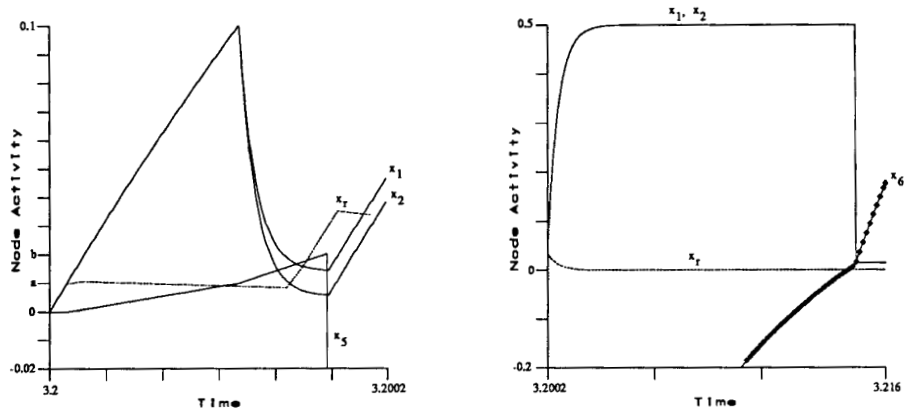

Figure 3: (A) Node activities leading to a reset during the presentation of pattern I^3. (B) Node activities after the reset during the presentation of pattern I^3.

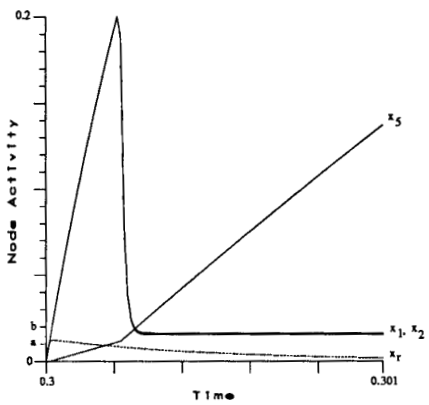

Figure 4: Node activities after the presentation of pattern I^3, when pattern I^1 has not been coded by v_5 on a previous pattern presentation.

References

[1] G. A. Carpenter. Neural network models for pattern recognition and associative memory. *Neural Networks*, 2(4):243–257, 1989.

[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. L. Heileman and M. Georgiopoulos. A real-time representation of the ART1 network. Technical Report EECE 91–001, University of New Mexico, January 1991.

[4] R. P. Lippmann. An introduction to computing with neural nets. *IEEE Acoustics Speech and Signal Processing Magazine*, 4(2):4–22, 1987.