Chapter 16

Basic ART Resonators

§ 1. Natural and Artificial ART

For the most part, when one reads or hears of an ART network, the network being discussed is an artificial neural network machine. Indeed, the first two widely disseminated papers in which the term "ART network" was used were published in *Computer, Vision, Graphics, and Image Processing* [CARP2] and in *Applied Optics* [CARP3]. As one might expect from these titles, the systems there described are machines intended for specific engineering applications, and I believe it is not unfair to say their connection to biological signal processing is latent and lies hidden in the shade of some of the references cited in these works. To use the time-worn phrase favored by artificial neural network theorists, ART1 and ART2 are "inspired by biology." This is to say that unless one believes biological neural networks exist that do such things as calculate $L_2$ norms (more commonly known as Euclidean distance), the artificial networks should not be held to be representations of natural neural networks, but at best merely approximations of neural function.

This is in no way meant to denigrate the artificial ART networks of the engineering and mathematical worlds. Quite the contrary. They are powerful and useful machines for carrying out various engineering signal processing tasks. Nor does this imply that these artificial networks carry no connection with biological signal processing. The artificial ART networks of the 1980s and 1990s owe their foundation to the 1970s work carried out by Grossberg in modeling neurological and psychological phenomena. But it is to say that they are fundamentally *algorithms* and the tasks they carry out are mathematico-engineering tasks. Many practitioners of artificial neural network theory are fond of implying that this field of engineering and mathematics is a manifestation of the same sort of underlying principles that govern the function of the brain, and perhaps to some degree this is true. But at the present time to make such an implication on behalf of *any* artificial neural network structure is to engage in romantic speculation rather than natural science.

Still, as was just noted, the *design strategy* for ART1, ART2, ART3, ARTMAP, etc. does draw from work that was aimed at understanding biological systems, and from mathematical consequences that appear to follow from empirical findings in psychology and neurology. Our pedagogical objective in this chapter is to introduce and understand what some of these consequences are. We shall concern ourselves with adaptive resonance theory and its structures.
insofar as they have relevance for biological signal processing on the very large scale and for computational neuroscience in general. In the process of coming to this understanding, the reader will also acquire the background needed to appreciate the types of approximations and simplifications engineers make in constructing artificial ART systems, and perhaps even to develop an eye for spotting the points where artificial ART systems employ mathematical structures for which there is no presently known biological foundation.

What is the objective of an artificial ART network design? Carpenter and Grossberg describe this quite succinctly:

Adaptive resonance architectures are neural networks that self-organize stable pattern recognition codes in real-time in response to arbitrary sequences of input patterns. This article introduces ART 2, a class of adaptive resonance architectures which rapidly self-organize pattern recognition categories in response to arbitrary sequences of either analog or binary input patterns. In order to cope with arbitrary sequences of analog input patterns, ART 2 architectures embody solutions to a number of design principles, such as the stability-plasticity tradeoff, the search-direct access tradeoff, and the match-reset tradeoff. In these architectures, top-down learned expectation and matching mechanisms are critical in self-stabilizing the code learning process. A parallel search scheme updates itself adaptively as the learning process unfolds, and realizes a form of real-time hypothesis discovery, testing, learning, and recognition. After learning self-stabilizes, the search process is automatically disengaged. Thereafter input patterns directly access their recognition codes without any search. Thus recognition time for familiar inputs does not increase with the complexity of the learned code. A novel input pattern can directly access a category if it shares invariant properties with the set of familiar exemplars of that category. A parameter called the attentional vigilance parameter determines how fine the categories will be. If vigilance increases (decreases) due to environmental feedback, then the system automatically searches for and learns finer (coarser) recognition categories. Gain control parameters enable the architecture to suppress noise up to a prescribed level. The architecture's global design enables it to learn effectively despite the high degree of nonlinearity of such mechanisms [CARP3].

Most of the properties and abilities cited here for ART2 are also directly relevant for natural neural network systems. We will see the requirement for these functional abilities arise as a quite natural part of the ART dynamics presented in this chapter. We will not, however, make any attempt to present a general "class of adaptive resonance architectures." To present a "class" is to present in the abstract, and it is a psychological fact that people do not learn from the abstract to the particular but, rather, from the reverse. It is one thing to present in the abstract in an archival journal – as most papers on ART do – and something else to present the ideas of a theory in a textbook. Therefore, here we will develop adaptive resonance theory through examples, from which we will be able to discern the origin of and the need for solutions to such things as "the stability-plasticity tradeoff, the search-direct access tradeoff, and the match-reset tradeoff" of which Carpenter and Grossberg speak in the quote just given. We will see the need for such solutions arising as consequences of specific signal processing issues and limitations of particular subsystems within the ART framework.
Chapter 16: Basic ART Resonators

§ 2. Resonator 1

The heart of an ART network is a subsystem called the adaptive resonator. There are a number of versions of adaptive resonators. Figure 16.1 illustrates the prototype upon which most others are based, either as specializations of it or enhancements of it. We will call this anatomy resonator 1 or R\(^{(1)}\) for short. We have already encountered all the pieces that go into its makeup in the previous chapters, especially chapter 15. It consists of three layers. Layer 0 is a Grossberg normalizer, which we will here assume to be GN\(^{(2)}\). This layer is actually optional, and if the resonator is driven by the Instar outputs of another resonator GN can be omitted. The GN layer is in some ways a redundancy since the signal normalization function it performs can be handled by layer \(v_1\). Layer 1 (\(v_1\)) is a competitive contrast enhancer, specifically the CE used in classifier CL\(^{(2)}\) in chapter 15. We will hereafter refer to this as CE\(^{(2)}\). Both GN and \(v_1\) have \(n\) nodes. Layer 2 (\(v_2\)) is also a competitive contrast enhancer with SNI nodes comprising a second CE\(^{(2)}\) network. However, in addition each SNI\(^{(3)}\) node in \(v_2\) also drives an Outstar node, which is the source of the feedback from \(v_2\) to \(v_1\). Layer \(v_2\) contains \(N\) nodes.

Each SNI in \(v_1\) receives a forward-path input signal \(I_i\) from its corresponding node in the GN and a set of \(N\) feedback signals \(K_{ij}\) from \(v_2\). These are summed to produce its total input signal \(\xi_i\). Each SNI in \(v_2\) receives a set of \(n\) feedforward signals \(J_{ij}\) from each SNI in \(v_1\), which are summed...
to produce its total input signal $\xi_j$. In vector-matrix form we have

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_N \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} + \gamma \cdot \begin{bmatrix} z_{11} & z_{21} & \cdots & z_{N1} \\ z_{12} & z_{22} & \cdots & z_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ z_{1n} & z_{2n} & \cdots & z_{Nn} \end{bmatrix} \cdot \begin{bmatrix} h_1(x_{21}) \\ h_2(x_{22}) \\ \vdots \\ h_2(x_{2N}) \end{bmatrix} \quad \text{or} \quad \xi_1 = I_1 + \gamma \cdot Z \cdot h_2(x_2). \quad (16.1)$$

Here $\gamma$ is a constant we will call the feedback gain and $h_2(u)$ is an activation function for the $v_2$ SNI excitation variables $x_{2j}$. $Z$ is the matrix of Outstar weights. The second term on the right-hand side of (16.1) defines the $K_{\mu j}$ variables. As for the $J_{\mu j}$ inputs to $v_2$, these are given by

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_N \end{bmatrix} = \begin{bmatrix} w_{11} & w_{21} & \cdots & w_{n1} \\ w_{12} & w_{22} & \cdots & w_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1N} & w_{2N} & \cdots & w_{nN} \end{bmatrix} \cdot \begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \end{bmatrix} \quad \text{or} \quad \xi_2 = W \cdot x_1. \quad (16.2)$$

where $W$ is the matrix of feedforward weights. For convenience we will abbreviate the columns of the $Z$ matrix as $Z_i$ and the rows of the $W$ matrix as $W_j^T$.

Each Instar node in $v_1$ and $v_2$ is governed by the dynamical equation (14.31) for SNI\(^3\) and simulated using the difference equation form (14.33). As we did in chapter 15, we will impose the constraint that all $x_{1j}$ and $x_{2j}$ variables must remain non-negative (i.e., we will clip them at zero if the equation would take them less than zero). As noted in chapter 15, this is equivalent to having the $x_{1j}$ and $x_{2j}$ variables in our equations represent $h(x_{ab})^1$ where $h$ is the Heaviside extractor activation function. In order to keep our mathematical notation as simple as possible, we will let it be understood that this clipping action is always implied in every calculation.

We can compactly write the mathematical description for $R^{(1)}$ in vector-matrix notation as

$$\begin{align*}
x_1(t+\Delta t) &= x_1(t) + \Delta t \cdot \left( - (A_1 + F_1 + \xi_1) \otimes x_1 + B_1 \cdot (f_1 + \xi_1) - D_1 \cdot (F_1 - f_1) \right) \\
x_2(t+\Delta t) &= x_2(t) + \Delta t \cdot \left( - (A_2 + F_2 + \xi_2) \otimes x_2 + B_2 \cdot (f_2 + \xi_2) - D_2 \cdot (F_2 - f_2) \right)
\end{align*} \quad (16.3)$$

Here it is understood that everything on the right-hand side is evaluated at time $t$. $A_1$ and $A_2$ are constant column vectors ($n \times 1$ and $N \times 1$, respectively) with elements $A_1$ and $A_2$. $F_1$ and $F_2$ are column vectors with elements $F_k = \sum f_k(x_{km})$, $k = 1$ or 2, $m = i$ or $j$, with the sum taken on $m$ over all the excitation variables in that layer according to the definition of $F$ introduced in chapter 15. $f_1$ and $f_2$ are the activation functions for $v_1$ and $v_2$, respectively. We will assume $f_1$ and $f_2$ are the same general function but with layer-specific parameters $g_{max}$, $u^{(1)}$, $u^{(2)}$ as per figure 14.7B. Without loss of generality, we will use (14.23) for the activation function in all examples.

\(^1 a = 1 \text{ or } 2, \ b = i \text{ or } j.\)
Chapter 16: Basic ART Resonators

\( f_1 \) and \( f_2 \) denote column vectors with entries \( f_1(x_{1i}) \) and \( f_2(x_{2j}) \), respectively. The symbol \( \otimes \) denotes term-by-term multiplication with vectors \( x_1 \) or \( x_2 \). The remaining scalar terms are as previously defined in chapter 15. The iteration step size \( \Delta t \) is constrained by (d14.19) for whichever layer gives the smallest bound. Equations (16.1) through (16.3) define the dynamical system for \( R^{(1)} \) except, of course, for its adaptation dynamics.

The first thing we must note about (16.3) is that these two matrix difference equations are coupled to one another through (16.1) and (16.2). The overall system of equations is therefore different than the system of equations for CL\(^{(1)}\) in chapter 15. The coupling is what makes \( R^{(1)} \) a resonator. There has been no widely disseminated paper comparable to [GROS14] published, which means we do not possess a set of theorems, analogous to those in chapter 15, that set down the mathematical properties of \( R^{(1)} \). It is true that we can expect this system to behave in many ways like the one we studied in chapter 15. For example, the layers of \( R^{(1)} \) each have a quenching threshold, QT\(_1\) and QT\(_2\). However, it should come as no surprise that this system also has different behaviors not in evidence in CE\(^{(1)}\). We will encounter some of these later on in the examples presented here. The reason most of the theorems presented in chapter 15 cannot be assumed for \( R^{(1)} \) is that the equation for \( R^{(1)} \) is not the same as the equation for CE\(^{(1)}\). Therefore we are not in possession of proofs of these theorems for the case of \( R^{(1)} \). For this we would need a "new GROS14" – a formidable undertaking. Nonetheless, in a great many cases we will be able to use our knowledge of the Grossberg theorems to understand a number of important behaviors this system exhibits.

The general theme describing \( R^{(1)} \) behavior is this. A vector of input signals is applied to \( v_1 \) and causes excitation of the \( x_{1i} \) variables. This is called an activity pattern – essentially just \( x_1 \) – and constitutes the "short term memory" STM\(_1\) of \( v_1 \). STM signals are projected to \( v_2 \) via \( W \) and set up an activity pattern \( x_2 \). Feedback from the \( v_2 \) activity pattern to \( v_1 \) alters \( x_1 \), which causes an alteration in \( x_2 \), etc. A resonance is said to occur when both \( x_1 \) and \( x_2 \) settle into steady state fixed point solutions in response to the original applied inputs \{ \( I_i \) \}. When this input changes the system undergoes another transient leading to another fixed point resonance. Our main task in this chapter is to gain an understanding of this resonance process.

The desired state of affairs for the resonances of \( R^{(1)} \) has two main features. First, it is desired that STM\(_1\) be a representation – typically with some amount of contrast enhancement – of the input vector to \( v_1 \). The STM\(_1\) vector, \( x_1 = \Theta \), is the basis for the adaptation of \( W \) and \( Z \). Second, the activity pattern STM\(_2\) = \( x_2 \) is desired to represent an encoding of the original input signal vector. The row vectors \( W_j^T \) constitute the classifying vectors for \( R^{(1)} \). The column vectors \( Z_i \) constitute a set of expectation vectors. These vectors are said to "learn the patterns" \( \Theta \). To the
extent that the $\Theta$ vectors represent the inputs $\{I_i\}$, $v_2$ is said to have categorized the inputs.

This explains what occurs. How does this work in $R^{(1)}$? The qualitative explanation is as follows. Let $I_1$ and $I_2$ be two external input patterns to $v_1$. Let us further suppose that both signals have previously been successfully coded by $v_2$ such that $I_1$ is encoded by a 0-1 distribution in $v_2$ in which node $x_{21}$ is the only active node. Similarly let $I_2$ be encoded by a 0-1 distribution in which $x_{22}$ is the active node. When $I_1$ is presented to $v_1$, $v_1$ responds by generating an activity pattern $STM_1$ that excites $v_2$. By our assumptions, this will produce an excitation only in node $x_{21}$. This, in turn, generates a feedback vector $K_1 = \gamma \cdot Z_1 \cdot h_2(x_{21})$. In the ideal case $K_1$ will equal $I_1$ except for a multiplicative factor determined by $\gamma$. $v_1$ now has a new input that is exactly the same as $I_1$ except for a multiplicative factor. Thus, $v_2$ continues to receive an input signal that is the same, except for the multiplicative factor, that it received when $I_1$ was first applied. This maintains the $STM_2$ pattern, the feedback to $v_1$, and thereby the system settles into a steady-state fixed point of operation. (This steady-state condition is what allowed $W_1$ and $Z_1$ to adapt to, ideally, a contrast-enhanced prototype of $I_1$).

Now suppose the input changes from $I_1$ to $I_2$. The total input to $v_1$ is now $I_2 + K_1$, which, again ideally, will equal $I_2 + cI_1$, where $c$ is the aforementioned scaling factor. This changes the activity pattern of $STM_1$ and the two inputs will compete, with $I_2$ attempting to establish its unique activity pattern and $K_1$ attempting to maintain the pattern associated with $I_1$. If $I_1$ and $I_2$ are sufficiently different and if $K_1$ is a sufficiently weaker signal than $I_2$, the $STM_1$ activity pattern will change. In the extreme case, if $I_2 + K_1$ produces a uniform net input signal vector, then $STM_1$ will be abolished altogether. This is because the $D_1$ term in the SNI$^{(3)}$ equation produces an effect similar to the $C$ term in GN$^{(2)}$. We recall that GN$^{(2)}$ suppresses uniform input patterns and produces a zero-vector output. The $v_1$ layer does the same. Thus, if $I_2 + K_1$ is uniform, $STM_1$ vanishes and, lacking an input excitation, $v_2$ also resets to the zero condition. But this removes the feedback signal $K_1$ and so now $v_1$ "sees" only $I_2$ and responds by producing the $STM_1$ pattern associated with $I_2$. This reactivates $v_2$, this time producing the $x_{22}$ 0-1 distribution. $v_2$ then feeds back $K_2$ – which again is ideally just a scaled version of $I_2$ – and the system "locks into" its new steady-state.

What happens if $I_1$ and $I_2$ are sufficiently different – which means they contain different features $R^{(1)}$ is using for its classification coding – but $I_2 + K_1$ does not produce a uniform input pattern? In this case, and again assuming $K_1$ is a weaker signal than $I_2$, the change in $STM_1$ will cause $x_{21}$ to decrease (it is no longer seeing as strong an input signal) and cause $x_{22}$ to increase (it is now "seeing" a stronger input because $STM_1$ contains the influence of $I_2$). $v_2$ will begin to reverberate and if its quenching threshold is sufficiently high its activity pattern will be erased.
Chapter 16: Basic ART Resonators

and replaced by the activity pattern for \( I_2 \), much as we saw for \( \text{CL}(2) \) in chapter 15. Reverberation in \( v_2 \) at first weakens and then quenches \( K_1 \) and replaces it with \( K_2 \), the feedback vector for \( x_{21} \).

This is the basic idea behind the operation of \( R^{(1)} \). To achieve it: (1) the feedback signals must be weaker than the external input signals from \( \text{GN}(2) \); (2) \( v_1 \) must have a sufficiently high QT that new patterns erase and overwrite old STM\(_1\) patterns; and (3) \( v_2 \) must likewise have a sufficiently high QT so that its STM does not persist when the input that caused it is removed. Whether or not \( I_1 \) and \( I_2 \) are "sufficiently different" depends on these vectors containing different distinguishing features that have been encoded into \( W_1 \) and \( W_2 \) (and into \( Z_1 \) and \( Z_2 \) which, ideally, are equal to \( W_1 \) and \( W_2 \), respectively, except perhaps for a constant scale factor). These are the Grossberg conditions for successful encoding and classification in \( R^{(1)} \).

It will not have escaped your attention that the word "ideally" appears several times in the description just given. In practice, things are rarely ideal and this is the case for \( R^{(1)} \). We have just ignored the details of the temporal dynamics that lead to all these changes in STM, and we have further invoked the still-vague notion \( I_1 \) and \( I_2 \) are "sufficiently different." We must meet and confront the nasty realities that lurk within the practical system and see how these nasty realities affect the ideal case we have just described. We will do this in the next section through the vehicle of a specific example system.

§ 3. Resonator 1 Dynamics with Binary-valued Input Patterns

It will suffice without loss of generality to study the dynamics of \( R^{(1)} \) for a specific case. In this section we will study an \( R^{(1)} \) network with \( n = 25 \) nodes in \( \text{GN}(2) \) and \( v_1 \), and \( N = 3 \) nodes in \( v_2 \). The parameters of the system will initially be set at:

- \( \text{GN}(2) \): \( B_0 = 1; A_0 = 0.5; C_0 = B_0/(n-1) \)
- \( v_1 \): \( B_1 = 1; A_1 = 0.5; D_1 = 1.5; g_{\text{max}1} = 1; u_1^{(1)} = 0.85; u_1^{(2)} = 0.98 \)
- \( v_2 \): \( B_2 = 1; A_2 = 0.5; D_2 = 1.0; g_{\text{max}2} = 1; u_2^{(1)} = 0.95; u_2^{(2)} = 0.98 \)

The feedback gain \( \gamma \) will be set to 0.05. These parameters were chosen empirically to yield an \( R^{(1)} \) network with good performance and reasonable robustness in the simulation cases studied. For the sake of simplicity, we will consider the LTM weights in \( W \) and \( Z \) to be fixed. Thus the simulations are considered to take place after the system's adaptation process has established the classifying vectors. Adaptation will be considered in chapter 17.

Although all input signals are column vectors, as an aid to visualization we will present them as if they were arranged in a \( 5 \times 5 \) square grid (a "retina") as we did in chapter 15. Figure 16.2 illustrates four representative input patterns, which we shall call T, J, X, and O, respectively. The weight vectors \( W_1 \) and \( Z_1 \) equal the T pattern except for scaling so that \( |W_1| = |Z_1| = 1 \). Similarly, \( W_2 \) and \( Z_2 \) are set to the J pattern, and \( W_3 \) and \( Z_3 \) are set to the X pattern.
Where two retina patterns share common "pixels" the patterns are said to have *common features*. If we compare the T and J patterns, we see that these two patterns do not intersect at any pixel. T and J are said to be "well separated" or "completely distinct" from one another. The X and O patterns, on the other hand, each have non-empty intersects with the other three patterns, i.e. they both share common features with the other three patterns.

Finally, we must define the $h_2$ activation function in (16.1). For this we will begin with a rather poor choice, the popular Heaviside step function, i.e. $h_2(x) = 1$ if $x > 0$ and $h_2(x) = 0$ otherwise. We will soon see the compelling reason for changing this choice to the Heaviside extractor. This completes our parameter definition of the example $R^{(1)}$ system for now.

We will find it useful to have a way to measure and assess how closely the activity pattern $x_1$ matches the input pattern $I$ presented to $v_1$ by $G^{(2)}$. In mathematics such a measuring function is called a *metric function* and the measure itself is called a *distance*. Mathematicians have defined
many different kinds of distance metrics, the most familiar of which is the Euclidean distance metric. Our purposes here – namely the study of biological signal processing – are best served if our distance metric function is one that a neural network system can produce. In general, a function $\rho$ is a metric function if it has the following three properties:

- For any pair $x$ and $y$, $\rho(x, y) = 0$ implies $x = y$;
- $\rho(x, y) \geq 0$ for any pair $x$ and $y$;
- For any $x, y,$ and $z$, $\rho(x, y) + \rho(y, z) \geq \rho(x, z)$.

If $I$ and $x$ are vectors with $n$ elements, the absolute value metric

$$\rho(I, x) = |I_1 - x_1| + |I_2 - x_2| + \ldots + |I_n - x_n| = |I - x|$$

is a valid metric function. It is easily verified that the simple Instar network of figure 16.3 generates this metric function, and so we will use the absolute value metric as our biologically plausible measure of the "distance" between $I$ and $x_1$.

Under the simplest form of adaptation – the IAR managed locally by $R^{(1)}$ – success depends on $v_2$ developing a 0-1 distribution at its outputs. Although we are not yet considering adaptation dynamics, the parameters of the system have been chosen to produce this type of coding. With $N$ nodes in $v_2$, $R^{(1)}$ can classify at most $N$ distinct categories. We will begin our examination by seeing how $R^{(1)}$ performs this task given weight settings that are exact copies of the T, J, and X patterns, and how it responds to the unclassified pattern O. As we will be applying patterns in a sequence of $T \rightarrow J \rightarrow X \rightarrow O$, our first simulation series will be called the TJXO series.

**TJXO1.** Let the TJXO pattern sequence be corrupted by uniformly distributed random noise over a range $(0, 0.02)$ relative to unit pixel amplitudes in the pattern. The noisy signal passes through $GN^{(2)}$ and the normalized input is applied to $v_1$. Figure 16.4 shows the excitation, $v_2$, and
Chapter 16: Basic ART Resonators

Figure 16.4: Response of $R^{(1)}$ to the TJXO sequence with upper noise bound of 0.02. The patterns were applied at the iteration steps indicated by the T, J, X, and O arrows in the mismatch metric trace. The total input pattern, including noise, was held constant through the duration of its application period.

The first thing we note is T and J were correctly classified quickly, and the low mismatch value attained indicates the pattern in $v_1$ was well matched to the input patterns $I$ in both cases. Some degree of mismatch is to be expected because $v_1$ will quench the low-valued noise pixels and transfer their excitations into the pattern pixels. We may also note that the transition from T to J input is quite obvious from the spike in the mismatch at time index 500. We may also note from the top traces of the total excitations $x(t)$ that $v_1$ rather quickly settled into a stable distribution, but that $v_2$’s response is considerably slower and had not leveled off by the time the transitions from T to J and from J to X took place.

Next we note that although X was eventually classified correctly with a low amount of $x_1$ mismatch, it took much longer for $R^{(1)}$ to reach its decision. $x_{22}$ ("J") and $x_{23}$ ("X") competed for about 1000 iteration steps before $x_{23}$ finally won out. It took over three times as long to settle this competition as it did for the system to switch from T to J. Not until time step 2000 did $x_{23}$ take over definitive classification of the X pattern. Recall that X shares two common features (pixels) with the predecessor pattern J. The classifier had difficulty in "deciding" if it was "seeing" pattern
Figure 16.5: Replication of the first TJXO simulation with different noise patterns. Only the specific noise patterns varied in this run. R(1) fails to correctly classify the X pattern.

J or pattern X. One might suspect that this "longer, harder" competition is a symptom of lack of robustness to extraneous noise, and as we will see in a moment, this suspicion is justified. Finally, we note that R(1) could not decide on pattern O with both x_{22} and x_{23} active. In this case, layer v_1 locked into a 0-1 final distribution with the "surviving" v_1 pixel being a common feature of both J and X.

Figure 16.5 shows the results of repeating the first simulation with new random noise values. The previous suspicion about lack of robustness is confirmed in this run. R(1) fails to classify the X pattern in this run. The cause of the failure was run-away quenching of the X pattern by v_1 over the course of the competition. What survived in the v_1 pattern was again a pixel common to both J and X. This is our first look at a performance property of R(1) that will assume some significant importance, namely the effect of contrast enhancing and quenching v_1 will perform on its input pattern. Figure 16.6 illustrates the v_1 STM at iteration indices 2000 (when category "J" begins to overtake category "X") and 2999 (just before O is applied). Figure 16.6A clearly shows the severe amount of contrast enhancement that has taken place at step 2000. The X pattern is still faintly visible, but the two dominant pixels are features in common with J. The near erasure of the
majority of the X features at time step 2000 (A) – which is what is meant here by the phrase "run-away quenching," – will become a total erasure by time step 2999 (B) except for the lone remaining dominant pixel.

Why did category J win out over category X in this simulation despite the X-pattern silhouette in 16.6(A) and the $x_{23} > x_{22}$ state at step 2000? There are two things to consider here. First, our use of the Heaviside step function for $h_2$ negates any amplitude advantage one $x_{2j}$ node might have over another. With the Heaviside step function activation, activation is all-or-nothing. So far as the feedback to $v_1$ is concerned, $K$ is as much determined by $x_{23}$ as it is by $x_{22}$. The second thing to consider is what was said earlier about uniform or uniform-like inputs to $v_1$ reducing its STM. With the Heaviside step function $h_2$, the total feedback-plus-input is "more uniform-like" than either X or J alone would be. We can observe from the top trace of figure 16.5 that $x(t)$ for $v_1$ is constantly on the decline all during the $x_{22}$-$x_{23}$ competition. Now recall also the theorems from chapter 15. As the lesser pixels decline in their $X_i$ normalized amplitudes, the largest pixel will increase in its $X_i$ amplitude. This appears to be the basic mechanism behind the excessive contrast enhancement dynamic in figures 16.5 and 16.6. The Heaviside step function is a poor choice for $h_2$ and leaves the system open to undue influence by very small amounts of pattern noise.

Replacing the Heaviside step function for $h_2$ with the Heaviside extractor, which you will recall preserves amplitude information for $x > 0$, makes a difference that may seem utterly out of proportion. Figure 5.7 illustrates a repeat of sequence TJXO under the same conditions except for the activation function $h_2$. Pattern X is easily and quickly classified by $R^{(1)}$. The decision dynamics are robust and $x_{22}$ (category "J") is driven to zero by the stronger "X" feedback signal. Figure 16.8 illustrates the STM patterns in $v_1$ at the end of the J-pattern application period and in the middle of the X-pattern application period. No excessive contrast enhancement occurs.
Chapter 16: Basic ART Resonators

Figure 16.7: Replication of sequence TJXO using the Heaviside extractor function for $h_2$, noise = (0, 0.02).

Figure 16.8: $v_1$ STM patterns just prior to pattern X application and at iteration step 2000.

As for pattern O, it is eventually classified as a "J" pattern. Interestingly, though, the residual STM in $v_1$ is a diminished O pattern roughly two-thirds the amplitude of 16.8B's pattern and about half as much as 16.8A. This is why the mismatch metric at time index 5000 in figure 16.7 is so small. O shares more common features with J than with either other classifying vector, and this is why $v_2$ classifies it as a J.
There is another nonlinear noise effect present in $R^{(1)}$. The previous examples have all been run at relatively modest noise levels, uniformly distributed in the range $(0, 0.02)$. At higher noise levels it is readily observed that pattern noise exerts an effect on the formation of STM in $v_1$. This effect is called noise-induced contrast enhancement or NICE. Despite the acronym, it is not a "nice" effect. We recall that IAR processing in its simplest form is fundamentally aimed at driving the classifying vectors to equal their prototype pattern models. When excessive amounts of contrast enhancement occur in $v_1$, this contrast enhancement will lead to a distortion of the classifying vectors.

The effect is exhibited modestly in figure 16.9 and more graphically in figure 16.10. The noise level for this simulation was set to the range $(0, 0.20)$, a factor of 10 increase in the range of the pattern noise. Indications of NICE are seen in figure 16.9 in the droop in $x(t)$ for $v_1$ in the first 1000 iteration steps, and even more clearly in the rise of the mismatch metric in this same interval. The effect is shown explicitly in the STM$_1$ plots of figures 16.10.

Because NICE is a nonlinear noise effect, there is a great deal of variability in the extent of the effect when the system operates in a NICE region. Figures 16.11 and 16.12 illustrate the effect at

---

**Figure 16.9:** Noise-induced contrast enhancement (NICE) in a TJXO sequence.
Figure 16.10: STM\textsubscript{1} showing extent of contrast enhancement for each pattern. (A) T-pattern at step 999. (B) J-pattern at step 1999. (C) X-pattern at step 2999. (D) O-pattern at step 4000.

Figure 16.11: NICE effect in a TXJO pattern sequence. Note the sharp rise in mismatch metrics for the T and J patterns as well as the corresponding droops in $x(t)$ for $v_1$. 
Chapter 16: Basic ART Resonators

Figure 16.12: NICE STM patterns corresponding to figure 16.11. (A) J-pattern. (B) X-pattern. (C) J-pattern. (D) O-pattern. These STMs are sample just prior to the next pattern application except for the O-pattern.

the same noise levels for a TXJO pattern. Note especially in figure 16.11 how NICE during the T pattern interferes with the subsequent processing of the X pattern, and how pronounced the NICE curve becomes for the J pattern. The corresponding STM plots for each pattern are shown in figures 16.12. These may be compared with their pattern-counterparts in figures 16.10.

For noise uniformly distributed from 0 to \( r \) the expected value of the noise in each pixel is \( r/2 \) and the mean-squared value per pixel is \( r^2/3 \). Thus, the average pattern noise power is \( nr^2/3 \). The signal power is the sum of the squares of the pixel amplitudes, which is 6 for the T and J patterns and 9 for the X pattern. For \( r = 0.20 \), this corresponds to a signal to noise ratio (SNR) arriving at the input of GN\(^{(2)}\) of 12.55 dB for T and J, and 14.3 dB for X. In contrast, for \( r = 0.02 \) the signal to noise ratios are 22.6 dB and 24.2 dB, respectively. Lower SNR implicates a greater spread in the \( X_i \) values being processed in \( v_1 \). We recall from Grossberg's theorems in chapter 15 that whether the response of a contrast enhancer will result in a fair distribution or a contrast enhancing distribution depends on the initial spread of \( X_i \) values and the quenching threshold factor \( QT_1 = u_1^{(1)}(B_1 - A_1/g_{max}) \). Raising \( QT_1 \) therefore will combat the onset of NICE.
Figures 16.13 and 16.14 illustrate the effect of raising $Q_T^1$ by increasing $A_i$ from our base value of 0.5 to a new value of 0.8. $\eta_1^{(1)}$ remains at 0.85. The effect is a dramatic decrease in NICE
for the noise distribution range (0, 0.20). Compare these results against figures 16.9 and 16.10.

To summarize where we are so far, we have seen that the choice of feedback activation function $h_2$ exerts an important effect on the dynamics of the system. We have compared the Heaviside step function against the Heaviside extractor and found the latter to be preferable in $R^{(1)}$. We have seen that the competition between $v_2$ nodes in establishing a resonance benefits from allowing the amplitude information in $v_2$ to act as a selection mechanism. We have examined the performance of $R^{(1)}$ as a classifier at high SNR levels and seen that it performs this task quite well. Finally, we have examined the nonlinear phenomenon of noise-induced contrast enhancement, found this to be a serious impairment at low SNR levels, and seen that the effect is combated by increasing $QT_1$. In our specific example, this was done by raising $A_1$ to 0.8$B_1$ from our starting value of 0.5$B_1$. We henceforth adopt these changes in our example system.

**Feedback Gain.** Up to this point, we have said nothing about the feedback gain factor $\gamma$. This parameter plays an important role in determining the dynamics of the system. For instance, in all our previous examples we have observed $v_2$ to produce 0-1 distributions for patterns corresponding to its classification vectors. However, the value for $D_2$ is $D_2 = B_2$, a level which in chapter 15 produced only a contrast-enhanced fair distribution and not a 0-1 distribution in $CL^{(2)}$ (see figure 14.22A). This difference between $R^{(1)}$ and $CL^{(2)}$ is caused by the feedback.

The feedback gain must be large enough so that, in conjunction with the magnitude of the weight settings $z_{ji}$, it supports persistence in $STM_1$ in the presence of an unchanging input pattern $I_1$ but, at the same time, promotes decay in $STM_1$ when a new input $I_2$ arrives that "belongs" to a different classifying vector $W_j$. If $\gamma$ is too small, it will fail to promote resonance in $R^{(1)}$. But if it is too large, it will prevent $R^{(1)}$ from properly responding to new inputs and instead push it into a "lockup" condition with the two STM patterns "stuck" on whatever initial input produced them. The effect is illustrated in figure 16.15 for a TJXO pattern with $\gamma = 0.50$, a tenfold increase over the example system's base value of 0.05. Observe that the system had difficulty in switching from the T-pattern input to the J-pattern input (two well-separated patterns), and that the system failed completely to switch from the J to the X when that pattern arrived. Observe how the X-weight node $x_{23}$ never responded at all to the X input.

Fortunately, the range of values for $\gamma$ is rather broad and its precise setting does not seem to be particularly critical for proper system operation. In the example system the pattern weights are of unit length, meaning, for example, that the non-zero $z_{ji}$ weights for the T-vector are $1/\sqrt{6} = 0.408$. Typical $x_{ji}$ values appearing in the signal traces are about 0.4 peak, and so $\gamma = 0.05$ yields a per-pixel feedback signal amplitude of about 0.008. This compares to $I_1$ per-pixel values on the order of about 0.04 after the pattern has passed through $GN^{(2)}$, a 5:1 ratio of input to feedback.
Figure 16.15: Effect of too-large feedback gain on $R^{(1)}$. Note the difficulty with which the system switched from the T to the J pattern, and its inability to recognize the X pattern. The O pattern was entirely overridden by the J-weight feedback, producing a J-pattern in STM in place of the O-pattern input.

§ 4. Resonator 1 Dynamics with Continuous-valued Input Patterns

Operating as a pattern categorizer with fixed weights and in the absence of noise, $R^{(1)}$ is remarkably robust to flat changes in pattern amplitudes, i.e. equal scaling of each pixel value. Figure 16.16 illustrates the example of a T-X-T-X pattern sequence in which the pattern amplitude of the T pattern is 1 and the pattern amplitudes of the two X patterns are 0.5 and 0.01, respectively. Both X patterns are successfully reproduced in $v_1$ STM (figures B and C) with a mere 4:1 amplitude difference (despite the fact that the X inputs going into GN$^{(2)}$ had a 50:1 difference). The first X pattern was successfully recognized by $v_2$. The smaller of the two X patterns was not successfully recognized ($v_2$ was mildly "bewildered" as shown in A) but the $x_{23}$ ("X") signal did come out much stronger than the other two. Other than for an additional delay in the activation of $x_{23}$ following the application of the X patterns, the resonance characteristics showed little pattern-to-pattern variation.

If one thinks about this a little, this is a fairly remarkable accomplishment for $R^{(1)}$. How did these things happen? There are several factors at work here. Let us begin with the actions of GN$^{(2)}$
There is good reproduction of the STM, other than for an amplitude drop. The second X pattern was not cleanly encoded, as shown in (A), although $x_{23}$ did have the largest amplitude of the $x_j$ variables. There was no pattern noise applied in this simulation. Note that although the two X pattern inputs differed in amplitude by a factor of 50:1 going into $\text{GN}^{(2)}$, the STM amplitude difference is only about 4:1.

normalization. Here we apply equation (14.12). If the non-zero pixel amplitudes have amplitude $a$ then for the X pattern the total signal activity is $G = 9a$ (there are 9 non-zero pixels in X). For the non-zero pixels the relative amplitudes are $\omega_i = 1/9$. Using $\beta = (n - 1)C$, we rewrite (14.12) and obtain
Chapter 16: Basic ART Resonators

\[ x_i = \frac{n}{n-1} \cdot \frac{BG}{A+G} \left( \omega_i - \frac{1}{n} \right) \]

We have \( B = 1, A = 0.5, \) and \( n = 25 \) for our \( \text{GN}^{(2)} \). For \( a = 0.5 \), this gives us \( x_i = 0.0667 \). For \( a = 0.01 \) we get \( x_i = 0.0113 \). What was a 50:1 amplitude ratio going into \( \text{GN}^{(2)} \) is only a 5.9:1 ratio coming out. The Grossberg normalizer takes large amplitude spreads and compresses them into a much smaller range.

Next let us consider the absence of sequence interaction. Why did the STM set up by the large \( T \) pattern not overwhelm the smaller incoming \( X \) patterns? The reason is because of the quenching threshold dynamics of \( v_1 \). At the feedback level we are using, feedback from \( v_2 \) is not sufficient to sustain \( \text{STM}_1 \) when the input pattern changes significantly. Thus, \( \text{STM}_1 \) begins to decay, and it takes the \( x_{2j} \) values with it during its decline. Thus, with a Heaviside extractor activation function for the feedback, the feedback signals \( K_{ji} \) likewise fall. This is why the prior \( T \) activity could not block the smaller \( X \) activity from initiating a new STM. This can clearly be seen in figure 16.16A. It is true that the amount by which \( x_{21} \) must decay before \( x_{23} \) can "take over" is larger for larger amplitude differences between the \( T \) and \( X \) patterns. This is what causes the small increase in the time required for \( x_{23} \) to become active in 16.16A. But once it does, the resonance quickly develops and \( R^{(1)} \) takes itself into the new \( X \) state in its STM patterns.

Next we note that the \( X \) pattern in \( \text{STM}_1 \) does not contrast-enhance. This is because the final distribution dynamics in a Grossberg CE layer (which is what \( v_1 \) is) do not depend on absolute individual signal amplitudes but, rather, on the normalized amplitudes \( X_i = x_i/x \). This is why per-pixel signal amplitude does not affect the development of \( \text{STM}_1 \) in the absence of noise.

Why did \( v_2 \) fail to cleanly categorize the smaller \( X \) pattern? This is because the final \( v_2 \) distribution pattern does depend on total activity \( x(t) \), which is smaller for the small \( X \) pattern. Instead of contrast-enhancing the pattern into a 0-1 distribution, the \( v_2 \) layer was only able to produce a partially contrast-enhanced output for its \( x_{2j} \) variables. We saw this same thing in some of the examples from chapter 15.

Unfortunately, this happy state of affairs does not maintain when pattern noise is present. If we make the usual assumption that pattern noise is statistically independent of the patterns, pattern noise does not drop in amplitude along with the input pattern. The result of this is, as you perhaps have already realized, noise-induced contrast enhancement (NICE) as a function of input pattern amplitude. We will call this amplitude-induced contrast enhancement or AICE.

Figure 16.17 illustrates AICE in a TXTX sequence. In this simulation the noise distribution is \((0, 0.20)\), the \( T \) pattern has amplitude 1 in its non-zero elements, and the \( X \) pattern amplitudes are 0.5 and 0.1, respectively.
Chapter 16: Basic ART Resonators

Figure 16.17: TXTX sequence with pattern amplitude sensitive NICE (AICE). The T pattern amplitudes are 1.0. The X pattern amplitudes are 0.50 (B) and 0.10 (C). The pattern noise distribution is (0, 0.20). The contrast enhancement distortion in (B) is not severe enough to affect pattern categorization. The AICE in (C) produces a coding failure.

We have so far considered only input patterns with underlying pixel values of either 1 or 0, or patterns that are merely scaled versions of these. These kinds of patterns are typically called "binary" patterns because a pixel is either "on" or "off". It is true enough that contrast enhancement and additive noise produce continuous-valued pattern signals, but the terminology is
Figure 16.18: Analog pattern definitions. The label under each figure will be used to identify the pattern in the simulations. The new patterns are not contrast-enhanced. Their underlying pattern is defined as shown.

the same nonetheless. We now turn to consideration of patterns with continuous-valued underlying pixel definitions. These are commonly called "analog" patterns. Figure 16.18 illustrates several such patterns we will define and use in the next series of simulations. For ease of comparison, the figure sets these new "analog" patterns against their "binary counterparts" we have been using so far.
Chapter 16: Basic ART Resonator

The significance in introducing analog patterns lies partly in the phenomenon of *fine discrimination* and partly in its opposite, generalization. As an example of the first, if you know a pair of "identical twins" it is possible and common after a short while to be able to tell one from the other by noting subtle differences that exist in their appearances, while other people, who do not know them, cannot tell them apart. Examples of the second are provided by any general class of identifications such as "man", "dog", "automobile", etc. These sorts of feature distinction and object assimilation are what analog patterns such as those above are used to model. Our next topic, then, is how $R^{(1)}$ performs when given subtly different yet distinct input patterns. As we are about to see, contrast enhancement effects (NICE and AICE) are important impairments in classifying analog patterns.

We will begin with generalization. In our first example $R^{(1)}$ is presented with a pattern sequence $T_3$-$X_1$-$X_2$-$T_3$ with the fourth pattern scaled to one-half the magnitude of the first. We assume moreover that no pattern noise is added. Figure 16.19 gives the trace results of the simulation. Despite the differences between these patterns and the baseline patterns stored in the weights $W$ and $Z$, the system has no problem in properly classifying the patterns.

**Figure 16.19:** Simulation traces for $T_3$-$X_1$-$X_2$-$0.5T_3$ sequence. The system successfully classifies all four patterns.
Chapter 16: Basic ART Resonators

Now consider the patterns developed in STM during this simulation. These are illustrated in figure 16.20. Comparing these plots to the original patterns in figure 16.18, we see that significant contrast enhancement has been performed on the patterns. R\(^{(1)}\) has thus not only generalized the patterns according to its classifying vectors, but has done so despite this contrast enhancement. This illustrates the ability of the network to generalize. But why has such radical contrast enhancement occurred?

The mechanism is, of course, the same as that which underlies NICE and AICE and is found in Grossberg's contour-enhancement theorems of chapter 15. Indeed, one interesting way to look at an analog pattern \(A\) is to regard it as a scaled version of the sum of a binary pattern \(B\) plus a noise pattern \(N\) such that \(A = s \cdot (B + N)/|B+N|\) where \(s\) is some scale factor. By making \(N\) sufficiently large, "missing pixels" -- pixels of level zero in \(A\) that are non-zero in \(B\) -- can be approximated by letting \(N\) have a zero-value in that pixel location. From this way of looking at things, contrast enhancement of analog patterns can be viewed as another species of NICE.

Since R\(^{(1)}\) successfully decoded the pattern in the example above, is this contrast enhancement
Figure 16.21: Simulation results for T3-X1-T1-T3 sequence with noise (0, 0.10). $R^{(1)}$ fails to discriminate between patterns T1 and T3 in the first and fourth locations.

Figure 16.22: STM patterns for the T3-X1-T1-T3 pattern sequence simulation.

really a problem? The answer is "no" if we consider only the generalization problem, and "yes"
when we come to the fine discrimination problem. Suppose we wish to discriminate between patterns T1 and T3, and to this end we replace the classifying vector for J in R\(^{(1)}\) with a classifying vector T3. Figure 16.21 gives the results of simulating a sequence T3-X1-T1-T3 in the presence of additive pattern noise drawn from a uniform distribution over the range (0, 0.10), and figure 16.22 illustrates the STM patterns for the simulation.

\(R^{(1)}\) is unable to categorize the initial T3 pattern, producing \(x_{21}\) and \(x_{22}\) activations of almost the same magnitude. Comparing 16.22A against the T patterns of figure 16.18, we see that the contrast-enhanced STM differs from both. In one sense it is a "blending" of both patterns, which is easy enough to understand from figure 16.21, where we see both classifying vectors contributing in nearly equal amounts to the feedback vector \(K\). But the STM pattern is by no means merely the average of the two patterns; it has been contrast enhanced and not simply summed and scaled. The final result is, in a sense, as much like classifying vector T1 as it is for T3.

\(R^{(1)}\) successfully classifies X1 – where its task in this case has been merely generalization – and it correctly classifies T1, which has undergone only relatively modest contrast enhancement due in part to the additive noise and in part to some initial feedback contributions from the T3 classifying vector (see figure 16.21). But it completely errs in classifying the final T3, mistaking it for a continuation of T1. Figure 16.21 shows that the T3 classifying vector is never even brought into play. The reason for this is easy to understand: the abolition of STM when inputs change is based on the new pattern producing a more uniformly distributed excitation of \(v_1\). T1 and T3 are simply too much alike for the latter to abolish the STM left over from the former.

Would \(R^{(1)}\) have performed better on this simulation if there had been no noise? The answer is, "no, it does not perform differently in any significant way." There are limits to what the basic ART resonator network can do by itself.

§ 5. Resonator 2

Historically, resonator 1 is the oldest of the published ART resonators. It is not, however, the only kind of ART resonator. In this section we will look at a simpler but still very effective ART resonator. We will call it resonator 2 (\(R^{(2)}\) for short).

In \(R^{(1)}\) layer \(v_2\) does make some modest contribution to contrast enhancement of STM, but the majority of NICE effects are due to the lateral feedback connections in \(v_1\). This implies that NICE can be reduced if these connections are removed. \(R^{(2)}\) does just that, replacing \(v_1\) with a non-recurrent layer of the form

\[
\dot{x}_{1i} = -A \cdot x_{1i} + (B - x_{1i}) \cdot J_i^+ - (D + x_{1i}) \cdot J_i^-
\]  

(16.4)
where we continue to use our convention that $x_{1i}$ actually denotes $h(x_{1i})$, where $h$ is the Heaviside extractor activation function. $J_i^+$ is the total excitatory input to node $i$ and $J_i^-$ is the total inhibitory input to node $i$. In the $v_1$ layer of $R^{(2)}$, neither of these terms contain any contribution from $v_1$ and so $v_1$ is, except for its Heaviside extractor activation function, a linear network for all $x_{1i}$ with activity $> 0$. NICE is an inherently nonlinear effect, and this is why this new $v_1$ layer does not contribute to it. We will see that the remaining residual NICE contributed by $v_2$ is much smaller than the magnitude of the effect in $R^{(1)}$.

When the two $J$ terms are zero, (16.4) tells us $x_{1i}$ will undergo an exponential decay to zero at a rate determined by $A$. $A$ is often called the neper frequency and its inverse is a time constant, $\tau$.

With some simple algebraic manipulation, we can rewrite (16.4) as

$$\tau \cdot x_{1i} = -x_{1i} + \tau \cdot (B - x_{1i}) \cdot J_i^+ - \tau \cdot (D + x_{1i}) \cdot J_i^- .$$

Now, if $\tau$ is very small, the transient response of this system will be much faster than that of the $v_2$ layer and $v_1$ will reach its forced response while $v_2$ is still reverberating. Setting the derivative equal to zero (the forced response condition if $v_1$ responds much faster than both $v_2$ and the input signal rate) and solving for $x_{1i}$, we obtain

$$x_{1i} = \tau \cdot B \cdot \frac{J_i^+ - B_3 \cdot J_i^-}{1 + \tau \cdot J_i^+ + \tau \cdot J_i^-} , \quad B_3 = \frac{D}{B} . \quad (16.5)$$

We will use (16.5) as our model equation for $v_1$ in $R^{(2)}$. We will designate the Instars in this layer as $\text{SNI}^{(4)}$.

It still remains to define the two $J$ signals. For $J_i^+$ we use (16.1) and obtain

$$J_i^+ = I_i + \gamma \cdot \sum_{j=1}^{N} z_{ij} \cdot h_2(x_{2j}) \quad (16.6)$$

which is the same excitation used by $R^{(1)}$. Here $h_2$ is the Heaviside extractor. For $J_i^-$ we use

$$J_i^- = \sum_{j=1}^{N} h_2(x_{2j}) . \quad (16.7)$$

Expression (16.7) is called a non-specific activity in ART terminology. Its employment in (16.4,5) is called a non-specific inhibition [GROS6, 16], [CARP2]. In the anatomy of $R^{(2)}$, the term $B_3 \cdot J_i^-$ is also called an attentional gain control because it exerts an effect by which $v_1$ is influenced differently by input signals $I$ than by $v_2$ excitatory feedback signals $K$. Replacing $v_1$ by the system defined by equations (16.5)-(16.7) gives us the system depicted in figure 16.23.
The behavior of this anatomy is illustrated in the following simulations. In these simulations we use \( GN^{(2)} \) for the normalizer and the same \( v_2 \) layer as used for the \( R^{(1)} \) simulations. The parameters for these subsystems remain unchanged from their previous values. For the \( v_1 \) layer the parameters are: (1) \( \tau \cdot B = 1 \); (2) \( \tau = 0.001 \); (3) \( B_3 = \mu \cdot \gamma \) with \( \mu = 0.5 \) and \( \gamma \) (the feedback gain for \( K \) from \( v_2 \)) equal to 0.05 as in \( R^{(1)} \).

**Figure 16.23:** Resonator 2 system.

The input parameters for this test sequence are the same as used previously in the first \( R^{(1)} \) examples. All patterns have the same relative weighting. The additive pattern noise distribution is \( (0, 0.20) \). The network correctly classifies the T, J, and X patterns. There is very little NICE in STM$_1$. Pattern O is misclassified and its STM$_1$ is recognizably the O pattern.

**Figure 16.24:** Standard TJXO sequence for \( R^{(2)} \). The input parameters for this test sequence are the same as used previously in the first \( R^{(1)} \) examples. All patterns have the same relative weighting. The additive pattern noise distribution is \( (0, 0.20) \). The network correctly classifies the T, J, and X patterns. There is very little NICE in STM$_1$. Pattern O is misclassified and its STM$_1$ is recognizably the O pattern.
Chapter 16: Basic ART Resonators

The first simulation is a repeat of the TJXO series we first used to examine $R^{(1)}$. Figure 16.24 provides the signal traces for this simulation. Acting as a classifier, the performance of $R^{(2)}$ is equivalent to that of $R^{(1)}$ for the known patterns, and more or less the same for the unknown pattern O. The same continues to hold true for the other pattern tests used earlier. The differences between the two resonators shows up in STM$_1$. We may first note that although $v_1$ has a very fast response to signals (as it should), the total amount of time to process the incoming signals is not significantly different from that of $R^{(1)}$; layer $v_2$ dominates this aspect of the system dynamics. The biggest difference is the near-total lack of contrast enhancement distortion (NICE) in STM$_1$. Almost all the pattern distortion in STM$_1$ is directly due to the additive pattern noise itself, which is to be expected. This point will be demonstrated in the next simulation, where the performance advantage of $R^{(2)}$ in regard to NICE is more graphically illustrated. STM$_1$ plots are not presented for the present simulation because, quite frankly, they are boring. The T, the J, and the X come out looking like T, J, and X with all background noise squelched and only a small amount of additive noise visible in the pattern pixels. This is demonstrated by the mismatch metric in 16.24.

Our second simulation is a T3-X1-T1-J sequence (all patterns equally weighted) with pattern noise distribution (0, 0.20). This test is almost identical to the simulation of figure 16.22, with the use of the J pattern at the end being the only difference of consequence. Figure 16.25 shows the signal traces for this simulation, and figure 16.26 provides the STM$_1$ patterns.

Looking first at figure 16.25, we observe that all four patterns were properly classified by $R^{(2)}$. Like $R^{(1)}$, $R^{(2)}$ is incapable of distinguishing the alternative T and X patterns given weights that are exact copies of these patterns. The mismatch metric is significantly less than for $R^{(1)}$, which is

![Figure 16.25: Signal traces for T3-X1-T1-J pattern sequence. $R^{(2)}$ correctly classifies all four patterns.](image)
Chapter 16: Basic ART Resonators

Figure 16.26: STM1 plots for the T3-X1-T1-J sequence. Note that there is very little distortion of the STM patterns. Comparison of these patterns with figure 16.22 demonstrates that NICE is suppressed by R(2) leaving primarily additive pattern noise distortion only. Close examination of the numerical results of the simulation shows that some NICE does persist, but it is negligible compared to the additive pattern noise.

Note also that the noise in this simulation is twice the amplitude of that used for figure 16.22.

A reflection of the absence of NICE effects in R(2). It is to be noted that the additive pattern noise distribution for this simulation is twice the amplitude of that used previously in the R(1) example simulation.

Next we examine figure 16.26. Comparing this figure to figure 16.22 from the R(1) simulation, we can easily see the dramatic improvement in preserving the input pattern in STM1 realized by R(2). Again, this better copy-fidelity is despite the fact that twice as much pattern noise was added prior to GN(2). Other simulations, corresponding to figure 16.21, show that the failure of R(2) to be able to make the fine discrimination between, say, T3 and T1 is not due to the STM but, rather, to the settings of the weights themselves. As we will see in chapter 17, the fine discrimination problem is fundamentally a problem Carpenter and Grossberg call the subset-superset problem [CARP2]. Its solution is intimately tied in with the solution of the stability-plasticity issue in adaptive neural network systems. This is something we take up in chapter 17.
§ 6. Summary

This chapter has introduced two basic resonator networks of the sort commonly used in ART network system models. By themselves, these networks do not yet constitute a full ART system. They are resonators, but we have not yet made them adaptive resonators. We will see in the next chapter that adaptation, methods to address fine discrimination, and the stability-plasticity problem require additions to the network system for controlling the adaptation process itself.

In this chapter we have given our attention to the basic resonance dynamics in play in minimal ART resonator networks. The theory presented here is largely comes from Grossberg's 1976 paper [GROS6], which preceded the first artificial ART network, ART 1 [CARP1], by almost a decade. As mentioned at the beginning of this chapter, our focus in this text is biological signal processing and computational neuroscience rather than engineering applications of ART. However, the material presented here should be of great help to those interested in engineering neural networks, and in helping to understand some of the thinking that goes into the classic ART papers of Carpenter and Grossberg.

Exercises

1. x