Adaptive Resonance Theory (ART)

Adaptive Resonance Theory (ART) networks perform completely unsupervised learning.

Their competitive learning algorithm is similar to the first (unsupervised) phase of CPN learning.

However, ART networks are able to grow additional neurons if a new input cannot be categorized appropriately with the existing neurons.

A vigilance parameter ρ determines the tolerance of this matching process.

A greater value of ρ leads to more, smaller clusters (= input samples associated with the same winner neuron).

Adaptive Resonance Theory (ART)

ART networks consist of an input layer and an output layer.

We will only discuss ART-1 networks, which receive binary input vectors.

Bottom-up weights are used to determine output-layer candidates that may best match the current input.

Top-down weights represent the “prototype” for the cluster defined by each output neuron.

A close match between input and prototype is necessary for categorizing the input.

Finding this match can require multiple signal exchanges between the two layers in both directions until “resonance” is established or a new neuron is added.

Adaptive Resonance Theory (ART)

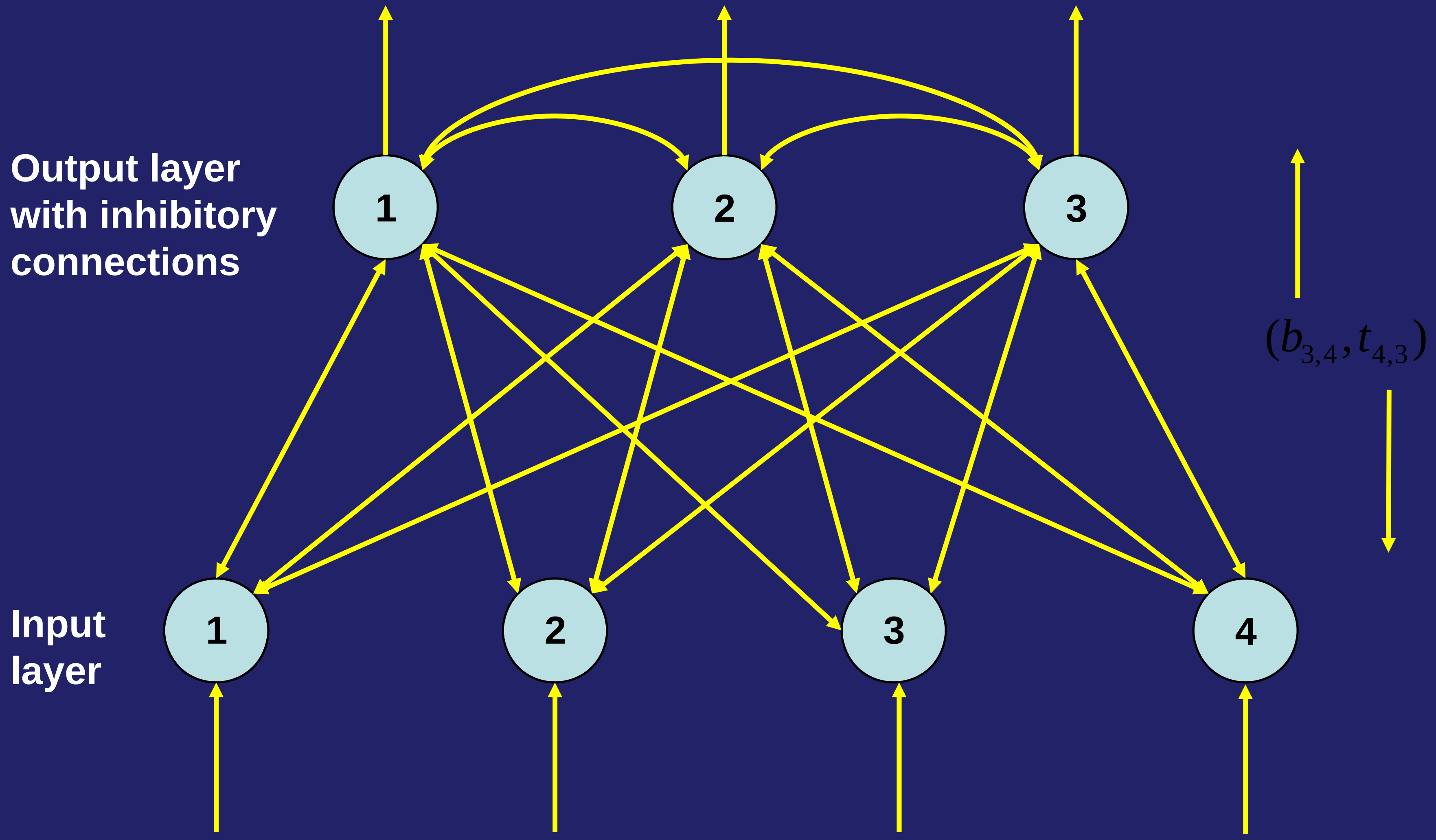
ART networks tackle the stability-plasticity dilemma:

Plasticity: They can always adapt to unknown inputs (by creating a new cluster with a new weight vector) if the given input cannot be classified by existing clusters.

Stability: Existing clusters are not deleted by the introduction of new inputs (new clusters will just be created in addition to the old ones).

Problem: Clusters are of fixed size, depending on ρ.

The ART-1 Network



```
A.  Initialize each top-down weight $t_{l,j}(0) = 1$;
B.  Initialize bottom-up weight $b_{j,l}(0) = \frac{1}{n+1}$;
C.  While the network has not stabilized, do
    1.  Present a randomly chosen pattern $x = (x_1,\ldots,x_n)$ for learning
    2.  Let the active set $A$ contain all nodes; calculate
            $y_j = b_{j,1} x_1 + \ldots + b_{j,n} x_n$ for each node $j \in A$;
    3.  Repeat
        a)  Let $j^*$ be a node in $A$ with largest $y_j$, with ties being broken arbitrarily;
        b)  Compute $s^* = (s^*_1,\ldots,s^*_n)$ where $s^*_l = t_{l,j^*} x_l$ ;
        c)  Compare similarity between $s^*$ and $x$ with the given vigilance parameter $\rho$:
            if $\frac{\sum_{l=1}^{n} s_l^*}{\sum_{l=1}^{n} x_l} \le \rho$ then remove $j^*$ from set $A$
          else associate $x$ with node $j^*$ and update weights:
            $b_{j^*l}(\text{new}) = \frac{t_{l,j^*}(old)x_l}{0.5 + \sum_{l=1}^{n} t_{l,j^*}(old)x_l}$        $t_{l,j^*}(\text{new}) = t_{l,j^*}(old)x_l$
      Until $A$ is empty or $x$ has been associated with some node $j$
    4.  If A is empty, then create new node whose weight vector coincides with current input
        pattern x;
  end-while
```

ART Example Computation

For this example, let us assume that we have an ART-1 network with 7 input neurons ($n = 7$) and initially one output neuron ($n = 1$).

Our input vectors are

{(1, 1, 0, 0, 0, 0, 1),
 (0, 0, 1, 1, 1, 1, 0),
 (1, 0, 1, 1, 1, 1, 0),
 (0, 0, 0, 1, 1, 1, 0),
 (1, 1, 0, 1, 1, 1, 0)}

and the vigilance parameter $\rho = 0.7$.

Initially, all top-down weights are set to $t_{l,1}(0) = 1$, and all bottom-up weights are set to $b_{1,l}(0) = 1/8$.

ART Example Computation

For the first input vector, (1, 1, 0, 0, 0, 0, 1), we get:

$$y_1 = \frac{1}{8}\cdot 1 + \frac{1}{8}\cdot 1 + \frac{1}{8}\cdot 0 + \frac{1}{8}\cdot 0 + \frac{1}{8}\cdot 0 + \frac{1}{8}\cdot 0 + \frac{1}{8}\cdot 1 = \frac{3}{8}$$

Clearly, y_1 is the winner (there are no competitors).

Since we have:

$$\frac{\sum_{l=1}^{7} t_{l,1} x_l}{\sum_{l=1}^{7} x_l} = \frac{3}{3} = 1 > 0.7,$$

the vigilance condition is satisfied and we get the following new weights:

$$b_{1,1}(1) = b_{1,2}(1) = b_{1,7}(1) = \frac{1}{0.5+3} = \frac{1}{3.5}$$

$$b_{1,3}(1) = b_{1,4}(1) = b_{1,5}(1) = b_{1,6}(1) = 0$$

ART Example Computation

Also, we have:

$$t_{l,1}(1) = t_{l,0}(0)x_l$$

We can express the updated weights as matrices:

$$B(1) = \begin{bmatrix} \frac{1}{3.5} & \frac{1}{3.5} & 0 & 0 & 0 & 0 & \frac{1}{3.5} \end{bmatrix}^{\mathrm{T}}$$

$$T(1) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$$

Now we have finished the first learning step and proceed by presenting the next input vector.

ART Example Computation

For the second input vector, (0, 0, 1, 1, 1, 1, 0), we get:

$$y_1 = \frac{1}{3.5} \cdot 0 + \frac{1}{3.5} \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 + \frac{1}{3.5} \cdot 0 = 0$$

Of course, y_1 is still the winner.

However, this time we do not reach the vigilance threshold:

$$\frac{\sum_{l=1}^{7} t_{l,1} x_l}{\sum_{l=1}^{7} x_l} = \frac{0}{4} = 0 < 0.7.$$

This means that we have to generate a second node in the output layer that represents the current input.

Therefore, the top-down weights of the new node will be identical to the current input vector.

ART Example Computation

The new unit's bottom-up weights are set to zero in the positions where the input has zeroes as well.

The remaining weights are set to:
1/(0.5 + 0 + 0 + 1 + 1 + 1 + 1 + 0)

This gives us the following updated weight matrices:

$$B(2) = \begin{bmatrix} 1/3.5 & 1/3.5 & 0 & 0 & 0 & 0 & 1/3.5 \\ 0 & 0 & 1/4.5 & 1/4.5 & 1/4.5 & 1/4.5 & 0 \end{bmatrix}^{\mathrm{T}}$$

$$T(2) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$$

ART Example Computation

For the third input vector, (1, 0, 1, 1, 1, 1, 0), we have:

$$y_1 = \frac{1}{3.5}; \qquad y_2 = \frac{4}{4.5}$$

Here, y_2 is the clear winner.

This time we exceed the vigilance threshold again:

$$\frac{\sum_{l=1}^{7} t_{l,2} x_l}{\sum_{l=1}^{7} x_l} = \frac{4}{5} = 0.8 > 0.7.$$

Therefore, we adapt the second node’s weights.

Each top-down weight is multiplied by the corresponding element of the current input.

ART Example Computation

The new unit's bottom-up weights are set to the top-down weights divided by (0.5 + 0 + 0 + 1 + 1 + 1 + 1 + 0).

It turns out that, in the current case, these updates do not result in any weight changes at all:

$$B(3) = \begin{bmatrix} 1/3.5 & 1/3.5 & 0 & 0 & 0 & 0 & 1/3.5 \\ 0 & 0 & 1/4.5 & 1/4.5 & 1/4.5 & 1/4.5 & 0 \end{bmatrix}^{\mathrm{T}}$$

$$T(3) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$$

ART Example Computation

For the fourth input vector, (0, 0, 0, 1, 1, 1, 0), it is:

$$y_1 = 0; \qquad y_2 = \frac{3}{4.5}$$

Again, y_2 is the winner.

The vigilance test succeeds once again:

$$\frac{\sum_{l=1}^{7} t_{l,2} x_l}{\sum_{l=1}^{7} x_l} = \frac{3}{3} = 1 > 0.7.$$

Therefore, we adapt the second node’s weights.

As usual, each top-down weight is multiplied by the corresponding element of the current input.

ART Example Computation

The new unit's bottom-up weights are set to the top-down weights divided by
(0.5 + 0 + 0 + 0 + 1 + 1 + 1 + 0).

This gives us the following new weight matrices:

$$B(4) = \begin{bmatrix} 1/3.5 & 1/3.5 & 0 & 0 & 0 & 0 & 1/3.5 \\ 0 & 0 & 0 & 1/3.5 & 1/3.5 & 1/3.5 & 0 \end{bmatrix}^{\mathrm{T}}$$

$$T(4) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$$

ART Example Computation

Finally, the fifth input vector, (1, 1, 0, 1, 1, 1, 0), gives us:

$$y_1 = \frac{2}{3.5}; \qquad y_2 = \frac{3}{3.5}$$

Once again, y_2 is the winner.

The vigilance test fails this time:

$$\frac{\sum_{l=1}^{7} t_{l,2} x_l}{\sum_{l=1}^{7} x_l} = \frac{3}{5} = 0.6 < 0.7.$$

This means that the active set A is reduced to contain only the first node, which becomes the uncontested winner.

ART Example Computation

The vigilance test fails for the first unit as well:

$$\frac{\sum_{l=1}^{7} t_{l,1} x_l}{\sum_{l=1}^{7} x_l} = \frac{2}{5} = 0.4 < 0.7.$$

We thus have to create a third output neuron, which gives us the following new weight matrices:

$$B(5) = \begin{bmatrix} 1/3.5 & 1/3.5 & 0 & 0 & 0 & 0 & 1/3.5 \\ 0 & 0 & 0 & 1/3.5 & 1/3.5 & 1/3.5 & 0 \\ 1/5.5 & 1/5.5 & 0 & 1/5.5 & 1/5.5 & 1/5.5 & 0 \end{bmatrix}^T$$

$$T(5) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}^T$$

ART Example Computation

In the second epoch, the first input vector, (1, 1, 0, 0, 0, 0, 1), gives us:

$$y_1 = \frac{3}{3.5}; \qquad y_2 = 0; \qquad y_3 = \frac{2}{5.5}$$

Here, y_1 is the winner, and the vigilance test succeeds:

$$\frac{\sum_{l=1}^{7} t_{l,1} x_l}{\sum_{l=1}^{7} x_l} = \frac{3}{3} = 1 > 0.7.$$

Since the current input is identical to the winner's top-down weights, no weight update happens.

ART Example Computation

The second input vector, (0, 0, 1, 1, 1, 1, 0), results in:

$$y_1 = 0; \qquad y_2 = \frac{3}{3.5}; \qquad y_3 = \frac{3}{5.5}$$

Now y_2 is the winner, and the vigilance test succeeds:

$$\frac{\sum_{l=1}^{7} t_{l,2} x_l}{\sum_{l=1}^{7} x_l} = \frac{3}{3} = 1 > 0.7.$$

Again, because the current input is identical to the winner’s top-down weights, no weight update occurs.

ART Example Computation

The third input vector, (1, 0, 1, 1, 1, 1, 0), give us:

$$y_1 = \frac{1}{3.5}; \qquad y_2 = \frac{3}{3.5}; \qquad y_3 = \frac{4}{5.5}$$

Once again, y_2 is the winner, but this time the vigilance test fails:

$$\frac{\sum_{l=1}^{7} t_{l,2} x_l}{\sum_{l=1}^{7} x_l} = \frac{3}{5} = 0.6 < 0.7.$$

This means that the active set is reduced to A = {1, 3}.

Since $y_3 > y_1$, the third node is the new winner.

ART Example Computation

The third node does satisfy the vigilance threshold:

$$\frac{\sum_{l=1}^{7} t_{l,3} x_l}{\sum_{l=1}^{7} x_l} = \frac{4}{5} = 0.8 > 0.7.$$

This gives us the following updated weight matrices:

$$B(8) = \begin{bmatrix} 1/3.5 & 1/3.5 & 0 & 0 & 0 & 0 & 1/3.5 \\ 0 & 0 & 0 & 1/3.5 & 1/3.5 & 1/3.5 & 0 \\ 1/4.5 & 0 & 0 & 1/4.5 & 1/4.5 & 1/4.5 & 0 \end{bmatrix}^T$$

$$T(8) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}^T$$

ART Example Computation

For the fourth vector, (0, 0, 0, 1, 1, 1, 0), the second node wins, passes the vigilance test, but no weight changes occur.

The fifth vector, (1, 1, 0, 1, 1, 1, 0), makes the second unit win, which fails the vigilance test.

The new winner is the third output neuron, which passes the vigilance test but does not lead to any weight modifications.

Further presentation of the five sample vectors do not lead to any weight changes; the network has thus stabilized.

Adaptive Resonance Theory

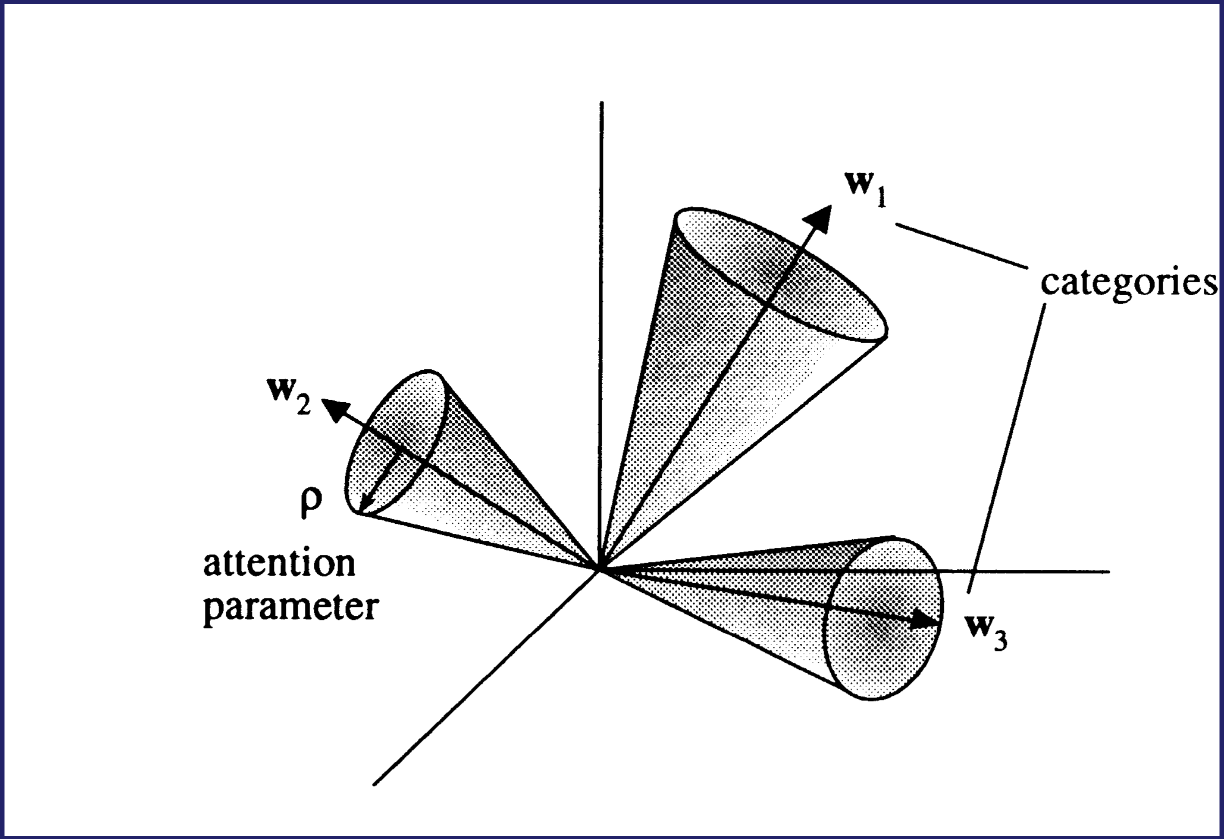


Illustration of the categories (or clusters) in input space formed by ART networks. Notice that increasing ρ leads to narrower cones and not to wider ones as suggested by the figure.

Adaptive Resonance Theory

A problem with ART-1 is the need to determine the vigilance parameter ρ for a given problem, which can be tricky.

Furthermore, ART-1 always builds clusters of the same size, regardless of the distribution of samples in the input space.

Nevertheless, ART is one of the most important and successful attempts at simulating incremental learning in biological systems.