Adaptive Resonance Theory (ART) Networks

Proposed by
Profs. Stephen Grossberg &
Gail Carpenter
Boston University
ART Network Varieties

- Several varieties:
  - ART1: Discrete (e.g. binary) patterns
  - ART2: Continuous patterns ...
  - ARTMAP: supervised learning
  - Fuzzy Art: Fuzzy version of ART1

- All work by competitive learning and a type of clustering, represented by stored prototypes, with other nuances
Stability-Plasticity Dilemma

- **stability**: Recognized patterns should be insensitive to noise.

- **plasticity**: System should be capable of learning new patterns.

- The conflict between these is one of the things that ART tries to resolve.
Some ART Applications

- Disease identification (HMC Math Clinic)
- Tech support email automation (text similarity) (HMC CS Clinic)
- Satellite data anomaly detection (HMC CS Clinic)
- Music recognition
- Distinguishing poisonous vs. edible mushrooms
- Modeling biological neural processes
ART Networks

- Originally biologically motivated by an ODE model
- Models short- and long-term memory
- Combine supervised and unsupervised (competitive, clustering)
- Dynamically create new categories, controllable by an attention parameter called “vigilance”
Competitive Learning in ART

- = category pattern
- = possible data points

implied clusters

vigilance threshold:
nearest prototype
too far from data
to be considered the correct category
Competitive Learning in ART

- clusters

- = category pattern
- = possible data points

new category created

data
ART: instars and outstars

**Input layer**: Normalize input, Compare input with Expectation

**Storage layer**: Competition, lateral Inhibition, Contrast enhancement

**Orienting subsystem**: (reset when pattern doesn’t match)

**in-stars**

**out-stars**

**Gain Control**

**Stored prototype patterns**

**Input**

**Classification (prototype)**
ATTENTIONAL SYSTEM

STM $F_1$

STM $F_2$

INPUT

Nonspecific inhibitory gain control

Reset and Search

Matching criterion: vigilance parameter

Carpenter and Grossberg (1987)
ART Hypothesis Testing and Learning Cycle

Choose category, or symbolic representation

Input

Mismatch reset

Test hypothesis

Vigilance

Try another category
Learning takes place at the synapses denoted by semi-circular endings in the F1 to F2 pathways. These are LTM.
1. The input pattern I is instated across the feature detectors at level $F_1$ as a short-term memory (STM) activity pattern $X$ represented by the hatched pattern across $F_1$.

2. Input I also \textit{nonspecifically} activates the orienting subsystem A

3. Pattern X both inhibits A and generates the output pattern S from $F_1$.

4. Pattern S is multiplied by long term memory (LTM) traces and added at $F_2$ nodes to form the input pattern T, which activates the STM pattern $Y$ across the categories coded at level $F_2$. 
5. Pattern Y generates the top-down output pattern U, which is multiplied by top-down LTM traces and added at $F_1$ nodes to form the prototype pattern V that encodes the learned expectation of the active $F_2$ nodes.

6. If V mismatches I at $F_1$, then a new STM pattern $X^*$ is generated at $F_1$. $X^*$ is represented by the hatched pattern and includes the features of I that are confirmed by V. Inactivated nodes corresponding to unconfirmed features of X are unhatched.

7. The reduction in total STM activity, which occurs when X is transformed into $X^*$ causes a decrease in the total inhibition from $F_1$ to A.
8. If inhibition decreases sufficiently [vigilance condition not met], \( A \) releases a nonspecific arousal wave to \( F_2 \), which resets the STM pattern \( Y \) at \( F_2 \).
10. After Y is inhibited, its top-down prototype signal is eliminated, and X can be reinstated at $F_1$.

11. **Enduring traces** of the prior reset lead X to activate a different STM pattern $Y^*$ at $F_2$.

12. If the top-down prototype due to $Y^*$ also mismatches I at $F_1$, then the **search** for an appropriate $F_2$ code continues **until a more appropriate** $F_2$ representation is selected.

13. Then an attentive **resonance** develops and learning of the attended data is initiated.
ART Learning

- The **resonant state**, rather than bottom-up activation, drives the learning process (hence “adaptive resonance” theory):
  - “resonance” = **mutual reinforcement** between input and storage layers
  - “adaptive” = weights are adjusted when **resonance** occurs

- ART systems **learn prototypes** rather than exemplars because the attendant **feature vector** $X^*$ rather than the input **exemplar itself** is learned.
ART1 Viewpoints

- Two kinds of explanations:
  - neural - as previously presented
  - algorithmic

- The first is more complicated, since it involves neural explanations for the control aspects of the algorithmic approach.
ART Algorithmic View

- Input pattern presented to input layer.
- Storage layer indicates tentative hypothetical classification.
- Input layer decides if hypothetical is close enough; if so, done.
- If not, storage layer indicates alternate hypothesis.
- The above two steps are repeated until the hypothetical classification is accepted.
- All hypotheses could be rejected; in this case, a new class is created in the storage layer.
ART1 Flowchart

1. Initialize weights $W, V, \rho$

2. Present pattern $x$ to MAXNET

3. Find the best matching cluster $j$ among $M$ existing clusters

4. Perform the similarity test for $x$ and cluster $j$

5. Update $W, V$ for the cluster

6. Disable node $j$ by forcing $y_j = 0$

Is there more than a single top layer node left?

Add new cluster

Is further adaptation needed?

Yes

No
Algorithm ART1:

Initialize each $t_{\ell,j}(0) = 1$, $b_{j,\ell}(0) = \frac{1}{n+1}$;

while the network has not stabilized, do

1. Let $A$ contain all nodes;

2. For a randomly chosen input vector $x$, compute $y_{j} = b_{j} \cdot x$ for each $j \in A$.

3. repeat

   (a) Let $j^*$ be a node in $A$ with largest $y_j$.

   (b) Compute $s^* = (s_1^*, \ldots, s_n^*)$ where $s_{\ell}^* = t_{\ell,j^*} x_{\ell}$;

   (c) If $\frac{\sum_{\ell=1}^{n} s_{\ell}^*}{\sum_{\ell=1}^{n} x_{\ell}} \leq \rho$ then remove $j^*$ from set $A$

   else associate $x$ with node $j^*$ and update weights:

   \[ b_{j^*,\ell}(\text{new}) = \frac{t_{\ell,j^*}(\text{old}) x_{\ell}}{0.5 + \sum_{\ell=1}^{n} t_{\ell,j^*}(\text{old}) x_{\ell}} \]

   \[ t_{\ell,j^*}(\text{new}) = t_{\ell,j^*}(\text{old}) x_{\ell} \]

   until $A$ is empty or $x$ is associated with some node;

4. If $A$ is empty, create new node with weight vector $x$; end-while.
Let x be the input pattern (a vector of \{0, 1\}). n is the dimension of x.

**For each prototype i**: compute the activation value

\[ y[i] = \frac{(w[i]^*x)}{(w[i]^*w[i])} \]

where \( w[i] \) is the weight vector of the prototype and \( ^* \) is inner product.

\( y[i] \) is high when the input is close to the weight vector. The denominator normalizes the product so that a combination of pattern and weight with a large number of 1's don't artificially dominate.

**For each prototype i**, in order of **decreasing value of** \( y[i] \) (i.e. increasing distance from input):

- **If** \( y[i] \leq \frac{x^*x}{n} \) (x is **too distant** from prototype i, and the distance is increasing.)
  
  break;

- **else if** \( \frac{(w[i]^*x)}{(x^*x)} \geq \text{vigilance} \) (“resonance”)
  
  replace \( w[i] \) with \( w[i] \& x \) (bit-wise and) then **return**.

  ("resonance": x is close enough to prototype i, and has enough bits in common with it (i.e. exceeding the set vigilance) that simple learning suffices, and the prototype is used as the match. This is more likely to happen if vigilance is low, i.e. a high vigilance is more likely to cause the creation of a new prototype.

If the iteration ends without an early return, **create a new prototype** with \( w = x \).
ART1 Issues

- Subset-Superset dilemma:
  - If one pattern is contained in another, then a given input may have the same inner product with two different prototypes.
  - Resolvable by keeping the prototypes normalized.
Example of Subset/Superset Dilemma

Suppose that $\mathbf{W}^{1:2} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ i.e. the prototypes are $\mathbf{w}^{1:2}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$, $\mathbf{w}^{1:2}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$.

Here $\mathbf{w}^{1:2}_1$ is a subset of $\mathbf{w}^{1:2}_2$.

If the output of layer 1 is $\mathbf{a}^1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ then the input to Layer 2 will be

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

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.
Subset/Superset Resolution

Normalize the prototype patterns.

\[
W^{1:2} = \begin{bmatrix}
\frac{1}{2} & 1 & 0 \\
\frac{1}{3} & 2 & 1 \\
\frac{1}{3} & 3 & 3
\end{bmatrix}
\]

\[
W^{1:2} a^{1} = \begin{bmatrix}
\frac{1}{2} & 1 & 0 \\
\frac{1}{3} & 2 & 1 \\
\frac{1}{3} & 3 & 3
\end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}
\]

Now we have the desired result; the first prototype has the larger inner product with the input.
Increasing vigilance causes the network to be more selective, introducing a new prototype when the match is weak.
ART2

- ART2 allows continuous-valued patterns
- whereas ART1 is limited to discrete-valued ones
ART2 Algorithm

Let: 
\(\alpha: \) positive number \(\alpha \leq 1/\sqrt{N_{EX}}\)
\(\beta: \) small positive number
\(\theta: \) normalization parameter \(1 < \theta < 1/\sqrt{N_{EX}}\)
\(\rho: \) vigilance parameter \(0 \leq \rho, 1\)

0. For each example \(x^{(n)}\) in the database
   0a. Normalize \(x^{(n)}\) to have magnitude 1
   0b. Zero out coordinates \(x^{(n)} < \theta\) (remove small noise signals)
   0c. Re-normalize \(x^{(n)}\)

1. Start with no prototype vectors (clusters)

2. Perform iterations until no example causes any change. At this point quit because stability has been achieved. For each iteration, choose the next example \(x^{(n)}\) in cyclic order

3. Find the prototype \(w_k\) (cluster) not yet tried during this iteration that maximizes \(w_k^T x^{(n)}\)

4. Test whether \(w_k\) is sufficiently similar to \(x^{(n)}\)
   \[w_k^T x^{(n)} \geq \alpha \sum_{j=1}^{N_{DIM}} x^{(n)}(j)\]
   4a. If not then
      4a1. Make a new cluster with prototype set to \(x^{(n)}\)
      4a2. End this iteration and return to step 2 for the next example
   4b. If sufficiently similar, then test for vigilance acceptability \(w_k^T x^{(n)} \geq \rho\)
      4b1. If acceptable then \(x^{(n)}\) belongs to \(w_k\). Modify \(w_k\) to be more like \(x^{(n)}\)
         \[w_k = \frac{(1 - \beta)w_k + \beta x^{(n)}}{\| (1 - \beta)w_k + \beta x^{(n)} \|}\]
         and go to step 2 for the next iteration with the next example
      4b2. If not acceptable, then make a new cluster with prototype set to \(x^{(n)}\)

[Garrett, 1993]
ART2 Clustering (at low vigilance)
ART2 Clustering (at higher vigilance)
Fuzzy ART Supplants ART2

- ART2 was evidently too complicated.

- Fuzzy-ART, based on Fuzzy Logic, was its replacement.

- Fuzzy Logic uses the min operator as conjunction. Values are continuous between 0 and 1, rather than discrete.
**ART 1** (BINARY)  \n\n**FUZZY ART** (ANALOG)  

---

**CATEGORY CHOICE**

\[ T_j = \frac{|I \cap w_j|}{\alpha + |w_j|} \quad \text{and} \quad T_j = \frac{|I \wedge w_j|}{\alpha + |w_j|} \]

**MATCH CRITERION**

\[ \frac{|I \cap w|}{|I|} \geq \rho \quad \text{and} \quad \frac{|I \wedge w|}{|I|} \geq \rho \]

**FAST LEARNING**

\[ w_j^{(new)} = I \cap w_j^{(old)} \quad \text{and} \quad w_j^{(new)} = I \wedge w_j^{(old)} \]

\[ \cap = \text{logical AND} \quad \wedge = \text{fuzzy AND intersection} \]

---

Fig. 2. Comparison of ART 1 and fuzzy ART.
Complement Encoding

An optional feature of Fuzzy ART is complement coding, a means of incorporating the absence of features into pattern classifications, which helps prevent inefficient and unnecessary category proliferation.

Complement encoding augments each input data vector $x$ with the complement of that vector $1-x$.

Complement encoding can also achieve the same effect as normalization in preventing the subset/superset dilemma.
ART Maps: Supervised Learning

- ARTMAP, also known as Predictive ART, combines two slightly modified ART1 or ART2 units into a supervised learning structure:
  - the first unit takes the input data and
  - the second unit takes the correct output data,

then makes the minimum possible adjustment of the vigilance parameter in the first unit in order to make the correct classification.

- Fuzzy ART Map does this with Fuzzy ART units, which subsumes both ART1 and ART2.
Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensionalal Maps

Gail A. Carpenter, Stephen Grossberg, Natalya Markuzon, John H. Reynolds, and David B. Rosen, Student Member, IEEE
During supervised learning, \( \text{ART}_a \) receives a stream \( \{a^{(p)}\} \) of input patterns, and \( \text{ART}_b \) receives a stream \( \{b^{(p)}\} \) of input patterns, where \( b^{(p)} \) is the correct prediction given \( a^{(p)} \). These modules are linked by an associative learning network and an internal controller that ensures autonomous system operation in real time. The controller is designed to create the minimal number of \( \text{ART}_a \) recognition categories, or “hidden units,” needed to meet accuracy criteria. It does this by realizing a minimax learning rule that enables an ARTMAP system to learn quickly, efficiently, and accurately as it conjointly minimizes predictive error and maximizes predictive generalization. This scheme automatically links predictive success to category size on a trial-by-trial basis using only local operations. It works by increasing the vigilance parameter \( \rho_a \) of \( \text{ART}_a \) by the minimal amount needed to correct a predictive error at \( \text{ART}_b \).
Fig. 1. Fuzzy ARTMAP architecture. The ART<sub>a</sub> complement coding preprocessor transforms the $M_a$ vector $a$ into the $2M_a$ vector $A = (a, a^c)$ at the ART<sub>a</sub> field $F_0^a$. $A$ is the input vector to the ART<sub>a</sub> field $F_1^a$. Similarly, the input to $F_1^b$ is the $2M_b$ vector $(b, b^c)$. When a prediction by ART<sub>a</sub> is disconfirmed at ART<sub>b</sub>, inhibition of map field activation induces the match tracking process. Match tracking raises the ART<sub>a</sub> vigilance ($\rho_a$) to just above the $F_1^a$ to $F_0^a$ match ratio $|x^a|/|A|$. This triggers an ART<sub>a</sub> search which leads to activation of either an ART<sub>a</sub> category that correctly predicts $b$ or to a previously uncommitted ART<sub>a</sub> category node.
Parameters: Fuzzy ART dynamics are determined by a choice parameter $\alpha > 0$; a learning rate parameter $\beta \in [0, 1]$; and a vigilance parameter $\rho \in [0, 1]$.

Category Choice: For each input $I$ and $F_2$ node $j$, the choice function, $T_j$, is defined by

$$T_j(I) = \frac{|I \land w_j|}{\alpha + |w_j|},$$

(2)

where the fuzzy AND [7] operator $\land$ is defined by

$$(p \land q)_i \equiv \min (p_i, q_i)$$

(3)

and where the norm $| \cdot |$ is defined by

$$|p| \equiv \sum_{i=1}^{M} |p_i|$$

(4)

for any $M$-dimensional vectors $p$ and $q$. For notational simplicity, $T_j(I)$ in (2) is often written as $T_j$ when the input $I$ is fixed.

The system is said to make a category choice when at most one $F_2$ node can become active at a given time. The category choice is indexed by $J$, where

$$T_J = \max \{T_j : j = 1 \cdots N\}.$$  

(5)
If more than one \( T_j \) is maximal, the category \( j \) with the smallest index is chosen. In particular, nodes become committed in order \( j = 1, 2, 3, \ldots \). When the \( J \)th category is chosen, \( y_J = 1 \); and \( y_j = 0 \) for \( j \neq J \). In a choice system, the \( F_1 \) activity vector \( x \) obeys the equation

\[
x = \begin{cases} 
  I & \text{if } F_2 \text{ is inactive} \\
  I \land w_j & \text{if the } J \text{th } F_2 \text{ node is chosen.}
\end{cases}
\]  

(6)

\textbf{Resonance or Reset: Resonance} occurs if the match function, \( |I \land w_J|/|I| \) of the chosen category meets the vigilance criterion:

\[
\frac{|I \land w_J|}{|I|} \geq \rho;
\]  

(7)

that is, by (6), when the \( J \)th category is chosen, resonance occurs if

\[
|x| = |I \land w_J| \geq \rho|I|.
\]  

(8)

Learning then ensues, as defined below. \textbf{Mismatch reset} occurs if

\[
\frac{|I \land w_J|}{|I|} < \rho;
\]  

(9)

that is, if

\[
|x| = |I \land w_J| < \rho|I|.
\]  

(10)

Then the value of the choice function \( T_J \) is set to 0 for the duration of the input presentation to prevent the persistent selection of the same category during search. A new index \( J \) is then chosen, by (5). The search process continues until the chosen \( J \) satisfies (7).
**Learning:** Once search ends, the weight vector $w_J$ is updated according to the equation

$$w_J^{\text{new}} = \beta \left( I \wedge w_J^{\text{old}} \right) + (1 - \beta) w_J^{\text{old}}. \quad (11)$$

*Fast learning* corresponds to setting $\beta = 1$. The learning law used in the EACH system of Salzberg [11]–[13] is equivalent to (11) in the fast-learn limit with the complement coding option described below.

*Fast-Commit Slow-Recode Option:* For efficient coding of noisy input sets, it is useful to set $\beta = 1$ when $J$ is an uncommitted node, and then to take $\beta < 1$ after the category is committed. Then $w_J^{\text{new}} = I$ the first time category $J$ becomes active. Moore [21] introduced the learning law (11), with fast commitment and slow recoding, to investigate a variety of generalized ART 1 models. Some of these models are similar to fuzzy ART, but none includes the complement coding option. Moore described a category proliferation problem that can occur in certain analog ART systems when a large number of inputs erode the norm of weight vectors. Complement coding solves this problem.
It has been noted that results of ART1 and Fuzzy ART depend critically upon the order in which the training data are processed.

The effect can be reduced to some extent by using a slower learning rate, but is present regardless of the size of the input data set.

In other words, ART1 and Fuzzy ART estimates do not possess the statistical property of consistency.

Critique by Ricardo Gutierrez-Osuna, TAMU (1)

The “stability-plasticity” dilemma
- A term coined by Grossberg that describes the problems endemic to competitive learning
- The network’s adaptability or plasticity causes prior learning to be eroded by exposure to more recent input patterns

ART resolves this problem by creating a new cluster every time an example is very dissimilar from the existing clusters
- **Stability**: previous learning is preserved since the existing clusters are not altered and
- **Plasticity**: the new example is incorporated by creating a new cluster

However, ART lacks a mechanism to avoid overfitting
- It has been shown that, in the presence of noisy data, ART has a tendency to create new clusters continuously, resulting in “category proliferation”
- Notice that ART is very similar to the leader-follower algorithm!
Leader-Follower Clustering
(John A. Hartigan, 1975)

1. Normalize all input patterns
2. Randomly select a pattern \( x^{(n)} \)
   2a Find the winner neuron
   \[ i = \arg\max_j [w_j^T x^{(n)}] \]
   2b If \( ||x^{(n)} - w_i|| < \theta \) (cluster and example are close)
   then update the winner neuron
   \[ w_i = w_i + \eta x^{(n)} \]
   else add a new neuron
   \[ w_{new} = x^{(n)} \]
   2c Normalize the neuron
   \[ w_k = \frac{w_k}{||w_k||} \text{ where } k \in \{i, new\} \]
3. Go to step 2 until no changes occur in \( N_{EX} \) runs

Un fortunately, ART also uses an obscure (biologically-inspired) terminology that clouds its simplicity

- Data are called an "arbitrary sequence of input patterns"
- The current training case stored in "short term memory" and clusters are "long term memory"
- A cluster is called a "maximally compressed pattern recognition code"
- The two stages of finding the nearest cluster are performed by an "Attentional Subsystem" and an "Orienting Subsystem"
  - The latter is said to perform "hypothesis testing", which simply refers to the comparison with the vigilance threshold, not to hypothesis testing in the statistical sense
- "Stable learning" means that the algorithm converges
- The claim that ART is "capable of rapid stable learning of recognition codes in response to arbitrary sequences of input patterns" simply means that ART converges to a solution
  - It does not mean that the clusters are insensitive to the sequence in which the training patterns are presented --quite the opposite is true. Extracted from [comp.ai.neural-nets FAQ]
ART and Other Models

- At some point, Grossberg related ARTMAPs to SOMs.

- There would seem to be a connection between ART and the GNG (Growing Neural Gas) model, as both have methods for introducing new units. See *Constructive Feedforward ART Clustering Networks*, Part II Andrea Baraldi and Ethem Alpaydın, IEEE Trans. Neural Networks, May 2002. (FOSART = Fully Self-Organizing Simplified ART).

- [Survey on Self-Generating Networks](http://osiris.sunderland.ac.uk/~cs0cwi/reports/growing.html) (PhD Proposal)

- What about ART vs. Deep Belief Networks (Hinton)?
1) unlike GNG and SOM, FOSART tries to minimize a quantization (sum-of-squares) error via a soft-to-hard competitive model transition.
2) Unlike Fuzzy ART, the system requires no complement coding of the input data.
3) Unlike SOM and NG, FOSART requires no randomization of the initial template vectors.
4) Unlike SOM and NG, the system requires no a priori knowledge of the size of the network.
5) Unlike SOM, the system requires no a priori knowledge of the topology of the network.
6) Unlike SOM, NG, and Fuzzy ART, FOSART explicitly deals with lateral connections.
7) Unlike GNG, FOSART attempts to address all constraints required to make the CHR guarantee perfect topology-preserving mapping in the sense proposed in [8].
8) Unlike parameters of SOM and NG, FOSART parameters are not affected by outliers which are instead mapped onto noise categories.
9) Unlike Fuzzy ART, the system is capable of removing noise categories to avoid overfitting.
10) Unlike Fuzzy ART, FOSART is competitive with other clustering models found in the literature when the Iris data set is clustered with three reference vectors [51].
## ART Map vs. Backpropagation

<table>
<thead>
<tr>
<th>Feature</th>
<th>Predictive ART</th>
<th>Back Propagation</th>
</tr>
</thead>
<tbody>
<tr>
<td>supervised</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>self-organizing</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>real-time</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>self-stabilizing</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>learning:</td>
<td>fast or slow match</td>
<td>slow mismatch</td>
</tr>
</tbody>
</table>
Extensions and Implementations

- MATLAB Central - File detail - Simplified Fuzzy ARTMAP Neural Network