ADAPTIVE RESONANCE THEORY (ART) NETWORK

- **Adaptive Resonance Theory** (ART) is a family of algorithms for unsupervised learning developed by Carpenter and Grossberg.

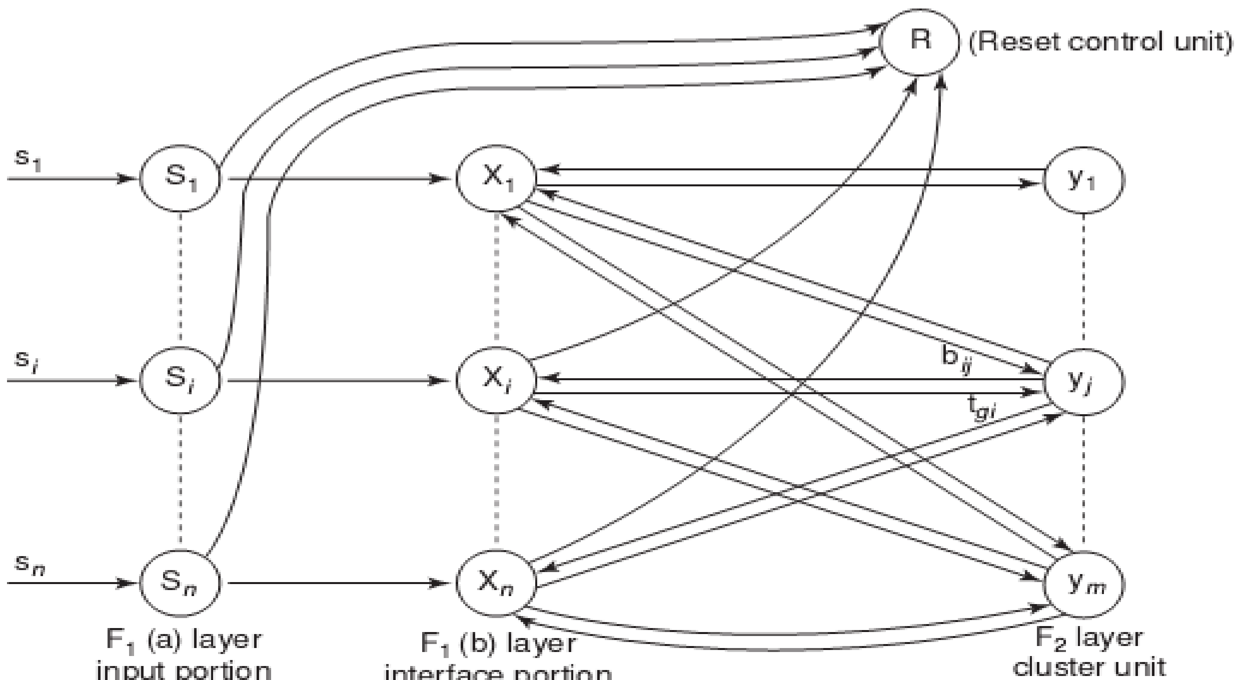
- **ART** is similar to many iterative clustering algorithms where each pattern is processed by

  - finding the "nearest" cluster (a.k.a. prototype or template) to that exemplar (desired).
  - updating that cluster to be "closer" to the exemplar.

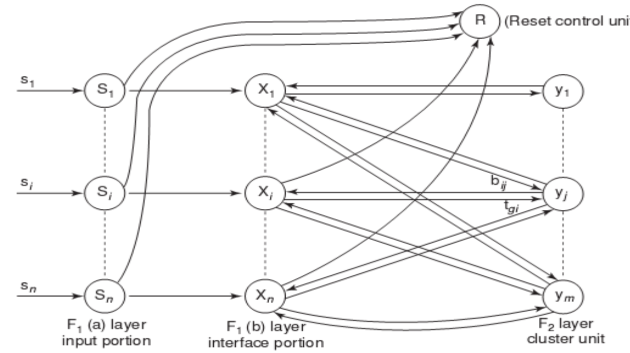
ADAPTIVE RESONANCE THEORY (ART) NETWORK

- Allow user to control the degree of similarity of pattern place on the same cluster.
- The *relative similarity* of input pattern with the weight vector is used rather than the *absolute difference*.
- Stability: a pattern does not oscillate among clusters.
- Plasticity: respond to a new pattern equally well at any stage of learning.
- ART nets are designed to be both stable and plastic.
- ART nets are also structured such that neural processes can control intricate operations of the net. This requires a number of neurons in addition to the input units, cluster units and units for the comparison of the input signal with the cluster unit's weight.

BASIC ARCHITECTURE OF ART1



Architecture

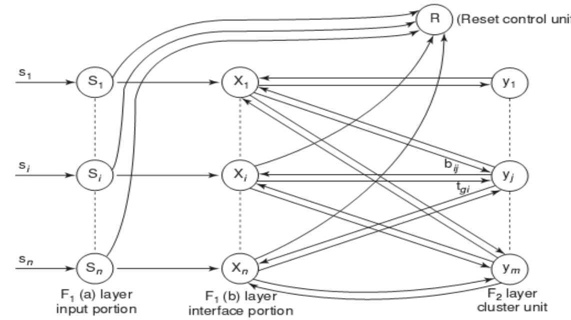


3 groups of neurons:

1. An input processing field (F1 layer).
2. Cluster unit (F2 layer).
3. A mechanism to control degree of similarity of patterns placed on the same cluster.

- F1 layer consists of
  1. Input portion F1(a)
  2. Interface portion F1(b) : combines the signals from input portion & F2 layer to compare the similarity of the input signal to the weight vector of the selected cluster unit
- F2 node is in one of the three states:
  - Active ("on", activation = d; d = 1 for ART1)
  - Inactive ("off", activation=0; but available to participate)
  - Inhibited ("off", activation = 0 & prevented for participate)
- Control of similarity: two sets of connections between each unit in the interface portion and cluster unit.
  - Bottom-up weights $\mathbf{b_{ij}}$ from ith F1 unit to jth F2 unit
  - Top-down weights $\mathbf{t_{ji}}$ from jth F2 unit to ith F1 unit

Operation



- the F2 layer is competitive layer.
  - Cluster unit with largest net input becomes the candidate to learn the input pattern.
  - activations of all other F2 units are set to zero.
- interface units combine the information from input & cluster units
- whether or not the cluster unit is allowed to learn the input patterns depends on how similar its weight vector is to the input vector and is decided by reset unit.
- if a cluster is not allowed to learn it is inhibited and a new cluster unit is selected as the candidate.
- degree of similarity is controlled by user defined vigilance parameter
- a pattern once presented continues to send its input signal until learning trial is completed.

ARCHITECTURES OF ART NETWORK

- **ART1**, designed for binary features.
- **ART2**, designed for continuous (analog) features.
- **ARTMAP**, a supervised version of ART.

and ART2 are presented ...

A *learning trial* in ART consists of the presentation of one input pattern. Before the pattern is presented, the activations of all units in the net are set to zero. All F_2 units are inactive. (Any F_2 units that had been inhibited on a previous learning trial are again available to compete.) Once a pattern is presented, it continues to send its input signal until the learning trial is completed.

The degree of similarity required for patterns to be assigned to the same cluster unit is controlled by a user-specified parameter, known as the *vigilance parameter*. Although the details of the reset mechanism for ART1 and ART2 differ, its function is to control the state of each node in the F_2 layer. At any time, an F_2 node is in one of three states:

active ("on," activation $= d$; $d = 1$ for ART1, $0 < d < 1$ for ART2),
inactive ("off," activation $= 0$, but available to participate in competition), or
inhibited ("off," activation $= 0$, and prevented from participating in any further competition during the presentation of the current input vector).

A summary of the steps that occur in ART nets in general is as follows:

Step 0. Initialize parameters.
Step 1. While stopping condition is false, do Steps 2–9.
  Step 2. For each input vector, do Steps 3–8.
    Step 3. Process F_1 layer.
    Step 4. While reset condition is true, do Steps 5–7.
      Step 5. Find candidate unit to learn the current input pattern:
      F_2 unit (which is not inhibited) with largest input.
      Step 6. $F_1(b)$ units combine their inputs from $F_1(a)$ and F_2.
      Step 7. Test reset condition (details differ for ART1 and ART2):
      If reset is true, then the current candidate unit is rejected (inhibited); return to Step 4.
      If reset is false, then the current candidate unit is accepted for learning; proceed to Step 8.
  Step 8. Learning: Weights change according to differential equations.
Step 9. Test stopping condition.

Basic Architecture

Learning trial: presentation of one pattern (Step 2)

Resonance: in order to learn an input vector by a cluster unit, the maintenance of the top-down and bottom-up signals for an extended period so that the weight changes occur.

Learning:
1. Fast Learning
2. Slow Learning

Basic Architecture

Fast Learning :

- weight updates occur rapidly during resonance as compared to the duration of time a pattern is presented in a trial
- Weights reach equilibrium faster
- Less presentations of patterns are required for learning as compared to slow learning
- Net is considered stabilized when each pattern chooses the correct cluster unit when it is presented
- ART1 – since the patterns are binary, weights associated with each cluster unit also stabilize fast. Equilibrium weights are easy to determine and differential equations control of weight updates is not required.
- ART2 – weights produced by fast learning continue to change each time pattern is presented. Equilibrium weights are not as easy to determine as in ART1. Net stabilizes after a few presentations of each pattern, but the differential equations for weight updates depend on the activation of units whose activations chan

Slow Learning:

- weight changes occur slowly relative to the duration of time a pattern is presented in a trial
- Weights reach equilibrium slowly
- More presentations of patterns are required for learning as compared to fast learning
- Weight changes do not reach equilibrium in any particular learning trial and more trials are required before the net stabilizes
- ART1 – Theoretically slow learning is possible, however generally fast learning is used
- ART2 – weights produced by slow learning are much more appropriate than those produced by fast learning

ART1 UNITS

ART1 is designed to cluster binary input vectors and has direct control of the degree of similarity among patterns placed on he same cluster unit.

ART1 Network is made up of two units

- Computational units
  - Input unit (F1 unit – input and interface).
  - Cluster unit (F2 unit – output).
- Supplemental units
  - One reset control unit. (controls degree of similarity).
  - Two gain control units.

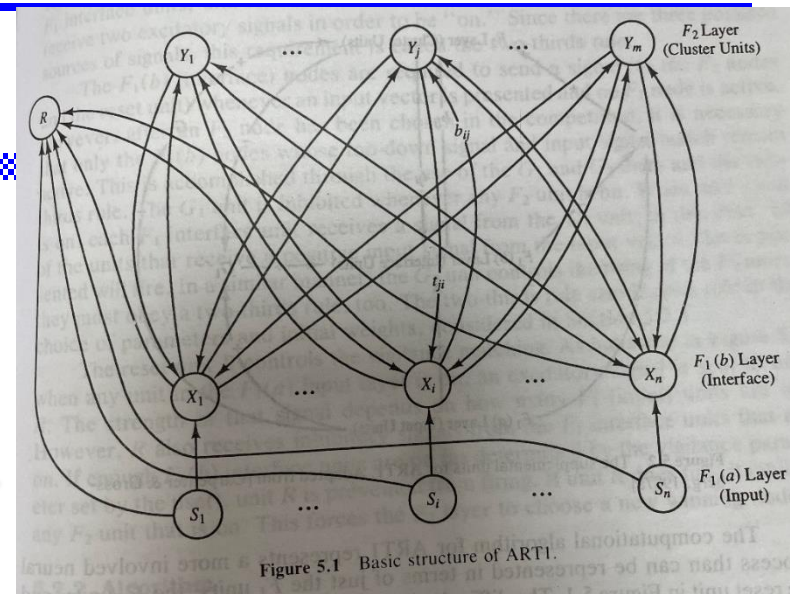
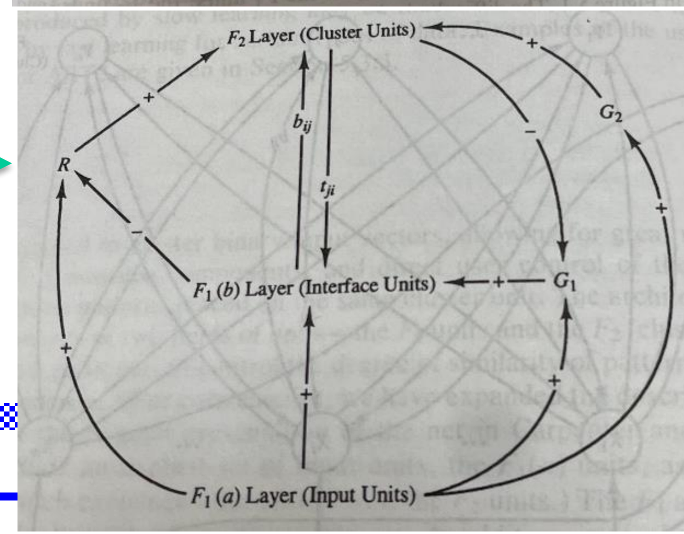


Figure 5.1 Basic structure of ART1.



Training algorithm

The training algorithm for an ART1 net is presented next. A discussion of the role of the parameters and an appropriate choice of initial weights follows.

Step 0. Initialize parameters:

$$L > 1,$$

$$0 < \rho \le 1.$$

Initialize weights:

$$0 < b_{ij}(0) < \frac{L}{L - 1 + n},$$

$$t_{ji}(0) = 1.$$

Step 1. While stopping condition is false, do Steps 2–13.

Step 2. For each training input, do Steps 3–12.

Step 3. Set activations of all F_2 units to zero.
Set activations of $F_1(a)$ units to input vector **s**.

Step 4. Compute the norm of **s**:

$$\|\mathbf{s}\| = \sum_i s_i.$$

Step 5. Send input signal from $F_1(a)$ to the $F_1(b)$ layer:

$$x_i = s_i.$$

Step 6. For each F_2 node that is not inhibited:
If $y_j \neq -1$, then

$$y_j = \sum_i b_{ij} x_i.$$

Step 7. While reset is true, do Steps 8–11.

Step 8. Find J such that $y_J \ge y_j$ for all nodes j.
If $y_J = -1$, then all nodes are inhibited, this pattern cannot be clustered.

Step 9. Recompute activation **x** of $F_1(b)$:

$$x_i = s_i t_{Ji}.$$

Step 10. Compute the norm of vector **x**:

$$\|\mathbf{x}\| = \sum_i x_i.$$

Step 11. Test for reset:

If $\frac{\|\mathbf{x}\|}{\|\mathbf{s}\|} < \rho$, then

$y_J = -1$ (inhibit node J) (and continue, executing Step 7 again).

If $\frac{\|\mathbf{x}\|}{\|\mathbf{s}\|} \ge \rho$,

then proceed to Step 12.

Step 12. Update the weights for node J (fast learning):

$$b_{iJ}(\text{new}) = \frac{L x_i}{L - 1 + \|\mathbf{x}\|},$$

$$t_{Ji}(\text{new}) = x_i.$$

Step 13. Test for stopping condition.

Example 5.1 An ART1 net to cluster four vectors: low vigilance

The values and a description of the parameters in this example are:

n	=	4	number of components in an input vector;
m	=	3	maximum number of clusters to be formed;
ρ	=	0.4	vigilance parameter;
L	=	2	parameter used in update of bottom-up weights;
$b_{ij}(0)$	=	$\dfrac{1}{1+n}$	initial bottom-up weights (one-half the maximum value allowed);
$t_{ji}(0)$	=	1	initial top-down weights.

The example uses the ART1 algorithm to cluster the vectors (1, 1, 0, 0), (0, 0, 0, 1), (1, 0, 0, 0), and (0, 0, 1, 1), in at most three clusters.

Application of the algorithm yields the following:

Step 0. Initialize parameters:

$L = 2,$

$\rho = 0.4;$

Initialize weights:

$b_{ij}(0) = 0.2,$

$t_{ji}(0) = 1.$

Step 1. Begin computation.

Step 2. For the first input vector, (1, 1, 0, 0), do Steps 3–12.

Step 3. Set activations of all F_2 units to zero.

Set activations of $F_1(a)$ units to input vector

$\mathbf{s} = (1, 1, 0, 0).$

Step 4. Compute norm of $\mathbf{s}$:

$\|\mathbf{s}\| = 2.$

Step 5. Compute activations for each node in the F_1 layer:

$\mathbf{x} = (1, 1, 0, 0).$

Step 6. Compute net input to each node in the F_2 layer:

$y_1 = .2(1) + .2(1) + .2(0) + .2(0) = 0.4,$

$y_2 = .2(1) + .2(1) + .2(0) + .2(0) = 0.4,$

$y_3 = .2(1) + .2(1) + .2(0) + .2(0) = 0.4.$

Step 7. While reset is true, do Steps 8–11.

Step 8. Since all units have the same net input,

$J = 1.$

Step 9. Recompute the F_1 activations:

$x_i = s_i t_{1i}$; currently, $\mathbf{t}_1 = (1, 1, 1, 1)$;

therefore, $\mathbf{x} = (1, 1, 0, 0)$

Step 10. Compute the norm of $\mathbf{x}$:

$\|\mathbf{x}\| = 2.$

Step 11. Test for reset:

$\dfrac{\|\mathbf{x}\|}{\|\mathbf{s}\|} = 1.0 \geq 0.4;$

therefore, reset is false.

Proceed to Step 12.

Step 12. Update b_1; for $L = 2$, the equilibrium weights are

$$b_{i1}(\text{new}) = \frac{2x_i}{1 + \|\mathbf{x}\|}.$$

Therefore, the bottom-up weight matrix becomes

$$\begin{bmatrix} .67 & .2 & .2 \\ .67 & .2 & .2 \\ 0 & .2 & .2 \\ 0 & .2 & .2 \end{bmatrix}$$

Update $\mathbf{t}_1$; the fast learning weight values are

$t_{Ji}(\text{new}) = x_i,$

therefore, the top-down weight matrix becomes

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Step 2. For the second input vector, (0, 0, 0, 1), do Steps 3–12.

Step 3. Set activations of all F_2 units to zero.

Set activations of $F_1(a)$ units to input vector

$\mathbf{s} = (0, 0, 0, 1).$

Step 4. Compute norm of $\mathbf{s}$:

$\|\mathbf{s}\| = 1.$

Step 5. Compute activations for each node in the F_1 layer:

$\mathbf{x} = (0, 0, 0, 1).$

Step 6. Compute net input to each node in the F_2 layer:

$y_1 = .67(0) + .67(0) + 0(0) + 0(1) = 0.0,$

$y_2 = .2(0) + .2(0) + .2(0) + .2(1) = 0.2,$

$y_3 = .2(0) + .2(0) + .2(0) + .2(1) = 0.2.$

Step 7. While reset is true, do Steps 8–11.

Step 8. Since units Y_2 and Y_3 have the same net input

$J = 2.$

Step 9. Recompute the activation of the F_1 layer:

$x_i = s_i t_{2i}$;

currently $\mathbf{t}_2 = (1, 1, 1, 1)$; therefore,

$\mathbf{x} = (0, 0, 0, 1).$

Step 10. Compute the norm of $\mathbf{x}$:

$\|\mathbf{x}\| = 1.$

Step 11. Test for reset:

$\dfrac{\|\mathbf{x}\|}{\|\mathbf{s}\|} = 1.0 \geq 0.4;$

therefore, reset is false. Proceed to Step 12.

Step 12. Update $\mathbf{b}_2$; the bottom-up weight matrix becomes

$$\begin{bmatrix} .67 & 0 & .2 \\ .67 & 0 & .2 \\ 0 & 0 & .2 \\ 0 & 1 & .2 \end{bmatrix}$$

Update $\mathbf{t}_2$; the top-down weight matrix becomes

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Step 2. For the third input vector, (1, 0, 0, 0), do Steps 3–12.

Step 3. Set activations of all F_2 units to zero.

Set activations of $F_1(a)$ units to input vector

$\mathbf{s} = (1, 0, 0, 0).$

Step 4. Compute norm of $\mathbf{s}$:

$\|\mathbf{s}\| = 1.$

Step 5. Compute activations for each node in the F_1 layer:

$\mathbf{x} = (1, 0, 0, 0).$

Step 6. Compute net input to each node in the F_2 layer:

$$y_1 = .67(1) + .67(0) + 0(0) + 0(0) = 0.67,$$
$$y_2 = 0(1) + 0(0) + 0(0) + 1(0) = 0.0,$$
$$y_3 = .2(1) + .2(0) + .2(0) + .2(0) = 0.2.$$

Step 7. While reset is true, do Steps 8–11.

Step 8. Since unit Y_1 has the largest net input,

$$J = 1.$$

Step 9. Recompute the activation of the F_1 layer:

$$x_i = s_i t_{1i};$$

current, $t_1 = (1, 1, 0, 0)$; therefore,

$$\mathbf{x} = (1, 0, 0, 0).$$

Step 10. Compute the norm of $\mathbf{x}$:

$$\|x\| = 1.$$

Step 11. $\|x\| / \|s\| = 1.0$ Proceed to Step 12.

Step 12. Update $\mathbf{b}_1$; the bottom-up weight matrix becomes

$$\begin{bmatrix} 1 & 0 & .2 \\ 0 & 0 & .2 \\ 0 & 0 & .2 \\ 0 & 1 & .2 \end{bmatrix}$$

Update $\mathbf{t}_1$; the top-down weight matrix becomes

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Step 2. For the fourth input vector, (0, 0, 1, 1), do Steps 3–12.

Step 3. Set activations of all F_2 units to zero.

Set activations of $F_1(a)$ units to input vector

$$\mathbf{s} = (0, 0, 1, 1).$$

Step 4. Compute norm of $\mathbf{s}$:

$$\|\mathbf{s}\| = 2.$$

Step 5. Compute activations for each node in the F_1 layer:

$$\mathbf{x} = (0, 0, 1, 1).$$

Step 6. Compute net input to each node in the F_2 layer:

$$y_1 = 1(0) + 0(0) + 0(1) + 0(1) = 0.0,$$
$$y_2 = 0(0) + 0(0) + 0(1) + 1(1) = 1.0,$$
$$y_3 = .2(0) + .2(0) + .2(1) + .2(1) = 0.4.$$

Step 7. While reset is true, do Steps 8–11.

Step 8. Since unit Y_2 has the largest net input,

$$J = 2.$$

Step 9. Recompute the activation of the F_1 layer:

$$x_i = s_i t_{2i};$$

currently, $t_2 = (0, 0, 0, 1)$; therefore,

$$\mathbf{x} = (0, 0, 0, 1).$$

Step 10. Compute the norm of $\mathbf{x}$:

$$\|\mathbf{x}\| = 1.$$

Step 11. Test for reset:

$$\frac{\|\mathbf{x}\|}{\|\mathbf{s}\|} = 0.5 \geq 0.4;$$

therefore, reset is false. Proceed to Step 12.

Step 12. Update $\mathbf{b}_2$; however, there is no change in the bottom-up weight matrix:

$$\begin{bmatrix} 1 & 0 & .2 \\ 0 & 0 & .2 \\ 0 & 0 & .2 \\ 0 & 1 & .2 \end{bmatrix}$$

Similarly, the top-down weight matrix remains

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Step 13. Test stopping condition.

(This completes one epoch of training.)

The reader can check that no further learning takes place on subsequent presentations of these vectors, regardless of the order in which they are presented. Depending on the order of presentation of the patterns, more than one epoch may be required, but typically, stable weight matrices are obtained very quickly.