19 Adaptive Resonance Theory

Objectives

In Chapter 16 and Chapter 18 we learned that one key problem of competitive networks is the stability of learning. There is no guarantee that, as more inputs are applied to the network, the weight matrix will eventually converge. In this chapter we will present a modified type of competitive learning, called adaptive resonance theory (ART), which is designed to overcome the problem of learning stability.

Theory and Examples

A key problem of the Grossberg network presented in Chapter 18 and the competitive networks of Chapter 16, is that they do not always form stable clusters (or categories). Grossberg did show [Gros76] that if the number of input patterns is not too large, or if the input patterns do not form too many clusters relative to the number of neurons in Layer 2, then the learning eventually stabilizes. However, he also showed that the standard competitive networks do not have stable learning in response to arbitrary input patterns. The learning instability occurs because of the network's adaptability (or plasticity), which causes prior learning to be eroded by more recent learning.

Stability/Plasticity

Grossberg refers to this problem as the "*stability / plasticity dilemma*." How can a system be receptive to significant new patterns and yet remain stable in response to irrelevant patterns? We know that biological systems are very good at this. For example, you can easily recognize your mother's face, even if you have not seen her for a long time and have met many new people in the mean time.

Grossberg and Gail Carpenter developed a theory, called adaptive resonance theory (ART), to address the stability/plasticity dilemma (see [CaGr87a], [CaGr87b], [CaGr90], [CaGrRe91] and [CaGrMa92]). The ART networks are based on the Grossberg network of Chapter 18. The key innovation of ART is the use of "expectations." As each input pattern is presented to the network, it is compared with the prototype vector that it most closely matches (the expectation). If the match between the prototype and the input vector is not adequate, a new prototype is selected. In this way, previously learned memories (prototypes) are not eroded by new learning.

It is beyond the scope of this text to discuss all of the variations of adaptive resonance theory. Instead, we will present one of the ART networks in detail — ART1 (see [CaGr87a]). This particular network is designed for binary input vectors only. However, from this one architecture, the key features of adaptive resonance theory can be understood.

Overview of Adaptive Resonance

The basic ART architecture is shown in Figure 19.1. It is a modification of the Grossberg network of Chapter 18 (compare with Figure 18.13), which is designed to stabilize the learning process. The innovations of the ART architecture consist of three parts: Layer 2 (L2) to Layer 1 (L1) expectations, the orienting subsystem and gain control. In this section we will describe the general operation of the ART system; then, in later sections, we will discuss each subsystem in detail.

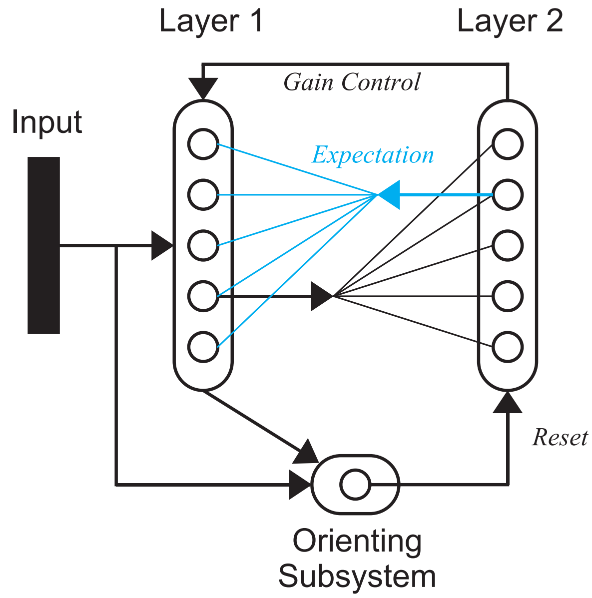


Figure 19.1 Basic ART Architecture

Recall from Chapter 18 that the L1-L2 connections of the Grossberg network are instars, which perform a clustering (or categorization) operation. When an input pattern is presented to the network, it is multiplied (after normalization) by the L1-L2 weight matrix. Then, a competition is performed at Layer 2 to determine which row of the weight matrix is closest to the input vector. That row is then moved toward the input vector. After learning is complete, each row of the L1-L2 weight matrix is a prototype pattern, which represents a cluster (or category) of input vectors.

In the ART networks, learning also occurs in a set of feedback connections from Layer 2 to Layer 1. These connections are outstars (see Chapter 15), which perform pattern recall. When a node in Layer 2 is activated, this reproduces a prototype pattern (the expectation) at Layer 1. Layer 1 then performs a comparison between the expectation and the input pattern.

When the expectation and the input pattern are not closely matched, the orienting subsystem causes a reset in Layer 2. This reset disables the current winning neuron, and the current expectation is removed. A new competition is then performed in Layer 2, while the previous winning neuron is disabled. The new winning neuron in Layer 2 projects a new expectation to Layer 1, through the L2-L1 connections. This process continues until the L2-L1 expectation provides a close enough match to the input pattern.

In the following sections we will investigate each of the subsystems of the ART system, as they apply to one particular ART network — ART1 ([CaGr87a]). We will first describe the differential equations that describe the subsystem operations. Then we will derive the steady state responses of each subsystem. Finally, we will summarize the overall operation of the ART1 system.

Layer 1

The main purpose of Layer 1 is to compare the input pattern with the expectation pattern from Layer 2. (*Both patterns are binary in ART1.*) If the patterns are not closely matched, the orienting subsystem will cause a reset in Layer 2. If the patterns are close enough, Layer 1 combines the expectation and the input to form a new prototype pattern.

Layer 1 of the ART1 network, which is displayed in Figure 19.2, is very similar to Layer 1 of the Grossberg network (see Figure 18.14). The differences occur at the excitatory and inhibitory inputs to the shunting model. For the ART1 network, no normalization is performed at Layer 1; therefore we don't have the on-center/off-surround connections from the input vector. The excitatory input to Layer 1 of ART1 consists of a combination of the input pattern and the L1-L2 expectation. The inhibitory input consists of the gain control signal from Layer 2. In the following we will explain how these inputs work together.

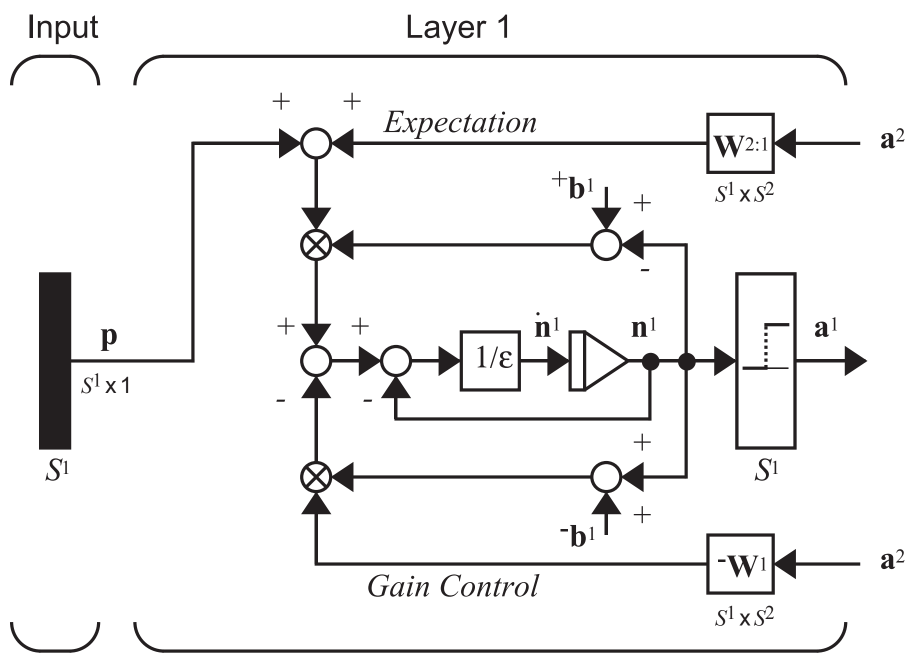


Figure 19.2 Layer 1 of the ART1 Network

The equation of operation of Layer 1 is

$$\varepsilon\frac{d\mathbf{n}^1(t)}{dt} = -\mathbf{n}^1(t) + ({}^{+}\mathbf{b}^1 - \mathbf{n}^1(t))\{\mathbf{p} + \mathbf{W}^{2:1}\mathbf{a}^2(t)\} - (\mathbf{n}^1(t) + {}^{-}\mathbf{b}^1)[{}^{-}\mathbf{W}^1]\mathbf{a}^2(t) \qquad (19.1)$$

and the output of Layer 1 is computed

$$\mathbf{a}^1 = \mathbf{hardlim}^+(\mathbf{n}^1), \tag{19.2}$$

where

$$hardlim^+(n) = \begin{cases} 1, & n > 0 \\ 0, & n \le 0 \end{cases}. \tag{19.3}$$

Eq. (19.1) is a shunting model with excitatory input $\mathbf{p} + \mathbf{W}^{2:1}\mathbf{a}^2(t)$, which is the sum of the input vector and the L2-L1 expectation. For example, assume that the jth neuron in Layer 2 has won the competition, so that its output is 1, and the other neurons have zero output. For this case we have

$$\mathbf{W}^{2:1}\mathbf{a}^2 = \begin{bmatrix} \mathbf{w}_1^{2:1} & \mathbf{w}_2^{2:1} & \ldots & \mathbf{w}_j^{2:1} & \ldots & \mathbf{w}_{S^2}^{2:1} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \end{bmatrix} = \mathbf{w}_j^{2:1}, \tag{19.4}$$

where $\mathbf{w}_j^{2:1}$ is the jth column of the matrix $\mathbf{W}^{2:1}$. (The $\mathbf{W}^{2:1}$ matrix is trained using an outstar rule, as we will show in a later section.) Now we can see that

$$\mathbf{p} + \mathbf{W}^{2:1}\mathbf{a}^2 = \mathbf{p} + \mathbf{w}_j^{2:1}. \tag{19.5}$$

Therefore the excitatory input to Layer 1 is the sum of the input pattern and the L2-L1 expectation. Each column of the L2-L1 matrix represents a different expectation (prototype pattern). Layer 1 combines the input pattern with the expectation using an AND operation, as we will see later.

The inhibitory input to Layer 1 is the gain control term $[{}^{-}\mathbf{W}^1]\mathbf{a}^2(t)$, where

$${}^{-}\mathbf{W}^1 = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{19.6}$$

Therefore, the inhibitory input to each neuron in Layer 1 is the sum of all of the outputs of Layer 2. Since we will be using a winner-take-all competition in Layer 2, whenever Layer 2 is active there will be one, and only one, nonzero element of $\mathbf{a}^2$ after the competition. Therefore the gain control input to Layer 1 will be one when Layer 2 is active, and zero when Layer 2 is inactive (all neurons having zero output). The purpose of this gain control will become apparent as we analyze the steady state behavior of Layer 1.

Steady State Analysis

The response of neuron i in Layer 1 is described by

$$\varepsilon \frac{dn_i^1}{dt} = -n_i^1 + ({}^{+}b^1 - n_i^1)\left\{ p_i + \sum_{j=1}^{S^2} w_{i,j}^{2:1} a_j^2 \right\} - (n_i^1 + {}^{-}b^1) \sum_{j=1}^{S^2} a_j^2 , \qquad (19.7)$$

where $\varepsilon \ll 1$ so that the short-term memory traces (the neuron outputs) change much faster than the long-term memory traces (the weight matrices).

We want to investigate the steady state response of this system for two different cases. In the first case Layer 2 is inactive, therefore $a_j^2 = 0$ for all j. In the second case Layer 2 is active, and therefore one neuron has an output of 1, and all other neurons output 0.

Consider first the case where Layer 2 is inactive. Since each $a_j^2 = 0$, Eq. (19.7) simplifies to

$$\varepsilon \frac{dn_i^1}{dt} = -n_i^1 + ({}^{+}b^1 - n_i^1)\{p_i\} . \qquad (19.8)$$

In the steady state ($dn_i^1(t)/dt = 0$) we have

$$0 = -n_i^1 + ({}^{+}b^1 - n_i^1)p_i = -(1 + p_i)n_i^1 + {}^{+}b^1 p_i . \qquad (19.9)$$

If we solve for the steady state neuron output n_i^1 we find

$$n_i^1 = \frac{{}^{+}b^1 p_i}{1 + p_i} . \qquad (19.10)$$

Therefore, if $p_i = 0$ then $n_i^1 = 0$, and if $p_i = 1$ then $n_i^1 = {}^{+}b^1/2 > 0$. Since we chose the transfer function for Layer 1 to be the $hardlim^+$ function, then we have

$$\mathbf{a}^1 = \mathbf{p} . \qquad (19.11)$$

Therefore, when Layer 2 is inactive, the output of Layer 1 is the same as the input pattern.

Now let's consider the second case, where Layer 2 is active. Assume that neuron j is the winning neuron in Layer 2. Then $a_j^2 = 1$ and $a_k^2 = 0$ for $k \neq j$. For this case Eq. (19.7) simplifies to

$$\varepsilon \frac{dn_i^1}{dt} = -n_i^1 + ({}^{+}b^1 - n_i^1)\{p_i + w_{i,j}^{2:1}\} - (n_i^1 + {}^{-}b^1) . \tag{19.12}$$

In the steady state ($dn_i^1(t)/dt = 0$) we have

$$\begin{aligned} 0 &= -n_i^1 + ({}^{+}b^1 - n_i^1)\{p_i + w_{i,j}^{2:1}\} - (n_i^1 + {}^{-}b^1) \\ &= -(1 + p_i + w_{i,j}^{2:1} + 1)n_i^1 + ({}^{+}b^1(p_i + w_{i,j}^{2:1}) - {}^{-}b^1) . \end{aligned} \tag{19.13}$$

If we solve for the steady state neuron output n_i^1 we find

$$n_i^1 = \frac{{}^{+}b^1(p_i + w_{i,j}^{2:1}) - {}^{-}b^1}{2 + p_i + w_{i,j}^{2:1}} . \tag{19.14}$$

Recall that Layer 1 should combine the input vector with the expectation from Layer 2 (represented by $\mathbf{w}_j^{2:1}$). Since we are dealing with binary patterns (both the input and the expectation), we will use a logical AND operation to combine the two vectors. In other words, we want n_i^1 to be less than zero when either p_i or $w_{i,j}^{2:1}$ is equal to zero, and we want n_i^1 to be greater than zero when both p_i and $w_{i,j}^{2:1}$ are equal to one.

If we apply these conditions to Eq. (19.14), we obtain the following equations:

$${}^{+}b^1(2) - {}^{-}b^1 > 0 , \tag{19.15}$$

$${}^{+}b^1 - {}^{-}b^1 < 0 , \tag{19.16}$$

which we can combine to produce

$${}^{+}b^1(2) > {}^{-}b^1 > {}^{+}b^1 . \tag{19.17}$$

For example, we can use ${}^{+}b^1 = 1$ and ${}^{-}b^1 = 1.5$ to satisfy these conditions.

Therefore, if Eq. (19.17) is satisfied, and neuron j of Layer 2 is active, then the output of Layer 1 will be

$$\mathbf{a}^1 = \mathbf{p} \cap \mathbf{w}_j^{2:1} , \tag{19.18}$$

where $\cap$ represents the logical AND operation.

Notice that we needed the gain control in order to implement the AND operation. Consider the numerator of Eq. (19.14):

$${}^{+}b^1(p_i + w_{i,j}^{2:1}) - {}^{-}b^1 . \tag{19.19}$$

The term ${}^{-}b^{1}$ is multiplied by the gain control term, which in this case is 1. If this term did not exist, then Eq. (19.19) would be greater than zero (and therefore n_i^1 would be greater than zero) whenever either p_i or $w_{i,j}^{2:1}$ was greater than zero. This would represent an OR operation, rather than an AND operation. As we will see when we discuss the orienting subsystem, it is critical that Layer 1 perform an AND operation.

When Layer 2 is inactive, the gain control term is zero. This is necessary because in that case we want Layer 1 to respond to the input pattern alone, since no expectation will be activated by Layer 2.

To summarize the steady state operation of Layer 1:

If Layer 2 is not active (i.e., each $a_j^2 = 0$),

$$\mathbf{a}^1 = \mathbf{p} . \tag{19.20}$$

If Layer 2 is active (i.e., one $a_j^2 = 1$),

$$\mathbf{a}^1 = \mathbf{p} \cap \mathbf{w}_j^{2:1} . \tag{19.21}$$

To demonstrate the operation of Layer 1, assume the following network parameters:

$$\varepsilon = 0.1 , \; {}^{+}b^{1} = 1 \text{ and } {}^{-}b^{1} = 1.5 . \tag{19.22}$$

Assume also that we have two neurons in Layer 2, two elements in the input vector and the following weight matrix and input:

$$\mathbf{W}^{2:1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \text{ and } \mathbf{p} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} . \tag{19.23}$$

If we take the case where Layer 2 is active, and neuron 2 of Layer 2 wins the competition, the equations of operation of Layer 1 are

$$(0.1)\frac{dn_1^1}{dt} = -n_1^1 + (1 - n_1^1)\{p_1 + w_{1,2}^{2:1}\} - (n_1^1 + 1.5) \tag{19.24}$$

$$= -n_1^1 + (1 - n_1^1)\{0 + 1\} - (n_1^1 + 1.5) = -3n_1^1 - 0.5$$

$$(0.1)\frac{dn_2^1}{dt} = -n_2^1 + (1 - n_2^1)\{p_2 + w_{2,2}^{2:1}\} - (n_2^1 + 1.5) \tag{19.25}$$

$$= -n_2^1 + (1 - n_2^1)\{1 + 1\} - (n_2^1 + 1.5) = -4n_2^1 + 0.5 .$$

These can be simplified to obtain

$$\frac{dn_1^1}{dt} = -30n_1^1 - 5, \tag{19.26}$$

$$\frac{dn_2^1}{dt} = -40n_2^1 + 5. \tag{19.27}$$

In this simple case we can find closed-form solutions for these equations. If we assume that both neurons start with zero initial conditions, the solutions are

$$n_1^1(t) = -\frac{1}{6}[1 - e^{-30t}], \tag{19.28}$$

$$n_2^1(t) = \frac{1}{8}[1 - e^{-40t}]. \tag{19.29}$$

These are displayed in Figure 19.3.

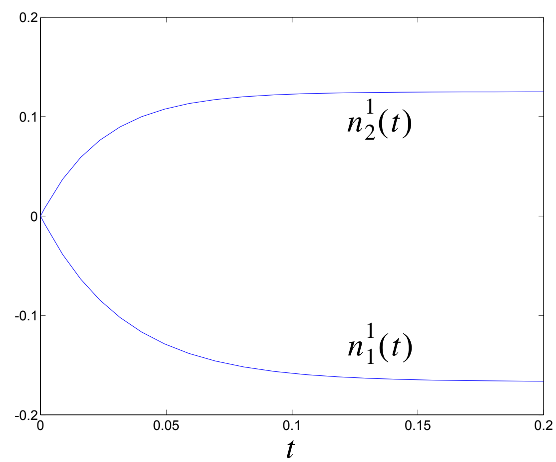
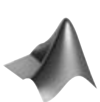


Figure 19.3 Response of Layer 1

Note that $n_1^1(t)$ converges to a negative value, and $n_2^1(t)$ converges to a positive value. Therefore, $a_1^1(t)$ converges to 0, and $a_2^1(t)$ converges to 1 (recall that the transfer function for Layer 1 is $hardlim^+$). This agrees with our steady state analysis (see Eq. (19.21)), since

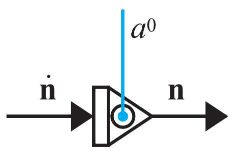
$$\mathbf{p} \cap {}_2\mathbf{w}^{2:1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cap \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \mathbf{a}^1. \tag{19.30}$$



To experiment with Layer 1 of the ART1 network, use the Neural Network Design Demonstration ART1 Layer 1 (`nnd16al1`).

Layer 2

Layer 2 of the ART1 network is almost identical to Layer 2 of the Grossberg network of Chapter 18. Its main purpose is to contrast enhance its output pattern. For our implementation of the ART1 network, the contrast enhancement will be a winner-take-all competition, so only the neuron that receives the largest input will have a nonzero output.



There is one major difference between the second layers of the Grossberg and the ART1 networks. Layer 2 of the ART1 network uses an integrator that can be reset. In this type of integrator, whose symbol is shown in the left margin, any positive outputs are reset to zero whenever the a^0 signal becomes positive. The outputs that are reset remain inhibited for a long period of time, so that they cannot be driven above zero. (By a “long” period of time we mean until an adequate match has occurred and the weights have been updated.)

In the original ART1 paper, Carpenter and Grossberg suggested that the reset mechanism could be implemented using a gated dipole field [CaGr87]. They later suggested a more sophisticated biological model, using chemical neurotransmitters, in their ART3 architecture [CaGr90]. For our purposes we will not be concerned with the specific biological implementation.

Figure 19.4 displays the complete Layer 2 of the ART1 network. Again, it is almost identical to Layer 2 of the Grossberg network (see Figure 18.16), with the primary exception of the resetable integrator. The reset signal, a^0, is the output of the orienting subsystem, which we will discuss in the next section. It generates a reset whenever there is a mismatch at Layer 1 between the input signal and the L2-L1 expectation.

One other small difference between Layer 2 of the ART1 network and Layer 2 of the Grossberg network is that two transfer functions are used in ART1. The transfer function $\mathbf{f}^2(\mathbf{n}^2)$ is used for the on-center/off-surround feedback connections, while the output of Layer 2 is computed as $\mathbf{a}^2 = \mathbf{hardlim}^+(\mathbf{n}^2)$. The reason for the second transfer function is that we want the output of Layer 2 to be a binary signal.

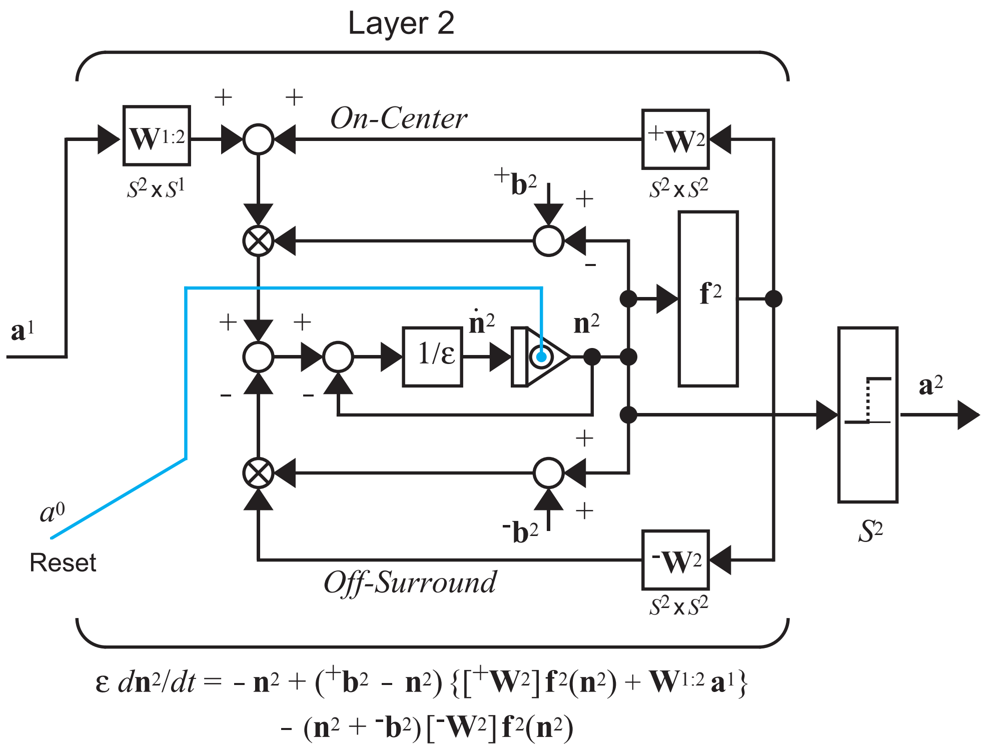


Figure 19.4 Layer 2 of the ART1 Network

The equation of operation of Layer 2 is

$$\varepsilon \frac{d\mathbf{n}^2(t)}{dt} = -\mathbf{n}^2(t) + ({}^{+}\mathbf{b}^2 - \mathbf{n}^2(t))\{[{}^{+}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t)) + \mathbf{W}^{1:2}\mathbf{a}^1\}$$
$$-(\mathbf{n}^2(t) + {}^{-}\mathbf{b}^2)[{}^{-}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t)) \,. \quad (19.31)$$

This is a shunting model with excitatory input $\{[{}^{+}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t)) + \mathbf{W}^{1:2}\mathbf{a}^1\}$, where ${}^{+}\mathbf{W}^2$ provides on-center feedback connections (identical to Layers 1 and 2 of the Grossberg network of Chapter 18, Eq. (18.6)), and $\mathbf{W}^{1:2}$ consists of adaptive weights, analogous to the weights in the Kohonen network. They are trained according to an instar rule, as we will see in a later section. The rows of $\mathbf{W}^{1:2}$, after training, will represent the prototype patterns.

The inhibitory input to the shunting model is $[{}^{-}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t))$, where ${}^{-}\mathbf{W}^2$ provides off-surround feedback connections (identical to Layers 1 and 2 of the Grossberg network — Eq. (18.7)).

2 +2 4

To illustrate the performance of Layer 2, consider a two-neuron layer with

$$\varepsilon = 0.1\,,\ {}^{+}\mathbf{b}^2 = \begin{bmatrix}1\\1\end{bmatrix},\ {}^{-}\mathbf{b}^2 = \begin{bmatrix}1\\1\end{bmatrix},\ \mathbf{W}^{1:2} = \begin{bmatrix}({}_1\mathbf{w}^{1:2})^T\\({}_2\mathbf{w}^{1:2})^T\end{bmatrix} = \begin{bmatrix}0.5 & 0.5\\1 & 0\end{bmatrix}, \tag{19.32}$$

and

$$f^2(n) = \begin{cases}10(n)^2, & n \ge 0\\ 0\,, & n < 0\end{cases}. \tag{19.33}$$

The equations of operation of the layer will be

$$(0.1)\frac{dn_1^2(t)}{dt} = -n_1^2(t) + (1 - n_1^2(t))\{f^2(n_1^2(t)) + ({}_1\mathbf{w}^{1:2})^T\mathbf{a}^1\} - (n_1^2(t) + 1)f^2(n_2^2(t)) \tag{19.34}$$

$$(0.1)\frac{dn_2^2(t)}{dt} = -n_2^2(t) + (1 - n_2^2(t))\{f^2(n_2^2(t)) + ({}_2\mathbf{w}^{1:2})^T\mathbf{a}^1\} - (n_2^2(t) + 1)f^2(n_1^2(t))\,. \tag{19.35}$$

This is identical in form to the Grossberg Layer 2 example in Chapter 18 (see Eq. (18.20) and Eq. (18.21)), except that ${}^{-}b^2 = 1$. This will allow $n_1^2(t)$ and $n_2^2(t)$ to range between -1 and +1.

The inputs to Layer 2 are the inner products of the prototype patterns (rows of the weight matrix $\mathbf{W}^{1:2}$) with the output of Layer 1. (The rows of this weight matrix are normalized, as will be explained in a later section.) The largest inner product will correspond to the prototype pattern that is closest to the output of Layer 1. Layer 2 then performs a competition between the neurons. The transfer function $f^2(n)$ is chosen to be a faster-than-linear transfer function (see Chapter 18, page 18-20, for a discussion of the effect of $f^2(n)$). This choice will force the neuron with largest input to have a positive n, and the other neuron to have a negative n (with appropriate choice of network parameters). After the competition, one neuron output will be 1, and the other neuron output will be zero, since we are using the $hardlim^+$ transfer function to compute the layer output.

Figure 19.5 illustrates the response of Layer 2 when the input vector is $\mathbf{a}^1 = \begin{bmatrix}1 & 0\end{bmatrix}^T$. The second row of $\mathbf{W}^{1:2}$ has a larger inner product with $\mathbf{a}^1$

than the second row, therefore neuron 2 wins the competition. At steady state, $n_2^2(t)$ has a positive value, and $n_1^2(t)$ has a negative value. The steady state Layer 2 output will then be

$$\mathbf{a}^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} . \tag{19.36}$$

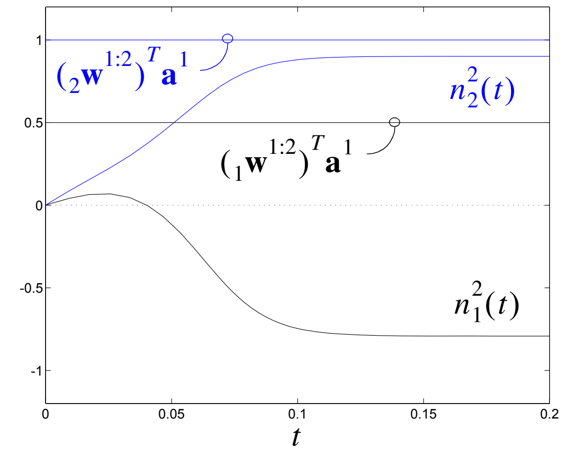
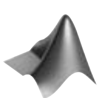


Figure 19.5 Response of Layer 2

We can summarize the steady state operation of Layer 2 as follows:

$$a_i^2 = \begin{cases} 1, & \text{if}(({}_i\mathbf{w}^{1:2})^T\mathbf{a}^1 = max[({}_j\mathbf{w}^{1:2})^T\mathbf{a}^1]) \\ 0, & \text{otherwise} \end{cases} . \tag{19.37}$$



To experiment with Layer 2 of the ART1 network, use the Neural Network Design Demonstration ART1 Layer 2 (`nnd16al2`).

Orienting Subsystem

One of the key elements of the ART architecture is the Orienting Subsystem. Its purpose is to determine if there is a sufficient match between the L2-L1 expectation and the input pattern. When there is not enough of a match, the Orienting Subsystem should send a reset signal to Layer 2. The reset signal will cause a long-lasting inhibition of the previous winning neuron, and thus allow another neuron to win the competition.

Figure 19.6 displays the Orienting Subsystem.

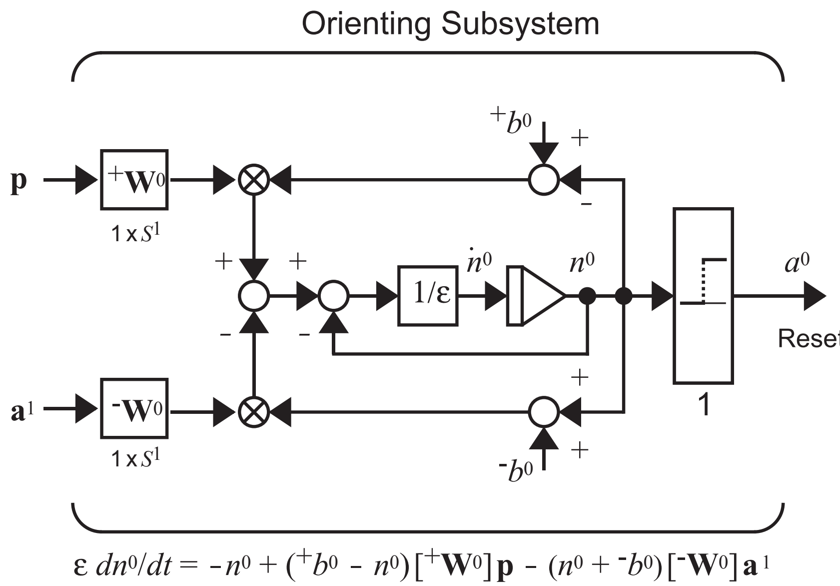


Figure 19.6 Orienting Subsystem of the ART1 Network

The equation of operation of the Orienting Subsystem is

$$\varepsilon\frac{dn^0(t)}{dt} = -n^0(t) + ({}^{+}b^0 - n^0(t))\{{}^{+}\mathbf{W}^0\mathbf{p}\} - (n^0(t) + {}^{-}b^0)\{{}^{-}\mathbf{W}^0\mathbf{a}^1\}\,. \quad (19.38)$$

This is a shunting model, with excitatory input ${}^{+}\mathbf{W}^0\mathbf{p}$, where

$${}^{+}\mathbf{W}^0 = \begin{bmatrix}\alpha & \alpha & \dots & \alpha\end{bmatrix}\,. \quad (19.39)$$

Therefore, the excitatory input can be written

$${}^{+}\mathbf{W}^0\mathbf{p} = \begin{bmatrix}\alpha & \alpha & \dots & \alpha\end{bmatrix}\mathbf{p} = \alpha\sum_{j=1}^{S^1} p_j = \alpha\|\mathbf{p}\|^2\,, \quad (19.40)$$

where the last equality holds because $\mathbf{p}$ is a binary vector.

The inhibitory input to the Orienting Subsystem is ${}^{-}\mathbf{W}^0\mathbf{a}^1$, where

$${}^{-}\mathbf{W}^0 = \begin{bmatrix}\beta & \beta & \dots & \beta\end{bmatrix}\,. \quad (19.41)$$

Therefore, the inhibitory input can be written

$${}^{-}\mathbf{W}^0\mathbf{a}^1 = \begin{bmatrix}\beta & \beta & \dots & \beta\end{bmatrix}\mathbf{a}^1 = \beta\sum_{j=1}^{S^1} a_j^1(t) = \beta\|\mathbf{a}^1\|^2\,. \quad (19.42)$$

Whenever the excitatory input is larger than the inhibitory input, the Orienting Subsystem will be driven on. Consider the following steady state operation:

$$\begin{aligned} 0 &= -n^0 + ({}^{+}b^0 - n^0)\{\alpha\|\mathbf{p}\|^2\} - (n^0 + {}^{-}b^0)\{\beta\|\mathbf{a}^1\|^2\} \\ &= -(1 + \alpha\|\mathbf{p}\|^2 + \beta\|\mathbf{a}^1\|^2)n^0 + {}^{+}b^0(\alpha\|\mathbf{p}\|^2) - {}^{-}b^0(\beta\|\mathbf{a}^1\|^2) \,. \end{aligned} \quad (19.43)$$

If we solve for n^0, we find

$$n^0 = \frac{{}^{+}b^0(\alpha\|\mathbf{p}\|^2) - {}^{-}b^0(\beta\|\mathbf{a}^1\|^2)}{(1 + \alpha\|\mathbf{p}\|^2 + \beta\|\mathbf{a}^1\|^2)} \,. \quad (19.44)$$

Let ${}^{+}b^0 = {}^{-}b^0 = 1$, then $n^0 > 0$ if $\alpha\|\mathbf{p}\|^2 - \beta\|\mathbf{a}^1\|^2 > 0$, or in other words:

$$n^0 > 0 \text{ if } \frac{\|\mathbf{a}^1\|^2}{\|\mathbf{p}\|^2} < \frac{\alpha}{\beta} = \rho \,. \quad (19.45)$$

Vigilance

This is the condition that will cause a reset of Layer 2, since $a^0 = hardlim^{+}(n^0)$. The term ρ is called the *vigilance* parameter, and must fall in the range $0 < \rho < 1$. If the vigilance is close to 1, a reset will occur unless $\mathbf{a}^1$ is close to $\mathbf{p}$. If the vigilance is close to 0, $\mathbf{a}^1$ need not be close to $\mathbf{p}$ to prevent a reset. The vigilance parameter determines the coarseness of the categorization (or clustering) created by the prototype vectors.

Recall from Eq. (19.21) that $\mathbf{a}^1 = \mathbf{p} \cap \mathbf{w}_j^{2:1}$ whenever Layer 2 is active. Therefore, $\|\mathbf{p}\|^2$ will always be greater than or equal to $\|\mathbf{a}^1\|^2$. They will be equal when the expectation $\mathbf{w}_j^{2:1}$ has a 1 wherever the input $\mathbf{p}$ has a 1. Therefore, the orienting subsystem will cause a reset when there is enough of a mismatch between $\mathbf{p}$ and $\mathbf{w}_j^{2:1}$. The amount of mismatch required for a reset is determined by the vigilance parameter ρ.

To demonstrate the operation of the Orienting Subsystem, suppose that $\varepsilon = 0.1$, $\alpha = 3$, $\beta = 4$ ($\rho = 0.75$),

$$\mathbf{p} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \text{ and } \mathbf{a}^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} . \quad (19.46)$$

The equation of operation becomes

$$\begin{aligned} (0.1)\frac{dn^0(t)}{dt} &= -n^0(t) + (1 - n^0(t))\{3(p_1 + p_2)\} \\ &\quad - (n^0(t) + 1)\{4(a_1^1 + a_2^1)\} \end{aligned} \quad (19.47)$$

or

$$\frac{dn^0(t)}{dt} = -110n^0(t) + 20 . \tag{19.48}$$

The response is plotted in Figure 19.7. In this case a reset signal will be sent to Layer 2, since $n^0(t)$ is positive. In this example, because the vigilance parameter is set to $\rho = 0.75$, and $\mathbf{p}$ has only two elements, we will have a reset whenever $\mathbf{p}$ and $\mathbf{a}^1$ are not identical. (If the vigilance parameter were set to $\rho = 0.25$, we would not have had a reset for the $\mathbf{p}$ and $\mathbf{a}^1$ of Eq. (19.46), since $\|\mathbf{a}^1\|^2 / \|\mathbf{p}\|^2 = 1/2$.)

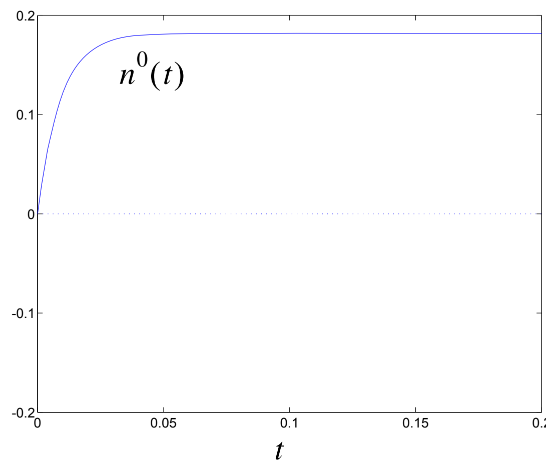


Figure 19.7 Response of the Orienting Subsystem

The steady state operation of the Orienting Subsystem can be summarized as follows:

$$a^0 = \begin{cases} 1 , & \text{if}[\|\mathbf{a}^1\|^2 / \|\mathbf{p}\|^2 < \rho] \\ 0 , & \text{otherwise} \end{cases} . \tag{19.49}$$

To experiment with the Orienting Subsystem, use the Neural Network Design Demonstration Orienting Subsystem (`nnd16os`).

Learning Law: L1-L2

The ART1 network has two separate learning laws: one for the L1-L2 connections, and another for the L2-L1 connections. The L1-L2 connections use a type of instar learning to learn to *recognize* a set of prototype patterns. The L2-L1 connections use outstar learning in order to *reproduce* (or recall) a set of prototype patterns. In this section we will describe the L1-

L2 instar learning law, and in the following section we will present the L2-L1 outstar learning law.

We should note that the L1-L2 connections and the L2-L1 connections are updated at the same time. Whenever the input pattern and the expectation have an adequate match, as determined by the Orienting Subsystem, both $\mathbf{W}^{1:2}$ and $\mathbf{W}^{2:1}$ are adapted. This process of matching, and subsequent adaptation, is referred to as *resonance*, hence the name adaptive resonance theory.

Resonance

Subset/Superset Dilemma

The learning in the L1-L2 connections of the ART1 network is very close to the learning in the Grossberg network of Chapter 18, with one major difference. In the Grossberg network, the input patterns are normalized in Layer 1, and therefore all of the prototype patterns will have the same length. In the ART1 network no normalization takes place in Layer 1. Therefore a problem can occur when one prototype pattern is a subset of another. For example, suppose that the L1-L2 connection matrix is

$$\mathbf{W}^{1:2} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \tag{19.50}$$

so that the prototype patterns are

$${}_1\mathbf{w}^{1:2} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \text{ and } {}_2\mathbf{w}^{1:2} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{19.51}$$

We say that ${}_1\mathbf{w}^{1:2}$ is a subset of ${}_2\mathbf{w}^{1:2}$, since ${}_2\mathbf{w}^{1:2}$ has a 1 wherever ${}_1\mathbf{w}^{1:2}$ has a 1.

If the output of Layer 1 is

$$\mathbf{a}^1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \tag{19.52}$$

then the input to Layer 2 will be

$$\mathbf{W}^{1:2}\mathbf{a}^1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}. \tag{19.53}$$

Both prototype vectors have the same inner product with $\mathbf{a}^1$, even though the first prototype is identical to $\mathbf{a}^1$ and the second prototype is not. This is called the subset/superset dilemma.

One solution to the subset/superset dilemma is to normalize the prototype patterns. That is, when a prototype pattern has a large number of nonzero entries, the magnitude of each entry should be reduced. For example, using our preceding problem, we could modify the L1-L2 matrix as follows:

$$\mathbf{W}^{1:2} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}. \tag{19.54}$$

The input to Layer 2 will then be

$$\mathbf{W}^{1:2}\mathbf{a}^1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{2}{3} \end{bmatrix}. \tag{19.55}$$

Now we have the desired result: the first prototype has the largest inner product with $\mathbf{a}^1$. The first neuron in Layer 2 will be activated.

In the Grossberg network of Chapter 18 we obtained normalized prototype patterns by normalizing the input patterns in Layer 1. In the ART1 network we will normalize the prototype patterns by using an on-center/off-surround competition in the L1-L2 learning law.

Learning Law

The learning law for $\mathbf{W}^{1:2}$ is

$$\frac{d[{}_i\mathbf{w}^{1:2}(t)]}{dt} = a_i^2(t)[\{{}^{+}\mathbf{b} - {}_i\mathbf{w}^{1:2}(t)\}\zeta[{}^{+}\mathbf{W}]\mathbf{a}^1(t) - \{{}_i\mathbf{w}^{1:2}(t) + {}^{-}\mathbf{b}\}[{}^{-}\mathbf{W}]\mathbf{a}^1(t)], \tag{19.56}$$

where

$${}^{+}\mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad {}^{-}\mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad {}^{+}\mathbf{W} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \text{ and } {}^{-}\mathbf{W} = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 0 \end{bmatrix}. \tag{19.57}$$

This is a modified form of instar learning. When neuron i of Layer 2 is active, the ith row of $\mathbf{W}^{1:2}$, ${}_i\mathbf{w}^{1:2}$, is moved in the direction of $\mathbf{a}^1$. The difference between Eq. (19.56) and the standard instar learning is that the elements of ${}_i\mathbf{w}^{1:2}$ compete, and therefore ${}_i\mathbf{w}^{1:2}$ is normalized. In the bracket on the right side of Eq. (19.56) we can see that it has the form of a shunting model, with on-center/off-surround input connections from $\mathbf{a}^1$. The excitatory bias is ${}^{+}\mathbf{b} = \mathbf{1}$ (a vector of 1's), and the inhibitory bias is ${}^{-}\mathbf{b} = \mathbf{0}$, which ensures that the elements of ${}_i\mathbf{w}^{1:2}$ remain between 0 and 1. (Recall our discussion of the shunting model in Chapter 18.)

Fast Learning

To verify that Eq. (19.56) causes normalization of the prototype patterns, let's investigate the steady state operation. For this analysis we will assume that the outputs of Layer 1 and Layer 2 remain constant until the weights reach steady state. This is called *fast learning*.

The equation for element $w^{1:2}_{i,j}$ is

$$\frac{dw^{1:2}_{i,j}(t)}{dt} = a^2_i(t)\left[(1 - w^{1:2}_{i,j}(t))\zeta a^1_j(t) - w^{1:2}_{i,j}(t)\sum_{k \neq j} a^1_k(t)\right]. \tag{19.58}$$

If we assume that neuron i is active in Layer 2 ($a^2_i(t) = 1$) and set the derivative to zero in Eq. (19.58), we see that

$$0 = \left[(1 - w^{1:2}_{i,j})\zeta a^1_j - w^{1:2}_{i,j}\sum_{k \neq j} a^1_k\right]. \tag{19.59}$$

To find the steady state value of $w^{1:2}_{i,j}$, we will consider two cases. First, assume that $a^1_j = 1$. Then we have

$$0 = (1 - w^{1:2}_{i,j})\zeta - w^{1:2}_{i,j}(\|\mathbf{a}^1\|^2 - 1) = -(\zeta + \|\mathbf{a}^1\|^2 - 1)w^{1:2}_{i,j} + \zeta, \tag{19.60}$$

or

$$w^{1:2}_{i,j} = \frac{\zeta}{\zeta + \|\mathbf{a}^1\|^2 - 1}. \tag{19.61}$$

(Note that $\sum_{k=1}^{S^1} a^1_k = \|\mathbf{a}^1\|^2$, since $\mathbf{a}^1$ is a binary vector.)

On the other hand, if $a^1_j = 0$, then Eq. (19.59) reduces to

$$0 = -w^{1:2}_{i,j}\|\mathbf{a}^1\|^2, \tag{19.62}$$

or

$$w_{i,j}^{1:2} = 0 . \tag{19.63}$$

To summarize Eq. (19.61) and Eq. (19.63):

$$_{i}\mathbf{w}^{1:2} = \frac{\zeta \mathbf{a}^{1}}{\zeta + \|\mathbf{a}^{1}\|^{2} - 1} , \tag{19.64}$$

where $\zeta > 1$ to ensure that the denominator will never equal zero.

Therefore the prototype patterns will be normalized, and this will solve the subset/superset dilemma. (By "normalized" here we do not mean that all prototype vectors will have unit length in Euclidean distance, but simply that the rows of $\mathbf{W}^{1:2}$ that have more nonzero entries will have elements with smaller magnitudes. In this case, vectors with more nonzero entries may actually have a smaller length than vectors with fewer nonzero entries.)

Learning Law: L2-L1

The L2-L1 connections, $\mathbf{W}^{2:1}$, in the ART1 architecture are trained using an outstar learning rule. The purpose of the L2-L1 connections is to recall an appropriate prototype pattern (the expectation), so that it can be compared and combined, in Layer 1, with the input pattern. When the expectation and the input pattern do not match, a reset is sent to Layer 2, so that a new prototype pattern can be chosen (as we have discussed in previous sections).

The learning law for $\mathbf{W}^{2:1}$ is a typical outstar equation:

$$\frac{d[\mathbf{w}_{j}^{2:1}(t)]}{dt} = a_{j}^{2}(t)[-\mathbf{w}_{j}^{2:1}(t) + \mathbf{a}^{1}(t)] . \tag{19.65}$$

Therefore, if neuron j in Layer 2 is active (has won the competition), then column j of $\mathbf{W}^{2:1}$ is moved toward the $\mathbf{a}^{1}$ pattern. To illustrate this, let's investigate the steady state operation of Eq. (19.65).

For this analysis we will assume the fast learning scenario, where the outputs of Layer 1 and Layer 2 remain constant until the weights reach steady state. Assume that neuron j in Layer 2 is active, so that $a_{j}^{2} = 1$. Setting the derivative in Eq. (19.65) to zero, we find

$$\mathbf{0} = -\mathbf{w}_{j}^{2:1} + \mathbf{a}^{1} , \text{ or } \mathbf{w}_{j}^{2:1} = \mathbf{a}^{1} . \tag{19.66}$$

Therefore column j of $\mathbf{W}^{2:1}$ converges to the output of Layer 1, $\mathbf{a}^{1}$. Recall from Eq. (19.20) and Eq. (19.21) that $\mathbf{a}^{1}$ is a combination of the input pattern and the appropriate prototype pattern. Therefore the prototype pattern is modified to incorporate the current input pattern (if there is a close enough match).

Keep in mind that $\mathbf{W}^{1:2}$ and $\mathbf{W}^{2:1}$ are updated at the same time. When neuron j of Layer 2 is active and there is a sufficient match between the expectation and the input pattern (which indicates a resonance condition), then row j of $\mathbf{W}^{1:2}$ and column j of $\mathbf{W}^{2:1}$ are adapted. In fast learning, column j of $\mathbf{W}^{2:1}$ is set to $\mathbf{a}^1$, while row j of $\mathbf{W}^{1:2}$ is set to a normalized version of $\mathbf{a}^1$.

ART1 Algorithm Summary

Now that we have investigated each of the subsystems of the ART1 architecture, we can gain some insight into its overall operation if we summarize the key steady state equations and organize them into an algorithm.

Initialization

The ART1 algorithm begins with an initialization of the weight matrices $\mathbf{W}^{1:2}$ and $\mathbf{W}^{2:1}$. The initial $\mathbf{W}^{2:1}$ matrix is set to all 1's. Thus, the first time a new neuron in Layer 2 wins a competition, resonance will occur, since $\mathbf{a}^1 = \mathbf{p} \cap \mathbf{w}_j^{2:1} = \mathbf{p}$ and therefore $\|\mathbf{a}^1\|^2 / \|\mathbf{p}\|^2 = 1 > \rho$. This means that any untrained column in $\mathbf{W}^{2:1}$ is effectively a blank slate and will cause a match with any input pattern.

Since the rows of the $\mathbf{W}^{1:2}$ matrix should be normalized versions of the columns of $\mathbf{W}^{2:1}$, every element of the initial $\mathbf{W}^{1:2}$ matrix is set to $\zeta / (\zeta + S^1 - 1)$.

Algorithm

After initialization, the ART1 algorithm proceeds as follows:

1. First, we present an input pattern to the network. Since Layer 2 is not active on initialization (i.e., each $a_j^2 = 0$), the output of Layer 1 is (Eq. (19.20))

$$\mathbf{a}^1 = \mathbf{p} . \tag{19.67}$$

2. Next, we compute the input to Layer 2,

$$\mathbf{W}^{1:2}\mathbf{a}^1 , \tag{19.68}$$

   and activate the neuron in Layer 2 with the largest input (Eq. (19.37)):

$$a_i^2 = \begin{cases} 1, & \text{if}(({}_i\mathbf{w}^{1:2})^T\mathbf{a}^1 = max[({}_k\mathbf{w}^{1:2})^T\mathbf{a}^1]) \\ 0, & \text{otherwise} \end{cases} . \tag{19.69}$$

In case of a tie, the neuron with the smallest index is declared the winner.

3. We then compute the L2-L1 expectation (where we assume neuron j of Layer 2 is activated):

$$\mathbf{W}^{2:1}\mathbf{a}^2 = \mathbf{w}_j^{2:1} . \tag{19.70}$$

4. Now that Layer 2 is active, we adjust the Layer 1 output to include the L2-L1 expectation (Eq. (19.21)):

$$\mathbf{a}^1 = \mathbf{p} \cap \mathbf{w}_j^{2:1} . \tag{19.71}$$

5. Next, the Orienting Subsystem determines the degree of match between the expectation and the input pattern (Eq. (19.49)):

$$a^0 = \begin{cases} 1, & \text{if}[\|\mathbf{a}^1\|^2 / \|\mathbf{p}\|^2 < \rho] \\ 0, & \text{otherwise} \end{cases} . \tag{19.72}$$

6. If $a^0 = 1$, then we set $a_j^2 = 0$, inhibit it until an adequate match occurs (resonance), and return to step 1. If $a^0 = 0$, we continue with step 7.

7. Resonance has occurred. Therefore we update row j of $\mathbf{W}^{1:2}$ (Eq. (19.61)):

$${}_j\mathbf{w}^{1:2} = \frac{\zeta\mathbf{a}^1}{\zeta + \|\mathbf{a}^1\|^2 - 1} . \tag{19.73}$$

8. We now update column j of $\mathbf{W}^{2:1}$ (Eq. (19.66)):

$$\mathbf{w}_j^{2:1} = \mathbf{a}^1 . \tag{19.74}$$

9. We remove the input pattern, restore all inhibited neurons in Layer 2, and return to step 1 with a new input pattern.

The input patterns continue to be applied to the network until the weights stabilize (do not change). Carpenter and Grossberg have shown [CaGr87a] that the ART1 algorithm will always form stable clusters for any set of input patterns.

See Problems P19.5, P19.6 and P19.7 for detailed examples of the ART1 algorithm.

To experiment with the ART1 algorithm, use the Neural Network Design Demonstration ART1 (`nnd16a1`).

Other ART Architectures

The ART1 network is just one example of adaptive resonance theory. Carpenter and Grossberg, and others in their research group, have developed many variations on this theme.

One disadvantage of the ART1 network is that it can only be used for binary input patterns. Carpenter and Grossberg developed a variation of ART1, called ART2, to handle either analog or binary patterns [CaGr87b]. The basic structure of ART2 is very similar to ART1, with the exception of Layer 1. In ART2 several sublayers take the place of Layer 1. These sublayers are needed because analog vectors, unlike binary vectors, can be arbitrarily close together. The sublayers perform a combination of normalization and noise suppression, in addition to the comparison of the input vector and the expectation that is needed by the orienting subsystem.

Carpenter and Grossberg later developed the ART3 network [CaGr90], which introduced a more sophisticated biological model for the reset mechanism required for ART. Up to the present time, this network has not been widely applied.

In 1991 Carpenter, Grossberg and Reynolds introduced the ARTMAP network [CaGrRe91]. In contrast with all of the previous ART networks, it is a supervised network. The ARTMAP architecture consists of two ART modules that are connected by an "inter-ART" associative memory. One ART module receives the input vector, while the other ART module receives the desired output vector. The network learns to predict the correct output vector whenever the input vector is presented.

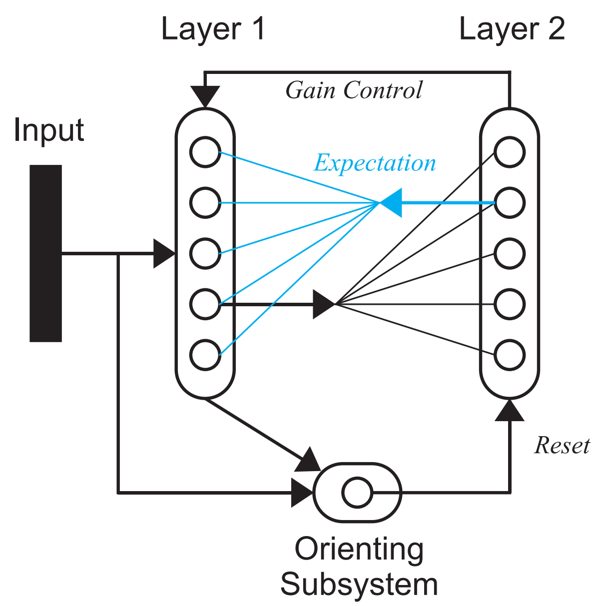
More recently, Carpenter, Grossberg, Markuzon, Reynolds and Rosen have modified the ARTMAP architecture to incorporate fuzzy logic. The result is referred to as Fuzzy ARTMAP [CaGrMa92]. It seems to improve performance, especially with noisy input patterns.

All of these ART architectures incorporate the key modules discussed in this chapter, including:

- L1-L2 instars for pattern recognition.
- L2-L1 outstars for pattern recall.
- Layer 2 for contrast enhancement (competition).
- Layer 1 for comparison of input and expectation.
- Orienting Subsystem for resetting when a pattern mismatch occurs.

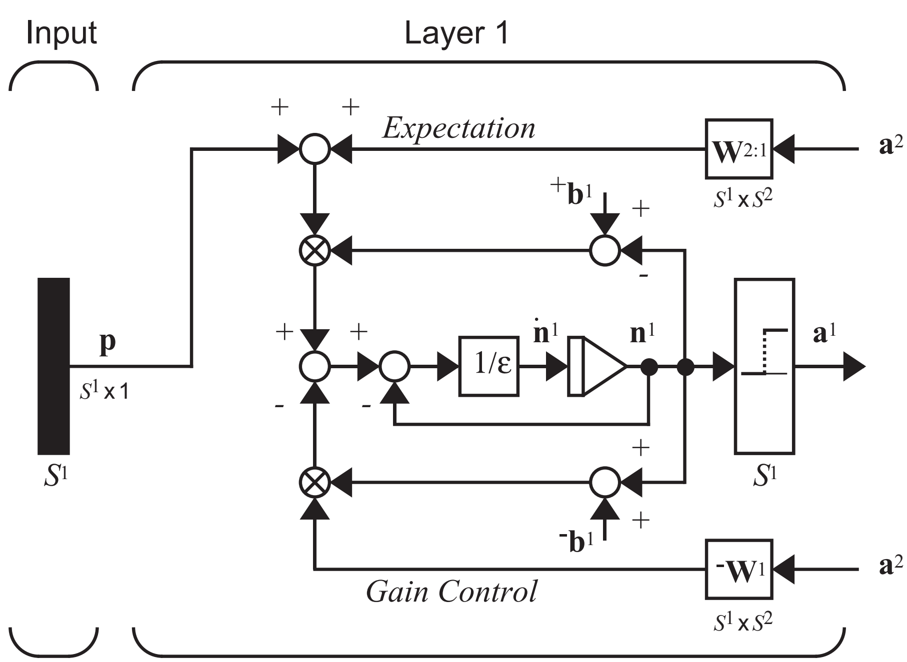
Summary of Results

Basic ART Architecture



ART1 Network (Binary Patterns)

ART1 Layer 1



$$\varepsilon \, d\mathbf{n}^1/dt = -\mathbf{n}^1 + ({}^{+}\mathbf{b}^1 - \mathbf{n}^1)\{\mathbf{p} + \mathbf{W}^{2:1}\mathbf{a}^2\} - (\mathbf{n}^1 + {}^{-}\mathbf{b}^1)[{}^{-}\mathbf{W}^1]\mathbf{a}^2$$

Layer 1 Equation

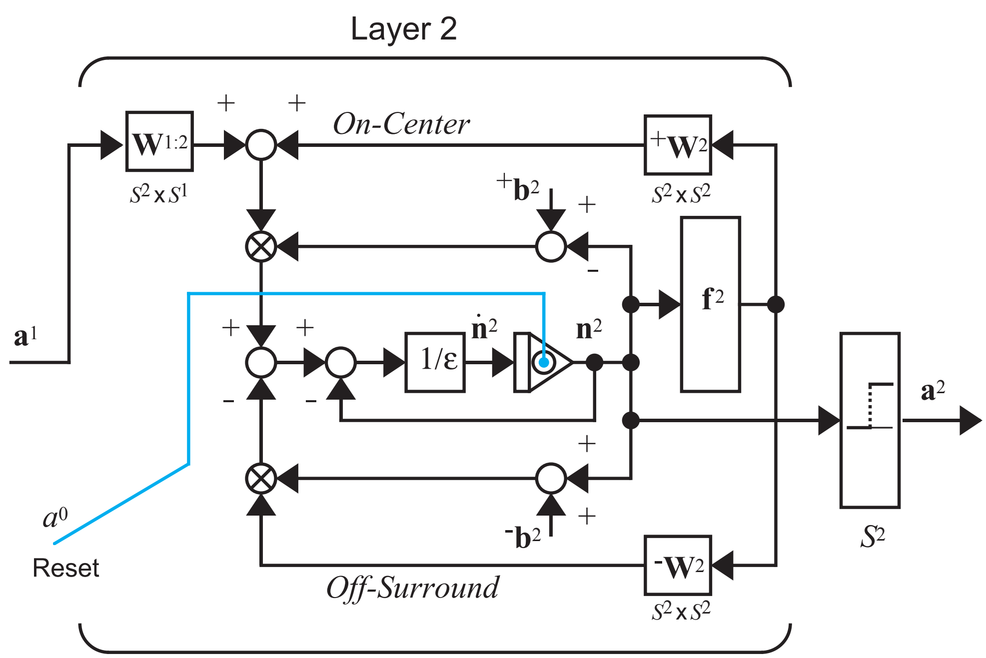
$$\varepsilon\frac{d\mathbf{n}^1(t)}{dt} = -\mathbf{n}^1(t) + ({}^{+}\mathbf{b}^1 - \mathbf{n}^1(t))\{\mathbf{p} + \mathbf{W}^{2:1}\mathbf{a}^2(t)\} - (\mathbf{n}^1(t) + {}^{-}\mathbf{b}^1)[{}^{-}\mathbf{W}^1]\mathbf{a}^2(t)$$

Steady State Operation

If Layer 2 is not active (i.e., each $a_j^2 = 0$), $\mathbf{a}^1 = \mathbf{p}$.

If Layer 2 is active (i.e., one $a_j^2 = 1$), $\mathbf{a}^1 = \mathbf{p} \cap {}_j\mathbf{w}^{2:1}$.

ART1 Layer 2



$$\varepsilon\, d\mathbf{n}^2/dt = -\mathbf{n}^2 + ({}^{+}\mathbf{b}^2 - \mathbf{n}^2)\{[{}^{+}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2) + \mathbf{W}^{1:2}\mathbf{a}^1\}$$
$$- (\mathbf{n}^2 + {}^{-}\mathbf{b}^2)[{}^{-}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2)$$

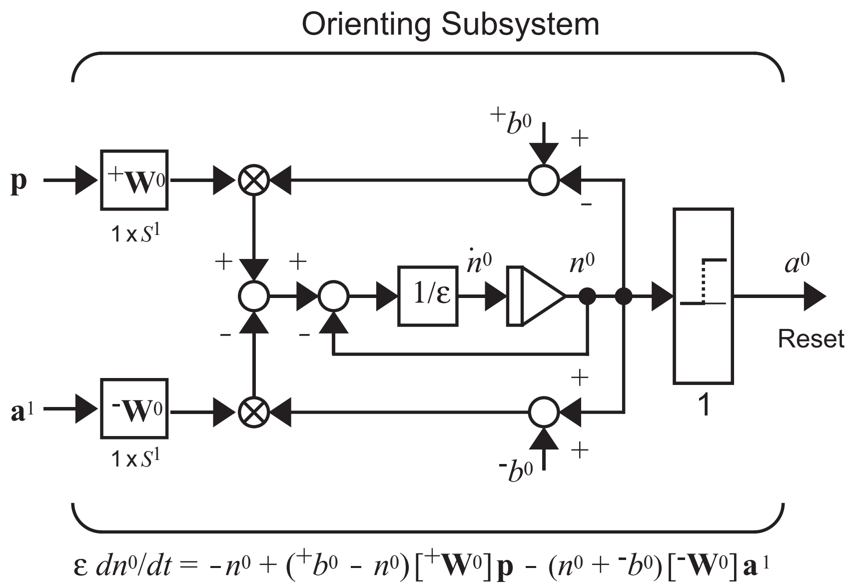
Layer 2 Equation

$$\varepsilon\frac{d\mathbf{n}^2(t)}{dt} = -\mathbf{n}^2(t) + ({}^{+}\mathbf{b}^2 - \mathbf{n}^2(t))\{[{}^{+}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t)) + \mathbf{W}^{1:2}\mathbf{a}^1\} - (\mathbf{n}^2(t) + {}^{-}\mathbf{b}^2)[{}^{-}\mathbf{W}^2]\mathbf{f}^2(\mathbf{n}^2(t))$$

Steady State Operation

$$a_i^2 = \begin{cases} 1, & \text{if}(({}_i\mathbf{w}^{1:2})^T\mathbf{a}^1 = max[({}_j\mathbf{w}^{1:2})^T\mathbf{a}^1]) \\ 0, & \text{otherwise} \end{cases}$$

Orienting Subsystem



Orienting Subsystem Equation

$$\varepsilon\frac{dn^0(t)}{dt} = -n^0(t) + ({}^+b^0 - n^0(t))\{{}^+\mathbf{W}^0\mathbf{p}\} - (n^0(t) + {}^-b^0)\{{}^-\mathbf{W}^0\mathbf{a}^1\}$$

$$\text{where } {}^+\mathbf{W}^0 = \begin{bmatrix}\alpha & \alpha & \dots & \alpha\end{bmatrix},\ {}^-\mathbf{W}^0 = \begin{bmatrix}\beta & \beta & \dots & \beta\end{bmatrix},\ {}^+b^0 = {}^-b^0 = 1$$

Steady State Operation

$$a^0 = \begin{cases} 1, & \text{if}\left[\|\mathbf{a}^1\|^2/\|\mathbf{p}\|^2 < \rho\right] \\ 0, & \text{otherwise} \end{cases}$$

L1-L2 Learning Law

$$\frac{d[{}_i\mathbf{w}^{1:2}(t)]}{dt} = a_i^2(t)[\{{}^+\mathbf{b} - {}_i\mathbf{w}^{1:2}(t)\}\zeta[{}^+\mathbf{W}]\mathbf{a}^1(t) - \{{}_i\mathbf{w}^{1:2}(t) + {}^-\mathbf{b}\}[{}^-\mathbf{W}]\mathbf{a}^1(t)]$$

$${}^+\mathbf{b} = \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} \quad {}^-\mathbf{b} = \begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix} \quad {}^+\mathbf{W} = \begin{bmatrix}1 & 0 & \cdots & 0\\0 & 1 & \cdots & 0\\\vdots & \vdots & & \vdots\\0 & 0 & \cdots & 1\end{bmatrix} \quad {}^-\mathbf{W} = \begin{bmatrix}0 & 1 & \cdots & 1\\1 & 0 & \cdots & 1\\\vdots & \vdots & & \vdots\\1 & 1 & \cdots & 0\end{bmatrix}$$

Steady State Operation (Fast Learning)

$$ {}_i\mathbf{w}^{1:2} = \frac{\zeta \mathbf{a}^1}{\zeta + \|\mathbf{a}^1\|^2 - 1} \quad \text{(Neuron } i \text{ in Layer 2 Active)} $$

L2-L1 Learning Law

$$ \frac{d[\mathbf{w}_j^{2:1}(t)]}{dt} = a_j^2(t)[-\mathbf{w}_j^{2:1}(t) + \mathbf{a}^1(t)] $$

Steady State Operation (Fast Learning)

$$ \mathbf{w}_j^{2:1} = \mathbf{a}^1 \quad \text{(Neuron } j \text{ in Layer 2 Active)} $$

ART1 Algorithm (Fast Learning) Summary

Initialization

The initial $\mathbf{W}^{2:1}$ matrix is set to all 1's.

Every element of the initial $\mathbf{W}^{1:2}$ matrix is set to $\zeta/(\zeta + S^1 - 1)$.

Algorithm

1. First, we present an input pattern to the network. Since Layer 2 is not active on initialization (i.e., each $a_j^2 = 0$), the output of Layer 1 is

$$ \mathbf{a}^1 = \mathbf{p} . $$

2. Next, we compute the input to Layer 2,

$$ \mathbf{W}^{1:2}\mathbf{a}^1 , $$

and activate the neuron in Layer 2 with the largest input:

$$ a_i^2 = \begin{cases} 1 , & \text{if}(({}_i\mathbf{w}^{1:2})^T\mathbf{a}^1 = max[({}_k\mathbf{w}^{1:2})^T\mathbf{a}^1]) \\ 0 , & \text{otherwise} \end{cases} . $$

In case of a tie, the neuron with the smallest index is declared the winner.

3. We then compute the L2-L1 expectation (where we assume neuron j of Layer 2 is activated):

$$ \mathbf{W}^{2:1}\mathbf{a}^2 = \mathbf{w}_j^{2:1} . $$

4. Now that Layer 2 is active, we adjust the Layer 1 output to include the L2-L1 expectation:

$$\mathbf{a}^1 = \mathbf{p} \cap \mathbf{w}_j^{2:1} .$$

5. Next, the Orienting Subsystem determines the degree of match between the expectation and the input pattern:

$$a^0 = \begin{cases} 1 , & \text{if}[\|\mathbf{a}^1\|^2 / \|\mathbf{p}\|^2 < \rho] \\ 0 , & \text{otherwise} \end{cases} .$$

6. If $a^0 = 1$, then we set $a_j^2 = 0$, inhibit it until an adequate match occurs (resonance), and return to step 1. If $a^0 = 0$, we continue with step 7.

7. Resonance has occurred, therefore we update row j of $\mathbf{W}^{1:2}$:

$${}_j\mathbf{w}^{1:2} = \frac{\zeta \mathbf{a}^1}{\zeta + \|\mathbf{a}^1\|^2 - 1} .$$

8. We now update column j of $\mathbf{W}^{2:1}$:

$$\mathbf{w}_j^{2:1} = \mathbf{a}^1 .$$

9. We remove the input pattern, restore all inhibited neurons in Layer 2, and return to step 1 with a new input pattern.

Solved Problems

P19.1 Consider Layer 1 of the ART1 network with the following parameters:

$$\varepsilon = 0.01 \quad {}^{+}b^{1} = 2 \quad {}^{-}b^{1} = 3 .$$

Assume two neurons in Layer 2, two elements in the input vector and the following weight matrix and input:

$$\mathbf{W}^{2:1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} .$$

Also, assume that neuron 1 of Layer 2 is active.

i. Find and plot the response $\mathbf{n}^1$.

ii. Check to see that the answer to part (i) satisfies the steady state response predicted by Eq. (19.21).

i. Since Layer 2 is active, and neuron 1 of Layer 2 wins the competition, the equations of operation of Layer 1 are

$$(0.01)\frac{dn_1^1}{dt} = -n_1^1 + (2 - n_1^1)\{p_1 + w_{1,1}^{2:1}\} - (n_1^1 + 3)$$

$$= -n_1^1 + (2 - n_1^1)\{1 + 0\} - (n_1^1 + 3) = -3n_1^1 - 1$$

$$(0.01)\frac{dn_2^1}{dt} = -n_2^1 + (2 - n_2^1)\{p_2 + w_{2,1}^{2:1}\} - (n_2^1 + 3)$$

$$= -n_2^1 + (2 - n_2^1)\{1 + 1\} - (n_2^1 + 3) = -4n_2^1 + 1 \quad .$$

These can be simplified to obtain

$$\frac{dn_1^1}{dt} = -300n_1^1 - 100 ,$$

$$\frac{dn_2^1}{dt} = -400n_2^1 + 100 .$$

If we assume that both neurons start with zero initial condition, the solutions are

$$n_1^1(t) = -\frac{1}{3}[1 - e^{-300t}],$$

$$n_2^1(t) = \frac{1}{4}[1 - e^{-400t}].$$

These are displayed in Figure P19.1.

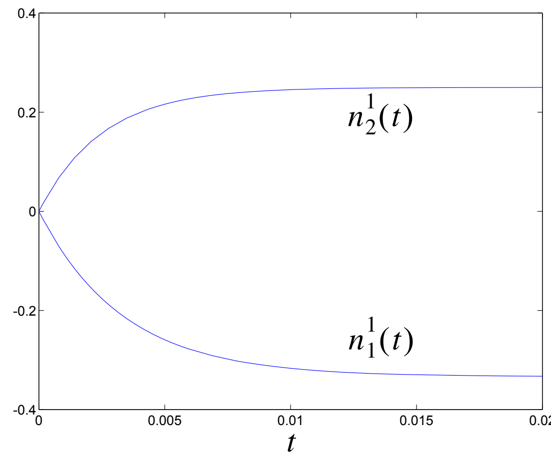


Figure P19.1 Response of Layer 1

ii. Note that $n_1^1(t)$ converges to a negative value, and $n_2^1(t)$ converges to a positive value. Therefore, $a_1^1(t)$ converges to 0, and $a_2^1(t)$ converges to 1 (recall that the transfer function for Layer 1 is $hardlim^+$). This agrees with our steady state analysis (see Eq. (19.21)), since

$$\mathbf{p} \cap \mathbf{w}_1^{2:1} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \cap \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \mathbf{a}^1 . \qquad (19.75)$$

P19.2 Consider Layer 2 of the ART1 network with the following parameters:

$$\varepsilon = 0.1 \quad {}^{+}\mathbf{b}^2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad {}^{-}\mathbf{b}^2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad \mathbf{W}^{1:2} = \begin{bmatrix} ({}_1\mathbf{w}^{1:2})^T \\ ({}_2\mathbf{w}^{1:2})^T \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 1 & 0 \end{bmatrix}$$

and

$$f^2(n) = \begin{cases} 10(n)^2 & n \ge 0 \\ 0 & n < 0 \end{cases}.$$

Assume that the output of Layer 1 is

$$\mathbf{a}^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

This is equivalent to the Layer 2 example in the text (page 19-12), with the exception of the bias values.

i. Write the equations of operation of Layer 2 and simulate and plot the response. Explain the effect of increasing the bias values.

ii. Verify that the steady state operation of Layer 2 is correct.

i. The equations of operation of the layer will be

$$(0.1)\frac{dn_1^2(t)}{dt} = -n_1^2(t) + (2 - n_1^2(t))\{f^2(n_1^2(t)) + ({}_1\mathbf{w}^{1:2})^T\mathbf{a}^1\}$$

$$-(n_1^2(t) + 2)f^2(n_2^2(t)),$$

$$(0.1)\frac{dn_2^2(t)}{dt} = -n_2^2(t) + (2 - n_2^2(t))\{f^2(n_2^2(t)) + ({}_2\mathbf{w}^{1:2})^T\mathbf{a}^1\}$$

$$-(n_2^2(t) + 2)f^2(n_1^2(t)).$$

Figure P19.2 illustrates the response of Layer 2 when the input vector is $\mathbf{a}^1 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$. The second row of $\mathbf{W}^{1:2}$ has a larger inner product with $\mathbf{a}^1$ than the first row, therefore neuron 2 wins the competition.

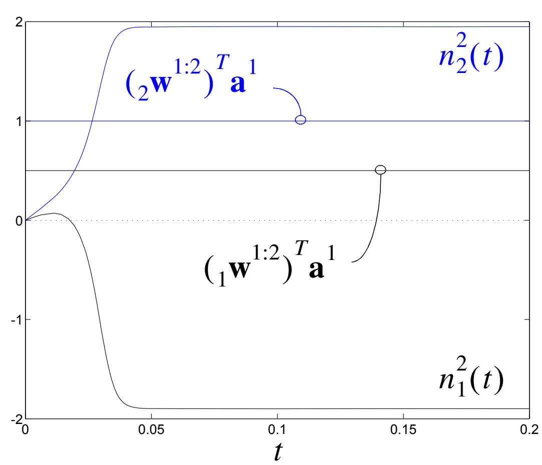


Figure P19.2 Response of Layer 2

If we compare Figure P19.2 with Figure 19.5, we can see that the bias value has three effects. First, the speed of response is increased; the neuron outputs move more quickly to their steady state values. Second, the range of the response is increased from $[-1, 1]$ to $[-2, 2]$. (Recall from Chapter 18 that for the shunting model the upper limit will be the excitatory bias ${}^{+}b$. The lower limit will be the inhibitory bias ${}^{-}b$.) Third, the neuron responses move closer to the upper and lower limits.

ii. At steady state, $n_1^2(t)$ has a positive value, and $n_2^2(t)$ has a negative value. The steady state Layer 2 output will then be

$$\mathbf{a}^2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

This agrees with the desired steady state response characteristics of Layer 2:

$$a_i^2 = \begin{cases} 1, & \text{if}\left(({}_i\mathbf{w}^{1:2})^T\mathbf{a}^1 = max[({}_j\mathbf{w}^{1:2})^T\mathbf{a}^1]\right) \\ 0, & \text{otherwise} \end{cases}.$$

P19.3 Consider the Orienting Subsystem of the ART1 network with the following parameters:

$$\varepsilon = 0.1 \quad \alpha = 0.5 \quad \beta = 2 \quad (\rho = 0.25) \quad {}^{+}b^0 = {}^{-}b^0 = 0.5.$$

The inputs to the Orienting Subsystem are

$$\mathbf{p} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{a}^1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

i. Find and plot the response of the Orienting Subsystem $n^0(t)$.

ii. Verify that the steady state conditions are satisfied.

The equation of operation of the Orienting Subsystem is

$$(0.1)\frac{dn^0(t)}{dt} = -n^0(t) + (0.5 - n^0(t))\{0.5(p_1 + p_2 + p_3)\}$$
$$- (n^0(t) + 0.5)\{2(a_1^1 + a_2^1 + a_3^1)\}$$

or

$$\frac{dn^0(t)}{dt} = -65n^0(t) - 12.5 .$$

The response is then

$$n^0(t) = -0.1923[1 - e^{-65t}]$$

This response is plotted in Figure P19.3. In this case, since $n^0(t)$ is negative, $a^0 = hardlim^+(n^0) = 0$, and therefore a reset signal will not be sent to Layer 2.

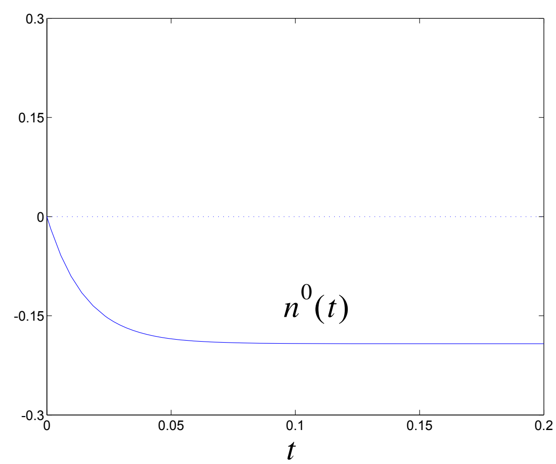


Figure P19.3 Response of the Orienting Subsystem

ii. The steady state operation of the Orienting Subsystem can be summarized as follows:

$$a^0 = \begin{cases} 1 , & \text{if}[\|\mathbf{a}^1\|^2 / \|\mathbf{p}\|^2 < \rho] \\ 0 , & \text{otherwise} \end{cases} .$$

For this problem

$$\|\mathbf{a}^1\|^2 / \|\mathbf{p}\|^2 = \left\| \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \right\|^2 \Big/ \left\| \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\|^2 = \frac{2}{3} > \rho = 0.25 .$$

Therefore $a^0 = 0$, which agrees with the results of part (i).

P19.4 Show that the learning equation for the L2-L1 connections is equivalent to the outstar equation described in Chapter 15.

The L2-L1 learning law (Eq. (19.65)) is

$$\frac{d[\mathbf{w}_j^{2:1}(t)]}{dt} = a_j^2(t)[-\mathbf{w}_j^{2:1}(t) + \mathbf{a}^1(t)].$$

If we approximate the derivative by

$$\frac{d[\mathbf{w}_j^{2:1}(t)]}{dt} \approx \frac{\mathbf{w}_j^{2:1}(t+\Delta t) - \mathbf{w}_j^{2:1}(t)}{\Delta t},$$

then we can rewrite Eq. (19.65) as

$$\mathbf{w}_j^{2:1}(t+\Delta t) = \mathbf{w}_j^{2:1}(t) + (\Delta t)a_j^2(t)\{-\mathbf{w}_j^{2:1}(t) + \mathbf{a}^1(t)\}.$$

This is equivalent to the outstar rule of Chapter 15 (Eq. (15.51)). Here the input to the L2-L1 connections is $a_j^2(t)$, and the output of the L2-L1 connections is $\mathbf{a}^1$.

P19.5 Train an ART1 network using the following input vectors:

$$\mathbf{p}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{p}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{p}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

Use the parameters $\zeta = 2$, and $\rho = 0.4$, and choose $S^2 = 3$ (3 categories).

Our initial weights will be

$$\mathbf{W}^{2:1} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \mathbf{W}^{1:2} = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{bmatrix}.$$

We now begin the algorithm.

1. Compute the Layer 1 response:

$$\mathbf{a}^1 = \mathbf{p}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

2. Next, compute the input to Layer 2:

$$\mathbf{W}^{1:2}\mathbf{a}^1 = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}.$$

Since all neurons have the same input, pick the first neuron as the winner. (In case of a tie, pick the neuron with the smallest index.)

$$\mathbf{a}^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

3. Now compute the L2-L1 expectation:

$$\mathbf{W}^{2:1}\mathbf{a}^2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \mathbf{w}_1^{2:1}.$$

4. Adjust the Layer 1 output to include the L2-L1 expectation:

$$\mathbf{a}^1 = \mathbf{p}_1 \cap \mathbf{w}_1^{2:1} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \cap \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

5. Next, the Orienting Subsystem determines the degree of match between the expectation and the input pattern:

$$\|\mathbf{a}^1\|^2 / \|\mathbf{p}_1\|^2 = \frac{1}{1} > \rho = 0.4\text{, therefore } a^0 = 0 \text{ (no reset).}$$

6. Since $a^0 = 0$, continue with step 7.

7. Resonance has occurred, therefore update row 1 of $\mathbf{W}^{1:2}$:

$$_1\mathbf{w}^{1:2} = \frac{2\mathbf{a}^1}{2 + \|\mathbf{a}^1\|^2 - 1} = \mathbf{a}^1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},\ \mathbf{W}^{1:2} = \begin{bmatrix} 0 & 1 & 0 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{bmatrix}.$$

8. Update column 1 of $\mathbf{W}^{2:1}$:

$$\mathbf{w}_1^{2:1} = \mathbf{a}^1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{W}^{2:1} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

9. Remove $\mathbf{p}_1$, and return to step 1 with input pattern $\mathbf{p}_2$.
1. Compute the new Layer 1 response (Layer 2 inactive):

$$\mathbf{a}^1 = \mathbf{p}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

2. Next, compute the input to Layer 2:

$$\mathbf{W}^{1:2}\mathbf{a}^1 = \begin{bmatrix} 0 & 1 & 0 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix}.$$

Since neurons 2 and 3 have the same input, pick the second neuron as the winner:

$$\mathbf{a}^2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{19.76}$$

3. Now compute the L2-L1 expectation:

$$\mathbf{W}^{2:1}\mathbf{a}^2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \mathbf{w}_2^{2:1} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

4. Adjust the Layer 1 output to include the L2-L1 expectation:

$$\mathbf{a}^1 = \mathbf{p}_2 \cap \mathbf{w}_2^{2:1} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \cap \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

5. Next, the Orienting Subsystem determines the degree of match between the expectation and the input pattern:

$$\|\mathbf{a}^1\|^2 / \|\mathbf{p}_2\|^2 = \frac{1}{1} > \rho = 0.4 \text{, therefore } a^0 = 0 \text{ (no reset).}$$

6. Since $a^0 = 0$, continue with step 7.

7. Resonance has occurred, therefore update row 2 of $\mathbf{W}^{1:2}$:

$$ {}_2\mathbf{w}^{1:2} = \frac{2\mathbf{a}^1}{2 + \|\mathbf{a}^1\|^2 - 1} = \mathbf{a}^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{W}^{1:2} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0.5 & 0.5 & 0.5 \end{bmatrix}. $$

8. Update column 2 of $\mathbf{W}^{2:1}$:

$$ \mathbf{w}_2^{2:1} = \mathbf{a}^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{W}^{2:1} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}. $$

9. Remove $\mathbf{p}_2$, and return to step 1 with input pattern $\mathbf{p}_3$.

1. Compute the Layer 1 response with the new input vector:

$$ \mathbf{a}^1 = \mathbf{p}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}. $$

2. Next, compute the input to Layer 2:

$$ \mathbf{W}^{1:2}\mathbf{a}^1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. $$

Since all neurons have the same input, pick the first neuron as the winner:

$$ \mathbf{a}^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. $$

3. Now compute the L2-L1 expectation:

$$ \mathbf{W}^{2:1}\mathbf{a}^2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{w}_1^{2:1} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. $$

4. Adjust the Layer 1 output to include the L2-L1 expectation:

$$\mathbf{a}^1 = \mathbf{p}_3 \cap \mathbf{w}_1^{2:1} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \cap \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

5. Next, the Orienting Subsystem determines the degree of match between the expectation and the input pattern:

$$\|\mathbf{a}^1\|^2 / \|\mathbf{p}_3\|^2 = \frac{1}{2} > \rho = 0.4 \text{, therefore } a^0 = 0 \text{ (no reset).}$$

6. Since $a^0 = 0$, continue with step 7.

7. Resonance has occurred, therefore update row 1 of $\mathbf{W}^{1:2}$:

$${}_1\mathbf{w}^{1:2} = \frac{2\mathbf{a}^1}{2 + \|\mathbf{a}^1\|^2 - 1} = \mathbf{a}^1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ \mathbf{W}^{1:2} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0.5 & 0.5 & 0.5 \end{bmatrix}.$$

8. Update column 1 of $\mathbf{W}^{2:1}$:

$$\mathbf{w}_2^{2:1} = \mathbf{a}^1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ \mathbf{W}^{2:1} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

This completes the training, since if you apply any of the three patterns again they will not change the weights. These patterns have been successfully clustered. This type of result (stable learning) is guaranteed for the ART1 algorithm, since it has been proven to always produce stable clusters.

P19.6 Repeat Problem P19.5, but change the vigilance parameter to $\rho = 0.6$.

The training will proceed exactly as in Problem P19.5, until pattern $\mathbf{p}_3$ is presented, so let's pick up the algorithm at that point.

1. Compute the Layer 1 response:

$$\mathbf{a}^1 = \mathbf{p}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

2. Next, compute the input to Layer 2:

$$\mathbf{W}^{1:2}\mathbf{a}^1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0.5 & 0.5 & 0.5 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Since all neurons have the same input, pick the first neuron as the winner:

$$\mathbf{a}^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

3. Now compute the L2-L1 expectation:

$$\mathbf{W}^{2:1}\mathbf{a}^2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{w}_1^{2:1} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

4. Adjust the Layer 1 output to include the L2-L1 expectation:

$$\mathbf{a}^1 = \mathbf{p}_3 \cap \mathbf{w}_1^{2:1} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \cap \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

5. Next, the Orienting Subsystem determines the degree of match between the expectation and the input pattern:

$$\|\mathbf{a}^1\|^2 / \|\mathbf{p}_3\|^2 = \frac{1}{2} < \rho = 0.6 \text{, therefore } a^0 = 1 \text{ (reset).}$$

6. Since $a^0 = 1$, set $a_1^2 = 0$, inhibit it until an adequate match occurs (resonance), and return to step 1.

1. Recompute the Layer 1 response (Layer 2 inactive):

$$\mathbf{a}^1 = \mathbf{p}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

2. Next, compute the input to Layer 2:

$$\mathbf{W}^{1:2}\mathbf{a}^{1} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Since neuron 1 is inhibited, neuron 2 is the winner:

$$\mathbf{a}^{2} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

3. Now compute the L2-L1 expectation:

$$\mathbf{W}^{2:1}\mathbf{a}^{2} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \mathbf{w}_2^{2:1} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

4. Adjust the Layer 1 output to include the L2-L1 expectation:

$$\mathbf{a}^{1} = \mathbf{p}_3 \cap \mathbf{w}_2^{2:1} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \cap \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

5. Next, the Orienting Subsystem determines the degree of match between the expectation and the input pattern:

$$\|\mathbf{a}^{1}\|^{2} / \|\mathbf{p}_3\|^{2} = \frac{1}{2} < \rho = 0.6, \text{ therefore } a^{0} = 1 \text{ (reset).}$$

6. Since $a^{0} = 1$, set $a_2^{2} = 0$, inhibit it until an adequate match occurs (resonance), and return to step 1.

1. Recompute the Layer 1 response:

$$\mathbf{a}^{1} = \mathbf{p}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

2. Next, compute the input to Layer 2:

$$\mathbf{W}^{1:2}\mathbf{a}^{1} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Since neurons 1 and 2 are inhibited, neuron 3 is the winner:

$$\mathbf{a}^2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

3. Now compute the L2-L1 expectation:

$$\mathbf{W}^{2:1}\mathbf{a}^2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{w}_3^{2:1} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

4. Adjust the Layer 1 output to include the L2-L1 expectation:

$$\mathbf{a}^1 = \mathbf{p}_3 \cap \mathbf{w}_3^{2:1} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \cap \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

5. Next, the Orienting Subsystem determines the degree of match between the expectation and the input pattern:

$$\|\mathbf{a}^1\|^2 / \|\mathbf{p}_3\|^2 = \frac{2}{2} > \rho = 0.6 \text{, therefore } a^0 = 0 \text{ (no reset).}$$

6. Since $a^0 = 0$, continue with step 7.
7. Resonance has occurred, therefore update row 3 of $\mathbf{W}^{1:2}$:

$${}_3\mathbf{w}^{1:2} = \frac{2\mathbf{a}^1}{2 + \|\mathbf{a}^1\|^2 - 1} = \frac{2}{3}\mathbf{a}^1 = \begin{bmatrix} \frac{2}{3} \\ \frac{2}{3} \\ 0 \end{bmatrix}, \mathbf{W}^{1:2} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ \frac{2}{3} & \frac{2}{3} & 0 \end{bmatrix}.$$

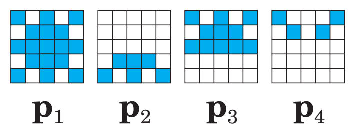
8. Update column 3 of $\mathbf{W}^{2:1}$:

$$\mathbf{w}_3^{2:1} = \mathbf{a}^1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \mathbf{W}^{2:1} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

This completes the training, since if you apply any of the three patterns again they will not change the weights. (Verify this for yourself by applying each input pattern to the network.) These patterns have been successfully clustered.

Note that in Problem P19.5, where the vigilance was $\rho = 0.4$, the patterns were clustered into two categories. In this problem, with vigilance $\rho = 0.6$, the patterns were clustered into three categories. The closer the vigilance is to 1, the more categories will be used. This is because an input pattern must be closer to a prototype in order to be incorporated into that prototype. When the vigilance is close to zero, many different input patterns can be incorporated into one prototype. The vigilance parameter adjusts the coarseness of the categorization.

P19.7 Train an ART1 network using the following input vectors (see [CaGr87a]):



Present the vectors in the order $\mathbf{p}_1 - \mathbf{p}_2 - \mathbf{p}_3 - \mathbf{p}_1 - \mathbf{p}_4$ (i.e., $\mathbf{p}_1$ is presented twice in each epoch). Use the parameters $\zeta = 2$ and $\rho = 0.6$, and choose $S^2 = 3$ (3 categories). Train the network until the weights have converged.

We begin by initializing the weight matrices. The initial $\mathbf{W}^{2:1}$ matrix is an $S^1 \times S^2 = 25 \times 3$ matrix of 1's. The initial $\mathbf{W}^{1:2}$ matrix is normalized, therefore it is an $S^2 \times S^1 = 3 \times 25$ matrix, with each element equal to

$$\frac{\zeta}{(\zeta + S^1 - 1)} = \frac{2}{(2 + 25 - 1)} = 0.0769 .$$

To create the input vectors we will scan each pattern row-by-row, where each blue square will be represented by a 1 and each white square will be represented by a 0. Since the input patterns are 5×5 grids, this will create 25-dimensional input vectors.

We now begin the training. Since it is not practical to display all of the calculations when the vectors are so large, we have summarized the results of the algorithm in Figure P19.4. Each row represents one iteration of the ART1 algorithm (presentation of one input vector). The left-most pattern in each row is the input vector. The remainder of the patterns represent the three columns of the $\mathbf{W}^{2:1}$ matrix. At each iteration, a star indicates the resonance point — the column of $\mathbf{W}^{2:1}$ that matched with the input pattern. Whenever a reset occurred, it is represented by a check mark. When more than one reset occurred in a given iteration, the number beside the check mark indicates the order in which the reset occurred.

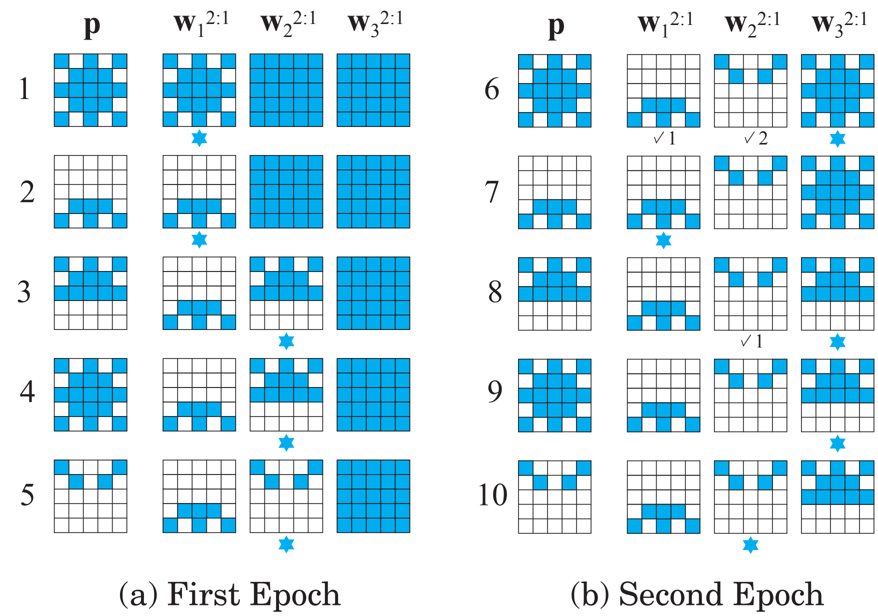


Figure P19.4 ART1 Iterations for Problem

A total of 10 iterations of the algorithm were performed (two epochs of the sequence $\mathbf{p}_1 - \mathbf{p}_2 - \mathbf{p}_3 - \mathbf{p}_1 - \mathbf{p}_4$). The weights are now stable. (You may want to check this by presenting each input pattern.)

There are several interesting points to notice in this example. First, notice that at iteration 4 both $\mathbf{p}_1$ and $\mathbf{p}_3$ are coded by $\mathbf{w}_2^{2:1}$. However, on iteration 5, when $\mathbf{p}_4$ is presented, $\mathbf{w}_2^{2:1}$ is modified to include $\mathbf{p}_4$. This new $\mathbf{w}_2^{2:1}$ no longer provides an adequate match with $\mathbf{p}_1$ and $\mathbf{p}_3$, as we can see at iterations 6 and 8. This requires them to take over neuron 3, which was unused during the first epoch.

The results of the algorithm could be modified by changing the vigilance parameter. How small would you have to make the vigilance, so that only two neurons in Layer 2 would be required to code all 4 input vectors? How large would the vigilance have to be before a fourth Layer 2 neuron was needed?

Epilogue

Competitive learning, and many other types of neural network training algorithms, suffer from a problem called the stability/plasticity dilemma. If a learning algorithm is sensitive to new inputs (plastic), then it runs the risk of forgetting prior learning (unstable). The ART networks were designed to achieve learning stability while maintaining sensitivity to novel inputs.

In this chapter, the ART1 network was used to illustrate the key concepts of adaptive resonance theory. The ART1 network is based on the Grossberg competitive network of Chapter 18, with a few modifications. The key innovation of ART is the use of "expectations." As each input pattern is presented to the network, it is compared with the prototype vector that it most closely matches (the expectation). If the match between the prototype and the input vector is not adequate, a new prototype is selected. In this way, previously learned memories (prototypes) are not eroded by new learning.

One important point to keep in mind when analyzing ART networks, is that they were designed to be biologically plausible mechanisms for learning. They have as much to do with understanding how the brain works as they do with inspiring practical pattern recognition systems. For this reason, the learning mechanisms are required to use only local information at each neuron. This is not true of all of the learning rules discussed in this text.

Although the ART networks solve the problem of learning instability, in which the network weights never stabilize, there is another stability problem that we have not yet discussed. This is the stability of the differential equations that implement the short-term memory equations of the network. In Layer 2, for example, we have a set of differential equations with nonlinear feedback. Can we make some general statement about the stability of such systems? Chapter 20 will present a comprehensive discussion of this problem.

Further Reading

[CaGr87a] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 54–115, 1987.

In this original presentation of the ART1 architecture, Carpenter and Grossberg demonstrate that the architecture self-organizes and self-stabilizes in response to an arbitrary number of binary input patterns. The key feature of ART is a top-down matching mechanism.

[CaGr87b] G. A. Carpenter and S. Grossberg, "ART2: Self-organization of stable category recognition codes for analog input patterns," *Applied Optics*, vol. 26, no. 23, pp. 4919–4930, 1987.

This article describes an extension of the ART1 architecture that is designed to handle analog input patterns.

[CaGr90] G. A. Carpenter and S. Grossberg, "ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures," *Neural Networks*, vol. 3, no. 23, pp. 129–152, 1990.

This article demonstrates how the Orienting Subsystem of the ART networks could be implemented in biological neurons through the use of chemical transmitters.

[CaGrMa92] G. A. Carpenter, S. Grossberg, N. Markuzon, J. Reynolds and D. Rosen, "Fuzzy ARTMAP: An adaptive resonance architecture for incremental learning of analog maps," *Proceedings of the International Joint Conference on Neural Networks*, Baltimore, MD, vol. 3, no. 5, pp. 309–314, 1992.

The authors present a modification of the ARTMAP architecture to include fuzzy logic that enables better performance in a noisy environment.

[CaGrRe91] G. A. Carpenter, S. Grossberg and J. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks*, vol. 4, no. 5, pp. 169–181, 1991.

This article presents an adaptive resonance theory network for supervised learning. The network consists of two interconnected ART modules. One module receives the input vector, and the other module receives the desired output vector.

[Gros76] S. Grossberg, "Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors," *Biological Cybernetics*, vol. 23, pp. 121–134, 1976.

Grossberg describes a continuous-time competitive network, inspired by the developmental physiology of the visual cortex. The structure of this network forms the foundation for other important networks.

[Gros82] S. Grossberg, *Studies of Mind and Brain*, Boston: D. Reidel Publishing Co., 1982.

This book is a collection of Stephen Grossberg papers from the period 1968 through 1980. It covers many of the fundamental concepts used in later Grossberg networks, such as the adaptive resonance theory networks.

Exercises

E19.1 Consider Layer 1 of the ART1 network with $\varepsilon = 0.02$. Assume two neurons in Layer 2, two elements in the input vector and the following weight matrix and input:

$$\mathbf{W}^{2:1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Also assume that neuron 2 of Layer 2 is active.

i. Find and plot the response $\mathbf{n}^1$ if ${}^{+}b^1 = 2$ and ${}^{-}b^1 = 3$.

ii. Find and plot the response $\mathbf{n}^1$ if ${}^{+}b^1 = 4$ and ${}^{-}b^1 = 5$.

iii. Find and plot the response $\mathbf{n}^1$ if ${}^{+}b^1 = 4$ and ${}^{-}b^1 = 4$.

iv. Check to see that the answers to parts (i)–(iii) satisfy the steady state response predicted by Eq. (19.21). Explain any inconsistencies.

v. Check your answers to parts (i)–(iii) by writing a MATLAB M-file to simulate Layer 1 of the ART1 network. Use the `ode45` routine. Plot the response for each case.

E19.2 Consider Layer 2 of the ART1 network with the following parameters:

$$\varepsilon = 0.1 \quad \mathbf{W}^{1:2} = \begin{bmatrix} ({}_1\mathbf{w}^{1:2})^T \\ ({}_2\mathbf{w}^{1:2})^T \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{2}{3} \\ 1 & 0 \end{bmatrix}$$

and

$$f^2(n) = \begin{cases} 10(n)^2, & n \ge 0 \\ 0, & n < 0 \end{cases}.$$

Assume that the output of Layer 1 is

$$\mathbf{a}^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

i. Write the equations of operation of Layer 2, and simulate and plot the response if the following bias vectors are used:

$$ {}^{+}\mathbf{b}^2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad {}^{-}\mathbf{b}^2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} . $$

ii. Repeat part (i) for the following bias vectors:

$$ {}^{+}\mathbf{b}^2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \quad {}^{-}\mathbf{b}^2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix} . $$

iii. Repeat part (i) for the following bias vectors:

$$ {}^{+}\mathbf{b}^2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \quad {}^{-}\mathbf{b}^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} . $$

iv. Do the results of all of the previous parts satisfy the desired steady state response described in Eq. (19.37)? If not, explain why.

E19.3 Consider the Orienting Subsystem of the ART1 network with the following parameters:

$$ \varepsilon = 0.1 \quad {}^{+}b^0 = {}^{-}b^0 = 2 . $$

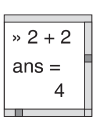
The inputs to the Orienting Subsystem are

$$ \mathbf{p} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{a}^1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} . $$

i. Find and plot the response of the Orienting Subsystem $n^0(t)$, for $\alpha = 0.5 \quad \beta = 4 \quad (\rho = 0.125)$.

ii. Find and plot the response of the Orienting Subsystem $n^0(t)$, for $\alpha = 0.5 \quad \beta = 2 \quad (\rho = 0.25)$.

iii. Verify that the steady state conditions are satisfied in parts (i) and (ii).



iv. Check your answers to parts (i) and (ii) by writing a MATLAB M-file to simulate the Orienting Subsystem.

E19.4 To derive the steady state conditions for the L1-L2 and L2-L1 learning rules, we have made the assumption that the input pattern and the neuron outputs remain constant until the weight matrices converge. This is called "fast learning." Show that this fast learning assumption is equivalent to setting the learning rate α to 1 in the instar and outstar learning rules presented in Chapter 15 and the Kohonen competitive learning rule in Chapter 16.

E19.5 Train an ART1 network using the following input vectors:

$$\mathbf{p}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{p}_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

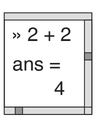
Use the parameter $\zeta = 2$, and choose $S^2 = 3$ (3 categories).

i. Train the network to convergence using $\rho = 0.3$.

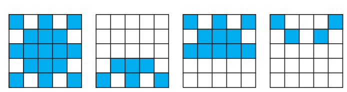
ii. Repeat part (i) using $\rho = 0.6$.

iii. Repeat part (ii) using $\rho = 0.9$.

E19.6 The ART1 algorithm can be modified to add a new neuron in Layer 2 whenever there is no adequate match between the existing prototypes and the input pattern. This involves creating a new row of the $\mathbf{W}^{1:2}$ matrix and a new column of the $\mathbf{W}^{2:1}$ matrix. Describe how this would be done.

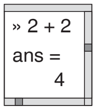


E19.7 Write a Matlab M-file to implement the ART1 algorithm (with the modification described in Exercise E19.6). Use this M-file to train an ART1 network using the following input vectors (see Problem P19.7):

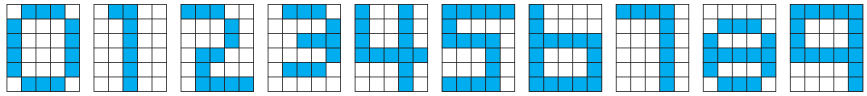


$\mathbf{p}_1$ $\mathbf{p}_2$ $\mathbf{p}_3$ $\mathbf{p}_4$

Present the vectors in the order $\mathbf{p}_1-\mathbf{p}_2-\mathbf{p}_3-\mathbf{p}_1-\mathbf{p}_4$ (i.e., $\mathbf{p}_1$ is presented twice in each epoch). Use the parameters $\zeta = 2$ and $\rho = 0.9$, and choose $S^2 = 3$ (3 categories). Train the network until the weights have converged. Compare your results with Problem P19.7.



E19.8 Recall the digit recognition problem described in Chapter 7 (page 7-10). Train an ART1 network using the digits $0-9$, as displayed below:



$\mathbf{p}_1$ $\mathbf{p}_2$ $\mathbf{p}_3$ $\mathbf{p}_4$ $\mathbf{p}_5$ $\mathbf{p}_6$ $\mathbf{p}_7$ $\mathbf{p}_8$ $\mathbf{p}_9$ $\mathbf{p}_{10}$

Use the parameter $\zeta = 2$, and choose $S^2 = 5$ (5 categories). Use the Matlab M-file from Exercise E19.7.

i. Train the network to convergence using $\rho = 0.3$.

ii. Train the network to convergence using $\rho = 0.6$.

iii. Train the network to convergence using $\rho = 0.9$.

iv. Discuss the results of parts (i)–(iii). Explain the effect of the vigilance parameter.