Machine Learning

Lecture 7

Adaptive resonance theory. Models ART-1, ART-2, Fuzzy-ART, ARTMAP

Adaptive Resonance Theory (ART)

- One of the nice features of human memory is its ability to learn many new things without necessarily forgetting things learned in the past.
- Stephen Grossberg: Stability-Plasticity dilemma
- How can a learning system remain adaptive (plastic) in response to significant input, yet remain stable in response to irrelevant input?
- How does the system known to switch between its plastic and its stable modes?
- How can the system retain previously learned information while continuing to learn new things?

Basic Concept of ART

- A key to solving the stability-plasticity dilemma is to add a feedback mechanism between the competitive layer and the input layer of a network.
- Grossberg and Carpenter: ART model
- ART is one of the unsupervised learning models.
- This kind of model was first established in the early 1960.
- Grossberg introduced the ART in 1976.
- G.A. Carpenter continued the research in ART.
- Now many modifications of ART exist: ART-1, ART-2, FuzzyART, ARTMAP,

ART Based Architectures

- Adaptive Resonance Theory by Grossberg (1976)
- Family of ART neural network architectures
  - unsupervised data classification
    - ART 1 → binary patterns (1987)
    - ART 2 → analog patterns (1987)
    - fuzzy ART → generalization of ART1 in fuzzy set domain
  - supervised mapping
    - ART-MAP
    - fuzzy ART-MAP (1992)

Basic Concept of ART (2)

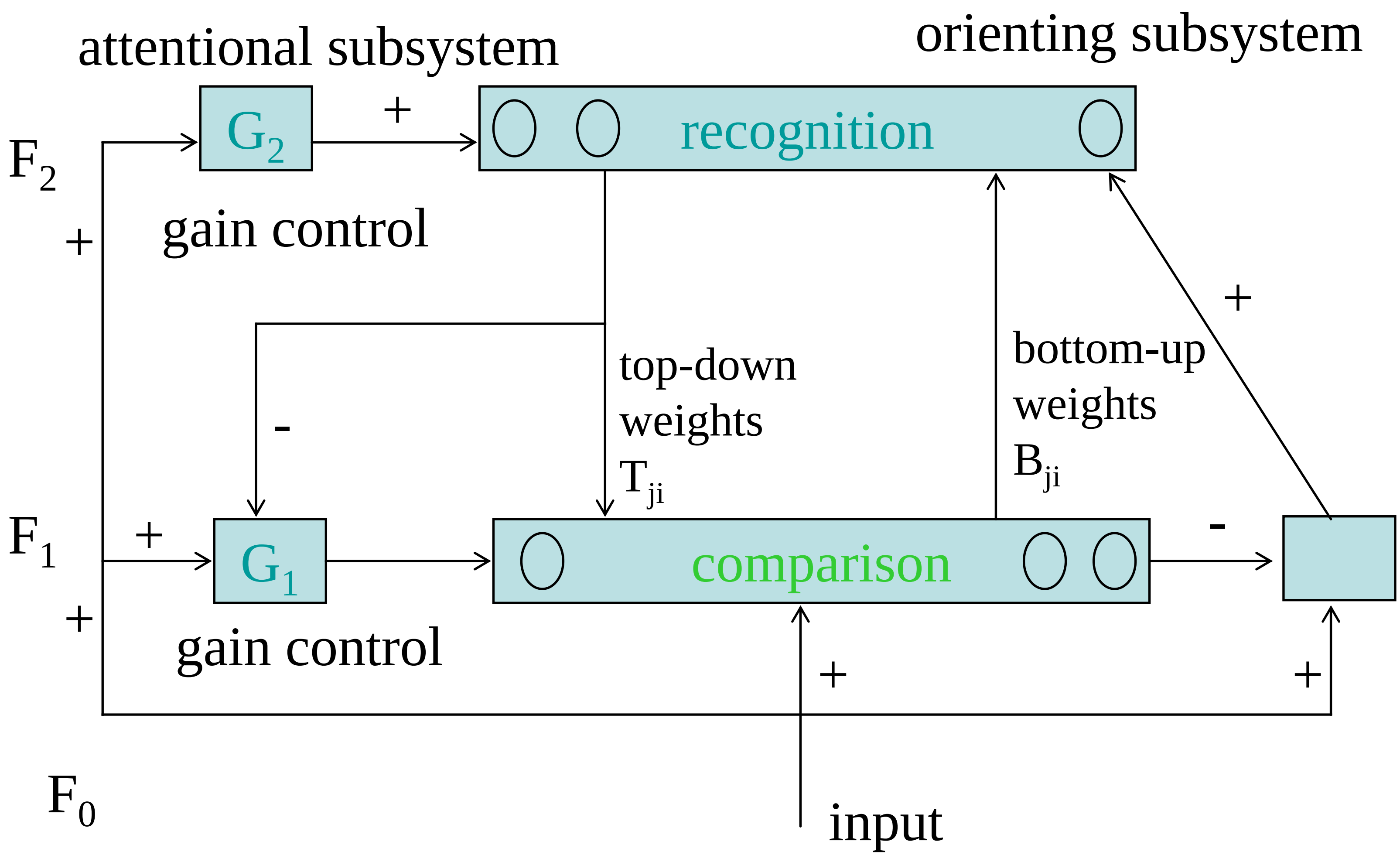
- ART 1: requires that the input vectors be binary
- ART 2: is suitable for processing analog, or gray scale, patterns
- ART gets its name from the particular way in which learning and recall interplay in the network.
- In physics, resonance occurs when a small-amplitude vibration of the proper frequency causes a large-amplitude vibration in an electrical or mechanical system.

Basic Concept of ART (3). Basic algorithm of ART

- Step 1: Initialization. Start with no cluster prototype vectors
- Step 2: Apply new input vector I
- Step 3: Find the closest cluster prototype vector (if any) P
- Step 4: If P is too far from I then, create a new cluster, returning to step 2
- Step 5: Update the matched prototype vector (update P by moving it closer to I)

Basic Concept of ART (4).

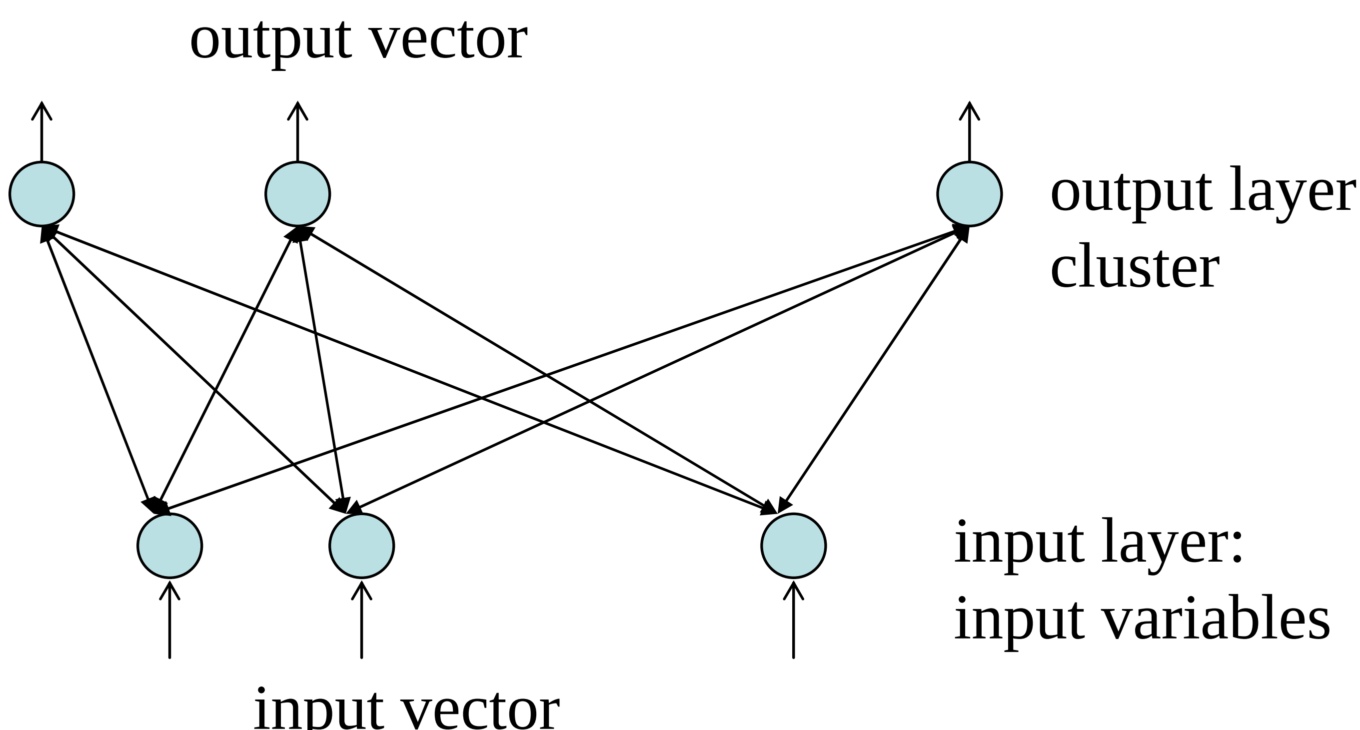
The ART network (Carpenter and Grossberg 1988).



Basic Concept of ART (5)

- Bji: Forward the output from F1 to F2 for competition.
- Tji: Forward the pattern of winner neuron to F1 for comparison.
- G1: To distinguish the feature of input pattern with stored patterns.
- G2: To reset the depressed neurons in F2 (i.e., reset losers).
- attentional subsystem: to rapidly classify the recognized patterns.
- orienting subsystem: to help attentional subsystem learn new patterns.

ART-1 Model


output vector
output layer:
cluster
input layer:
input variables
input vector

ART-1 Model (2)

- Input layer: input patterns or characteristic vectors. Activation function f(x)=x. Inputs are binary values.
- Output layer: representing the clustering of training patterns. This is similar to SOFM except that SOFM has the neighborhood concept. Initially, there is only one output node. The number of output nodes increases when learning proceeds. When the stability is achieved, the learning process stops.

ART-1 Model. Algorithm

- Set the network parameter: Nout=1.
- Set the initial weighting matrices: $w^{t}[i][1] = 1$

$$w^{b}[i][1] = \frac{1}{1+N}$$

- Input the training vector *X*.
- Calculate the matching value: $net[j] = \sum_{i} w^{b}[i][j] \cdot X[i]$

$$I_{count} = 0$$

- Find the max matching value in the output nodes:

$$net[j^{*}] = \max_{j} net[j]$$

ART-1 Model. Algorithm (2)

$$\| X \| = \sum_i X[i]$$

$$\| w^t_{j^*} \cdot X \| = \sum_i w^t[i][j^*] \cdot X[i]$$

$$V_{j^*} = \frac{\| w^t_{j^*} \cdot X \|}{\| X \|}$$

- (6) Calculate the similarity
- value:
-
- (7) Test the similarity value:
- $V < \rho \text{(vigilance)}$ then go to step (8).
- Otherwise go to step (9).
- (8) Test whether there are output nodes applicable to the rule.
- If *Icount*<*Nout*, then try the second max matching value in the output nodes.

ART-1 Model. Algorithm (3)

- Set *Icount*=*Icount*+1; net[j*]=0, go to step (5).
- otherwise
- (a) generate new cluster

$$\text{set } N_{out} = N_{out} + 1$$
$$\text{set new weighting matrix } w$$
$$w^t[\mathrm{i}][\mathrm{N}_{\mathrm{out}}] = \mathrm{x}$$
$$\mathrm{w}^{\mathrm{b}}[\mathrm{i}][\mathrm{N}_{\mathrm{out}}] = \frac{\mathrm{x}}{0.5 + |\,\mathrm{w}^{\mathrm{t}} \cdot X\,|}$$

- (b) set the output values for output nodes:
- if $j=j^*$, then $Y[j]$=1
- else $Y[j]$=0.
- (c) go to step (3) (input new vector X)

ART-1 Model. Algorithm (4)

- (9) Adjust the weighting matrix
- (a) adjust the weights:

$$w^t[\mathrm{i}][\mathrm{j}^*] = w^t[i][j^*] \cdot X[i]$$

$$\mathrm{w}^{\mathrm{b}}[\mathrm{i}][\mathrm{j}^*] = \frac{\mathrm{w}^{\mathrm{t}}[i][j^*] \cdot X[i]}{0.5 + \sum_i w^t[i][j^*] \cdot X[i]}$$

- (b) set the output values for output nodes:
- if *j=j**, then *Y[j]*=1
- else *Y[j]*=0.
- (c) go to step (3) (input new vector *X*)

ART-1 Model. Example.

- Given an input vector: X=[1 1 1 1 1 0 0 0 0 0]
- Assume 5 output nodes. 3 cases for comparisons.
- Case 1:

$$w_1^b = \frac{1}{5.5}[1,1,1,1,1,0,0,0,0,0],\ w_1^t = [1,1,1,1,1,0,0,0,0,0]$$

$$w_2^b = \frac{1}{4.5}[1,1,1,1,0,0,0,0,0,0],\ w_2^t = [1,1,1,1,0,0,0,0,0,0]$$

$$w_3^b = \frac{1}{3.5}[1,1,1,0,0,0,0,0,0,0],\ w_3^t = [1,1,1,0,0,0,0,0,0,0]$$

$$w_4^b = \frac{1}{2.5}[1,1,0,0,0,0,0,0,0,0],\ w_4^t = [1,1,0,0,0,0,0,0,0,0]$$

$$w_5^b = \frac{1}{1.5}[1,0,0,0,0,0,0,0,0,0],\ w_5^t = [1,0,0,0,0,0,0,0,0,0]$$

ART-1 Model. Example (2)

- Node 1: matching value=5/5.5=0.909, similarity value=5/5=1.0.
- Node 2: matching value=4/4.5=0.888, similarity value=4/5=0.8.
- Node 3: matching value=3/3.5=0.857, similarity value=3/5=0.6.
- Node 4: matching value=2/2.5=0.8, similarity value=2/5=0.4.
- Node 5: matching value=1/1.5=0.667, similarity value=1/5=0.2.
- The matching value is proportional to similarity value.

ART-1 Model. Example (3)

- Case 2:
- Assume 6 output nodes.

$$w_1^b = \frac{1}{3.5}[1,1,1,0,0,0,0,0,0,0],\ w_1^t = [1,1,1,0,0,0,0,0,0,0]$$

$$w_2^b = \frac{1}{4.5}[1,1,1,0,0,1,0,0,0,0],\ w_2^t = [1,1,1,0,0,1,0,0,0,0]$$

$$w_3^b = \frac{1}{5.5}[1,1,1,0,0,1,1,0,0,0],\ w_3^t = [1,1,1,0,0,1,1,0,0,0]$$

$$w_4^b = \frac{1}{6.5}[1,1,1,0,0,1,1,1,0,0],\ w_4^t = [1,1,1,0,0,1,1,1,0,0]$$

$$w_5^b = \frac{1}{7.5}[1,1,1,0,0,1,1,1,1,0],\ w_5^t = [1,1,1,0,0,1,1,1,1,0]$$

$$w_6^b = \frac{1}{8.5}[1,1,1,0,0,1,1,1,1,1],\ w_6^t = [1,1,1,0,0,1,1,1,1,1]$$

ART-1 Model. Example (4)

- Node 1: matching value=3/3.5=0.857, similarity value=3/5=0.6.
- Node 2: matching value=3/4.5=0.666, similarity value=3/5=0.6.
- Node 3: matching value=3/5.5=0.545, similarity value=3/5=0.6.
- Node 4: matching value=3/6.5=0.462, similarity value=3/5=0.6.
- Node 5: matching value=3/7.5=0.4, similarity value=3/5=0.6.
- Node 6: matching value=3/8.5=0.353, similarity value=3/5=0.6.

ART-1 Model. Example (5)

- The same similarity value but different matching value.
- If the number of corresponding bits of output vectors to input vector are the same, the one with less ones in output vector will be selected for vigilance test.

ART-1 Model. Example (6)

- Case 3:
- Assume 3 output nodes.

$$w_1^b = \frac{1}{3.5}[1,1,1,0,0,0,0,0,0,0],\ w_1^t = [1,1,1,0,0,0,0,0,0,0]$$

$$w_2^b = \frac{1}{3.5}[0,1,1,1,0,0,0,0,0,0],\ w_2^t = [0,1,1,1,0,0,0,0,0,0]$$

$$w_3^b = \frac{1}{3.5}[0,0,1,1,1,0,0,0,0,0],\ w_3^t = [0,0,1,1,1,0,0,0,0,0]$$

$$w_4^b = \frac{1}{3.5}[0,0,0,1,1,1,0,0,0,0],\ w_4^t = [0,0,0,1,1,1,0,0,0,0]$$

ART-1 Model. Example 7

- Node 1: matching value=3/3.5=0.857, similarity value=3/5=0.6.
- Node 2: matching value=2/3.5=0.571, similarity value=2/5=0.4.
- Node 3: matching value=1/3.5=0.286, similarity value=1/5=0.2.
- Node 4: matching value=0/3.5=0.0, similarity value=0/5=0.0.
- Although the number of 1's in the output vector are the same, the matching value and similarity values are all different. But the matching value is proportional to similarity value.

Continuous-Valued ART (ART-2)

Procedures:

- Given a new training pattern, a MINNET (min net) is adopted to select the winner, which yields the min distance $\| x - w_j \|$.
- Vigilance test: A neuron j* passes the vigilance test if $\| x - w_{j^*} \| < \rho$
- where the vigilance value ρ determines the radius of a cluster.
- If the winner fails the vigilance test, a new neuron unit k is created with weight $w_k = x$.

Continuous-Valued ART (ART-2) (2)

- If the winner passes the vigilance test, adjust the weight of the winner j* by

$$w_{j^*}^{new} = \frac{x + w_{j^*}^{(old)} \| cluster_{j^*}^{(old)} \|}{1 + \| cluster_{j^*}^{(old)} \|}$$

where $\|cluster_i\|$ denotes the number of members in cluster *i*.

Continuous-Valued ART (ART-2) (3)

- Effect of different order of pattern presentation:
  - The ART is sensitive to the presenting order of the input patterns.
- Effect of vigilance thresholds:
  - The smaller vigilance threshold leads to the more clusters are generated.
- Effect of re-clustering:
  - Use the current centroids as the initial reference for clustering.
  - Re-cluster one by one each of the training patterns.
  - Repeat the entire process until there is no change of clustering during one entire sweep.

order	pattern	winner	test value	decision	cluster 1 centroid	cluster 2 centroid	cluster 3 centroid
1	(1.0,0.1)	-	-	new cluster	(1.0,0.1)		
2	(1.3,0.8)	1	1.0	pass test	(1.15,0.45)		
3	(1.4,1.8)	1	1.6	fail⇒ new cluster		(1.4,1.8)	
4	(1.5,0.5)	1	0.4	pass test	(1.27,0.47)		
5	(0.0,1.4)	2	1.8	fail⇒ new cluster			(0.0,1.4)
6	(0.6,1.2)	3	0.8	pass test			(0.3,1.3)
7	(1.5,1.9)	2	0.2	pass test		(1.45,1.85)	
8	(0.7,0.4)	1	0.63	pass test	(1.13,0.45)		
9	(1.9,1.4)	2	0.9	pass test		(1.6,1.7)	
10	(1.5,1.3)	2	0.5	pass test		(1.58,1.6)	

(a) Original order.

The execution sequence of the ART-2 with the vigilance threshold 1.5.

order	pattern	winner	test value	decision	cluster 1 centroid	cluster 2 centroid	
1	(1.5,1.3)	-	-	new cluster	(1.5,1.3)		
2	(1.9,1.4)	1	0.5	pass test	(1.7,1.35)		
3	(0.7,0.4)	1	1.95	fail$\Rightarrow$ new cluster		(0.7,0.4)	
4	(1.5,1.9)	1	0.75	pass test	(1.63,1.53)		
5	(0.6,1.2)	2	0.9	pass test		(0.65,0.8)	
6	(0.0,1.4)	2	1.25	pass test		(0.43,1.0)	
7	(1.5,0.5)	1	1.17	pass test	(1.6,1.28)		
8	(1.4,1.8)	1	0.72	pass test	(1.56,1.38)		
9	(1.3,0.8)	1	0.84	pass test	(1.52,1.28)		
10	(1.0,0.1)	2	1.47	pass test		(0.58,0.78)	

(b) Reverse order

Implementation of algorithm of ART-2

```
for (int j = 0; j < Net.NOut; j++)
    {
      Net.S2[j]=0;
      for (int i = 0; i < Net.NR; i++)
        Net.S2[j]=Net.S2[j]+(Net.S1[i]-Net.w[i][j])*(Net.S1[i]-Net.w[i][j]);
      Net.S2[j]=sqrt(Net.S2[j]);
      Temp=Net.S2[j];
    };
  jmin=0;
  j=0;
  S2Min=9999999999;
  Temp=Net.S2[1];
  while (j<Net.NOut)
   {
     if (Net.S2[j]<S2Min)
       {
         S2Min=Net.S2[j];
         jmin=j;
       };
     j=j++;
   };
```

Implementation of algorithm of ART-2 (2)

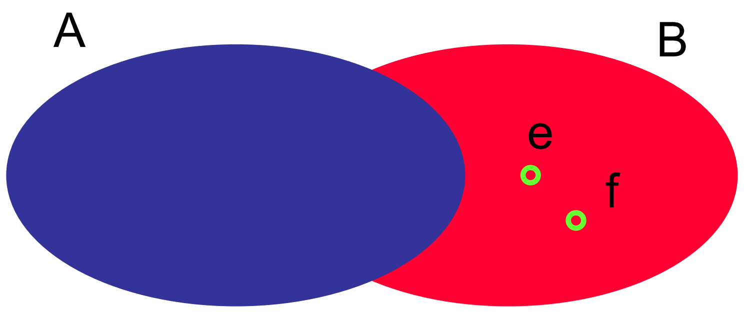
```
    if (S2Min > Net.r)   // test to recognizing is not OK
     {
//  forming of new output neuron
       Net.NOut++;
       Net.S2[Net.NOut-1]=0;
       Net.Number[Net.NOut-1]=1;
       for (int i = 0; i < Net.NR; i++)
        {
          Net.w[i][Net.NOut-1]=Net.S1[i];
          Net.S2[Net.NOut-1]=Net.S2[Net.NOut-1]+(Net.S1[i]-
Net.w[i][Net.NOut-1])*(Net.S1[i]-Net.w[i][Net.NOut-1]);
          jmin=Net.NOut-1;
        };
       Net.S2[Net.NOut-1]=sqrt(Net.S2[Net.NOut-1]);
     }
     else
```

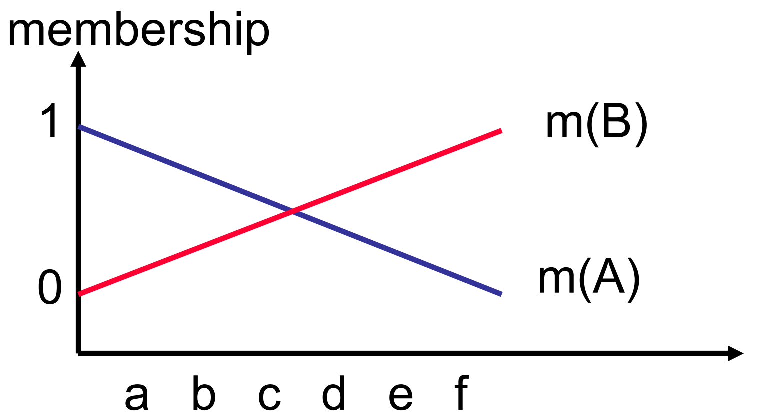
Implementation of algorithm of ART-2 (3)

```
Net.Number[jmin]++;
     for (int i = 0; i < Net.NR; i++)
      {
       Net.w[i][jmin]=Net.w[i][jmin]+(Net.S1[i]-
Net.w[i][jmin])/(1+Net.Number[jmin]);
      };
     Form1->Memo1->Clear();
     for (int i = 0; i < Net.NOut; i++)
       Form1->Memo1->Lines->Add(FloatToStr(Net.S2[i]));
   Form1->NOut->Text=IntToStr(Net.NOut);
   Form1->Edit1->Text=IntToStr(jmin+1);
   return(jmin);
   }
```

Fuzzy Logic – the concept

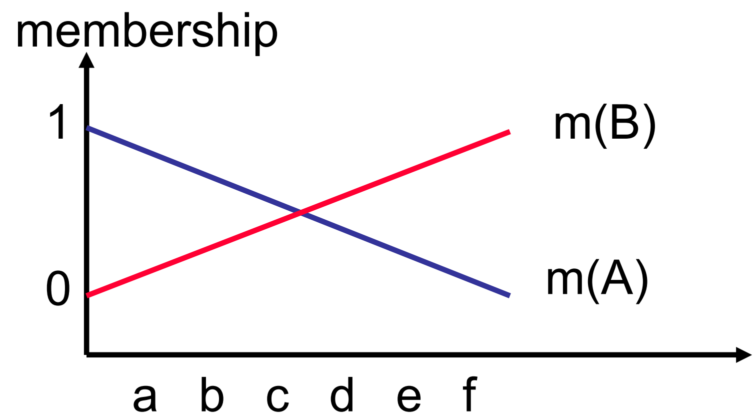
- Fuzzy logic extends this membership concept from binary to analog: That is the membership of an element to set can have any value between 0 and 1.
- It then re-defines the classic set operators, like $\cap$ $\cup$ $\supset$ $\supseteq$ $\not\subset$ $\subset$ such that they boil down to the regular set operators for binary membership functions.

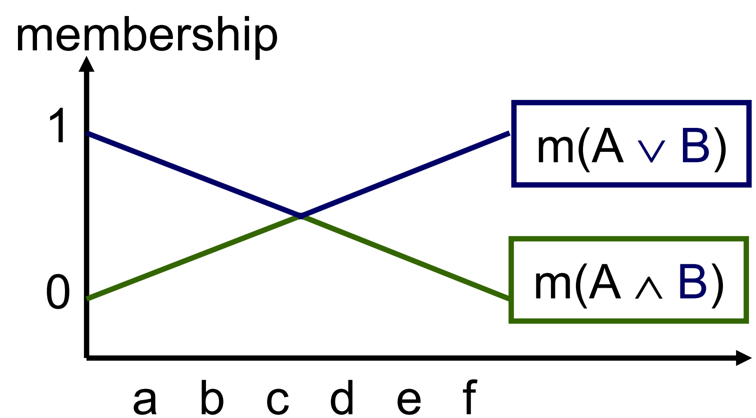




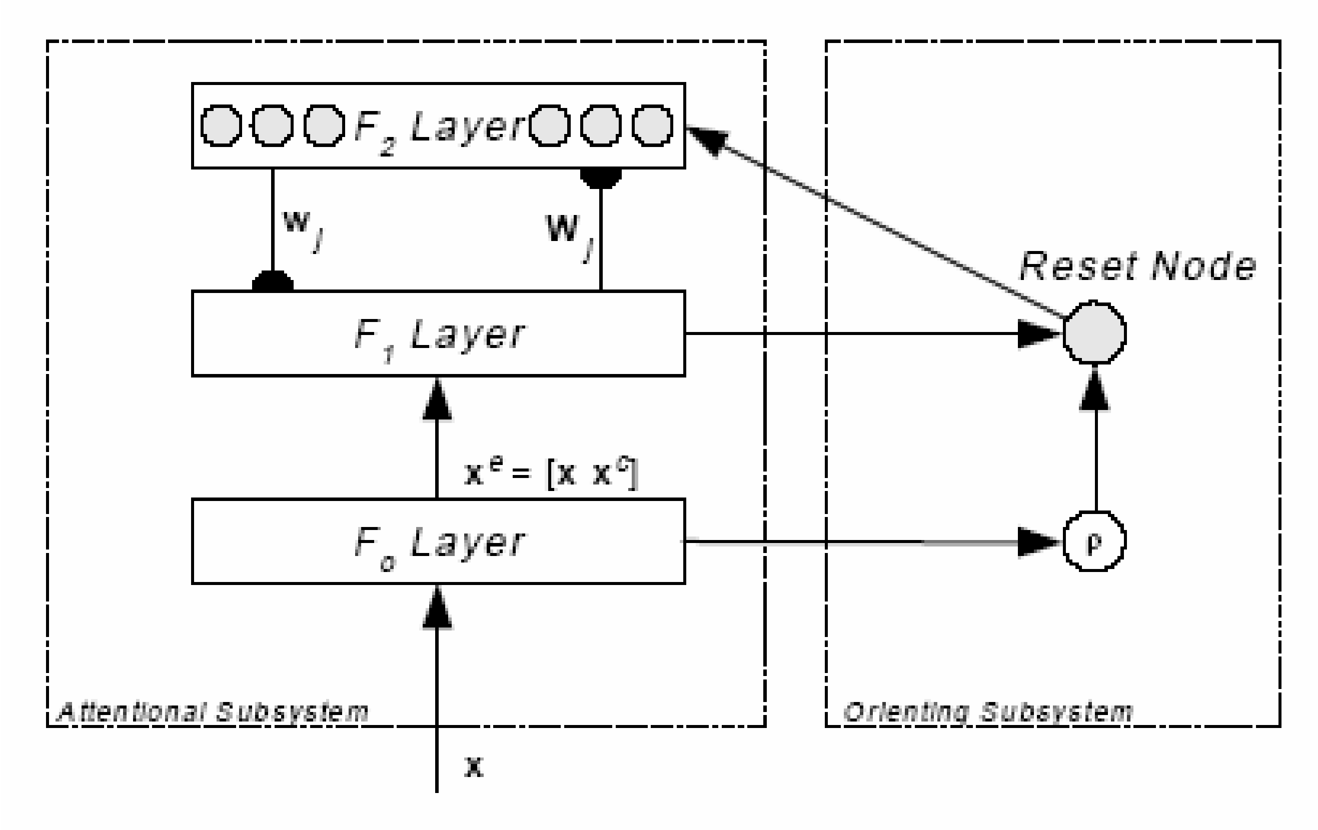
Fuzzy Logic - operators

- Intersection: $\wedge$ means $\cap$
- Union: $\vee$ means $\cup$
- The membership of an element is defined as
- $m(A \wedge B) = \min(m(A), m(B))$
- $m(A \vee B) = \max(m(A), m(B))$





Structure of Fuzzy ART



Structure of Fuzzy ART (2)

- A preprocessing field of nodes, F0 modifies (normalize and complimentary code) the current input vector
- A field F1 receives both bottom-up input from F0 and top-down input from the field F2
- We do not need to distinguish between the connection weights of the top-down feedback paths and the bottom-up feedforward paths between the fields F1 and F2 in the Fuzzy ART module, both will be implemented by the same weights
- Three parameters determine the dynamics of a Fuzzy ART network
  - a choice parameter $\alpha > 0$;
  - a learning rate parameter $\beta = [0, 1]$
  - a vigilance parameter $\rho = [0, 1]$

Algorithm of Fuzzy ART

1. *Initialize*

   (a) Initially, each cluster is said to be *uncommitted*, after a category is selected for coding it becomes *committed*, and the weight vector W_j is set as $W_{j1}(0) = W_{j2}(0) = \ldots = W_{jm}(0) = 1$

   (b) Then, a choice parameter α, a learning rate β, and a vigilance parameter ρ are set: $\alpha > 0, \beta \in [0, 1], \rho \in [0, 1]$.

2. *Complement Coding*

   (a) To improve the reliability of category choice, input a is expanded with complement. $I = (a, a^c)$, $a^c = 1 - a$.

3. *Category choice*

   (a) For each I and cluster (F_2 node) j, the choice function T_j is defined by $T_j(I) = \frac{|I \wedge W_j|}{\alpha + |W_j|}$, where the fuzzy AND operator $\wedge$ is defined by $(x \wedge y)_i = min(x_i, y_i)$ and $|\ |$ is defined by $|x| = \sum_{i=1}^{m} |x_i|$

   (b) The system makes a cluster selection when more than one cluster could be selected at a given time. The index J denotes the chosen cluster, where

$$T_J = max\{T_j : j = 1, \ldots, n\} \qquad (2)$$

Algorithm of Fuzzy ART (2)

If more than one T_J is maximal, the system chooses the category with the smallest j index. Nodes become *committed* in order $j = 1, 2, 3, \ldots$.

4. *Resonance or Reset*

(a) The resonance occurs if the match function of the chosen cluster meets the vigilance criterion; where

$$\frac{|I \wedge W_J|}{|I|} \geq \rho \tag{3}$$

The learning processes is done according to the equation

$$W_J^{(new)} = \beta(I \wedge W_J) + (1 - \beta)W_J^{(old)}. \tag{4}$$

Fast learning corresponds to $\beta = 1$, which is the learning rule in Eq. 4

$$W_J^{(new)} = \beta(I \wedge W_J) \tag{5}$$

(b) *Mismatch reset* occurs if $\frac{|I \wedge W_J|}{|I|} < \rho$.

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 selection during search. A new index J is chosen by (2). The search continues until the chosen J satisfies (3). If no one of the clusters is selected, then a new cluster must be incremented in $F2$ field.

Algorithm of Fuzzy ART (3)

If without vectors:

The fuzzy ART learning rule is given by

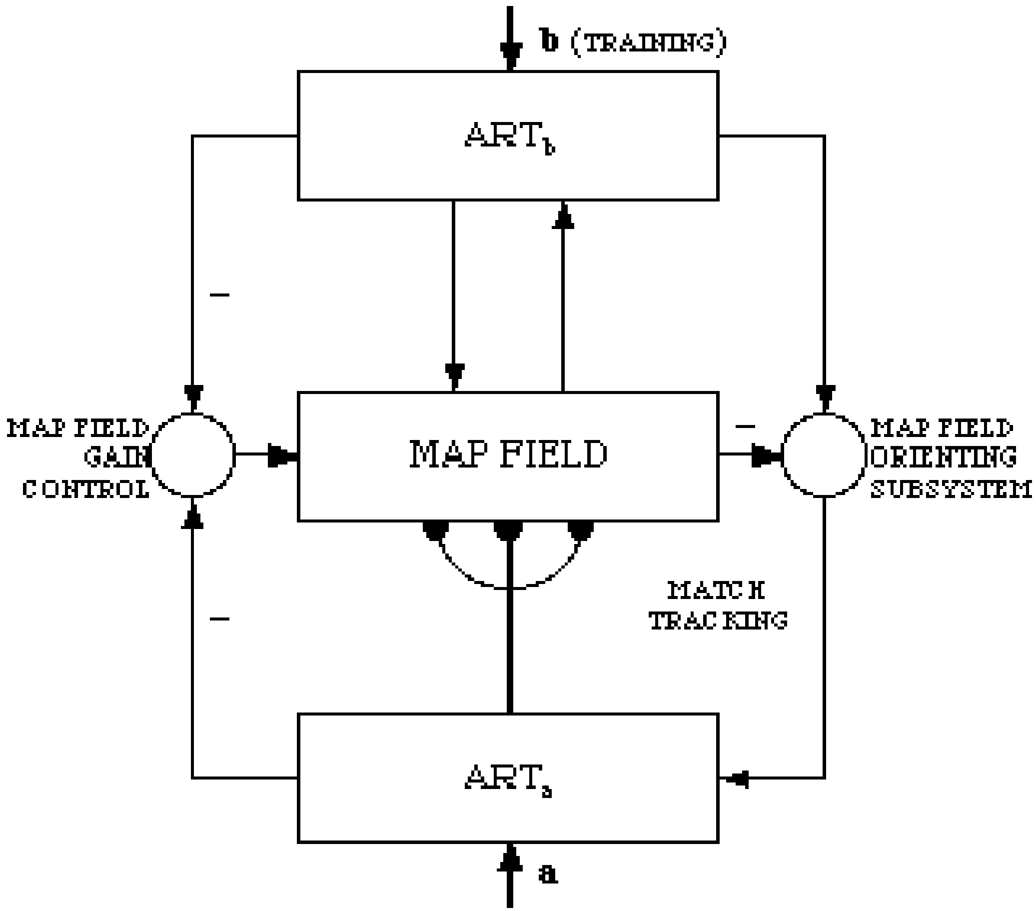
$$w_{Ji}^{new} = \begin{cases} w_{Ji}^{old} & w_{Ji} \le I_i \\ w_{Ji}^{old} - \beta\left(w_{Ji}^{old} - I_i\right) & w_{Ji} > I_i \end{cases},$$

where $0 < \beta \le 1$.

ARTMAP and Fuzzy ARTMAP

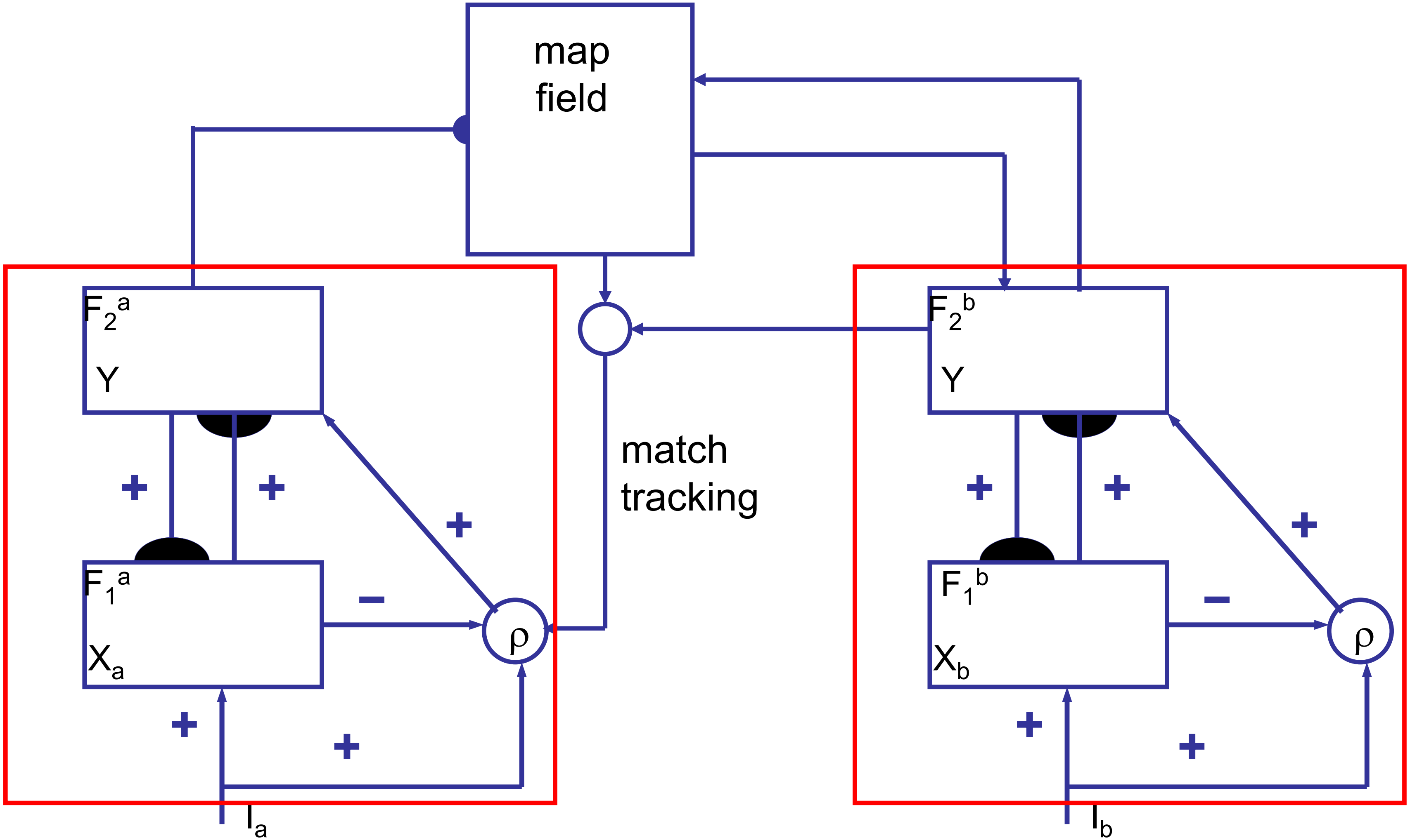
- A supervised learning system constructed from two unsupervised ART networks.
- There are two variations:
  - ARTMAP which works with binary data
  - Fuzzy ARTMAP which works with continuous data
- The ART or Fuzzy ART modules cluster the input and output patterns.

ARTMAP

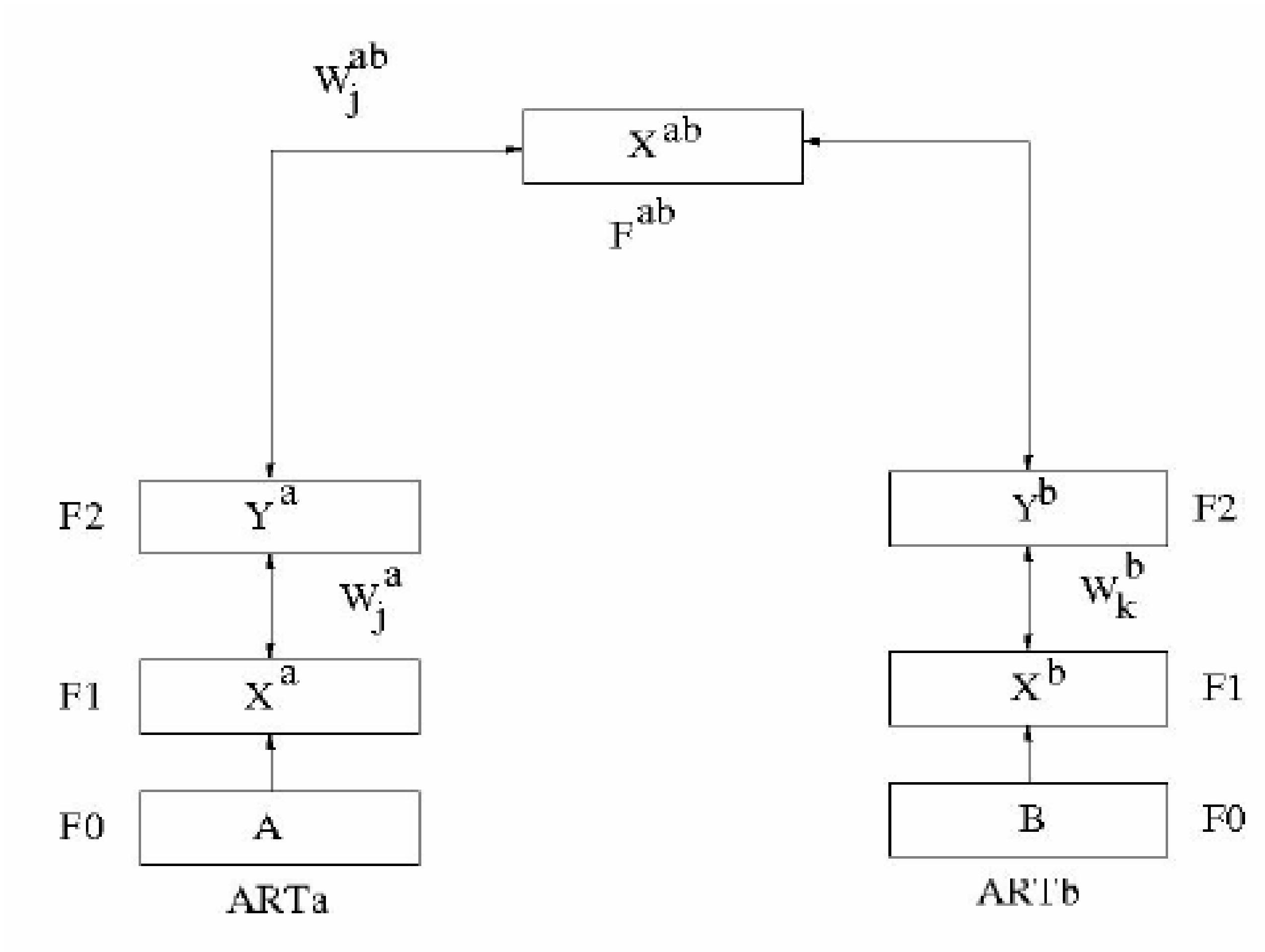


b (TRAINING)
ART$_b$
MAP FIELD GAIN CONTROL
MAP FIELD
MAP FIELD ORIENTING SUBSYSTEM
MATCH TRACKING
ART$_a$
a

Fuzzy ARTMAP architecture



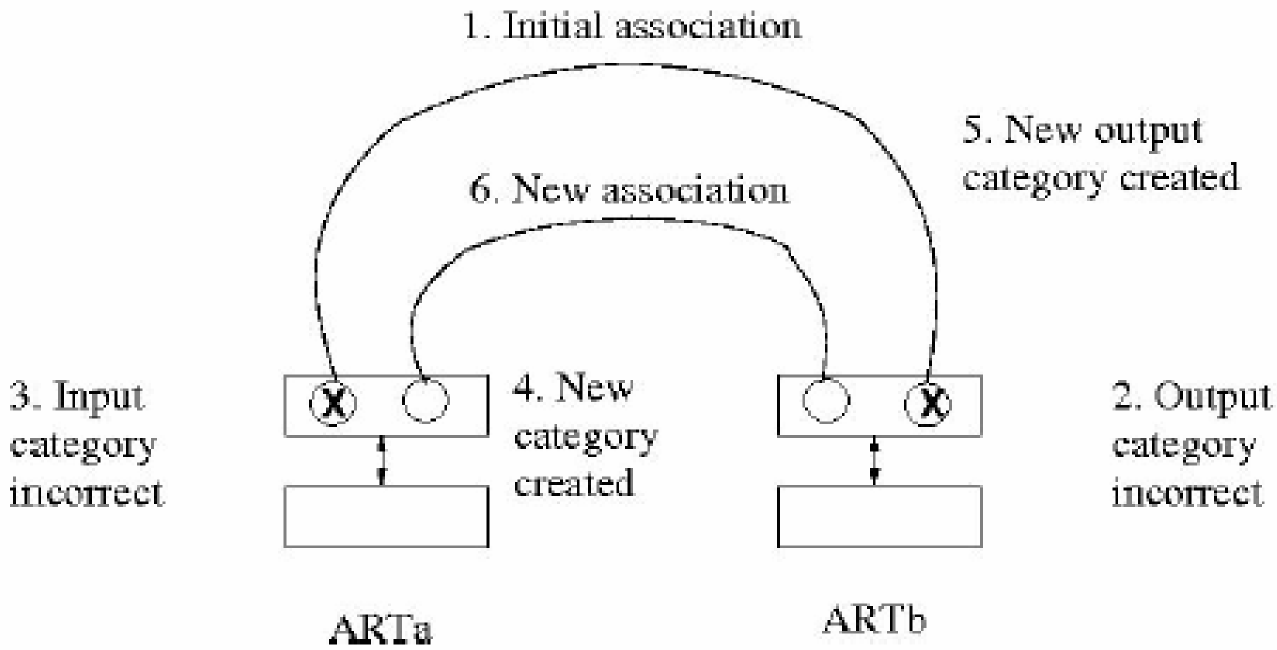
Fuzzy ARTMAP Architecture (2)



Fuzzy ARTMAP

- It uses a minimax learning rule which minimizes error and maximizes code compression (the number of patterns stored in hidden units).
- The vigilance parameter p_a in ART_a is modified to control error. If a new category is needed the p_a is increased to force its creation.
- Lower values of p_a allow larger categories to form (greater compression) and when these are not appropriate then p_a is increased. The system modifies its parameters to reduce error.
- If ART_b does not predict the appropriate output for a given input to ART_a (predictive failure) then p_a is increased by enough to force a new category to be created in ART_a. This operation is named **match tracking**.

Match tracking



The first association is not correct so a new category is forced in ART_a which will create a new association in ART_b.

Algorithm

- Initially the weights from ART_a to the associative memory are set equal to 1.0.
- When either ART_a or ART_b becomes active then the map field also becomes active which creates an association in the other ART network.
- If F_a becomes active, it is filtered through the weight layer w_J^{ab} and F^{ab} activation is derived from these weight values.
- If F_b becomes active then the activation values are **copied** to F^{ab}. In this case the activation values in both layers are identical. F^{ab} and F_2^b must contain the same number of nodes.
- If bother F_2^a and F_2^b are active during training then F^{ab} reflects the correctness of the association between F_a and F_b.

Algorithm (2)

$$x^{ab} = \begin{cases} y^b \wedge w_J^{ab} & \text{f } J^{th} F_2^a \text{ is active, } F_2^b \text{ is active} \\ w_J^{ab} & \text{f } J^{th} F_2^a \text{ is active, } F_2^b \text{ is inactive} \\ y^b & \text{f } F_2^a \text{ is inactive, } F_2^b \text{ is active} \\ 0 & \text{if both } F_2^a \text{ and } F_2^b \text{ are inactive} \end{cases}$$

- Once inputs to both F^a and F^b are present and activation in F^{ab} has occurred then the correctness of the association is compared.
- Match tracking occurs when $|x^{ab}| < P_{ab}$ where p_{ab} is the map field F^{ab} vigilance parameter.

- **Match tracking** and **fast learning** allow the network to learn rare events which occur within a collection of frequent events which lead to different predictions.
- The network forces a category to be created for unique events so they can be identified as different.

Association

- Categories in either F_2^a or F_2^b can be used to activate the associated category in the other layer.
- All associations between the two networks is a simple set of weights with values of either 1.0 or 0.0 which are used to switch the classification on and off.

Functionality

- **In the input layer F0**, the input vector a is normalised by means of complementary code, to yield I = (a;a'),
- where a' = 1 - a.
- **In the choice layer F2,** the input pattern is compared to existing templates, calculating for each template j,
a choice function Tj = |Wj & I| / (alpha + |Wj|)
- where **|.|** denotes the **L1 norm**, and **&** is the **fuzzy intersection** *or* **min operator**.

- Since the category J with highest Tj is selected to win the competition, the **choice parameter alpha** is used to favour the selection of small categories over large (general) categories.

Functionality (2)

- **In the matching layer F1**, Once a template wins, its degree of matching to the pattern needs to be sufficient to meet the match condition in F1, given by
- **|Wj & I|/|I|>=ro**,
- where **ro** is the **vigilance parameter**.

- If this condition is not met, category J is inhibited and another template is selected, or a new one committed

- Once a category J has been selected in ART-a that meets the vigilance criterion and predicts the correct output class K selected in ART-b, **the network enters resonance**, and weights are updated, by the formula

$$w_j^{new} = \beta(w_j^{old} \& I) + (1+\beta)w_j^{old}$$

Functionality (3)

- Because Fuzzy ARTMAP uses **complementary coding**, and **fuzzy intersection &,** if we denote Wj = (Uj ; Vj'), then weights Wj represent a **hyperbox Rj** in the input space, where Uj and Vj are the lower and upper corners, respectively

- Each **hypebox Rj** contains all the training patterns that selected category j during learning, and is used as the descriptive element associated to a category.

Functionality (4)

- When a category learns a pattern, the **choice function** determines the winner category, the **vigilance condition** sets an upper limit on the hyperbox size.

- Since in Fuzzy ARTMAP the training is supervised, the **match tracking mechanism** ensure that, for a given input sample, the category that resonates has a better match, so that if the pattern is presented again this category will be selected.

Functionality (5)

- On the contrary, if for example a pattern **a** is presented and category **R** is selected, but their associated labels differ, the match tracking mechanism will create a new category. This category will be selected next time **a** is presented and the prediction would be correct.

- **During the training, the number of categories created is usually large, forming the problem of Category propagation, which is a serious limitations of the Fuzzy ARTMAP.**

- The presence of **noises** in the training Set augments the category propagation, since noises complicates relations among input and output data, and may introduce contradictory training pattern.

Fuzzy ART-MAP

- solves stability-plasticity dilemma (ART)
  - self-stabilizing memory
  - vs.
  - continual learning of novelties
- automatic feature selection via ‘expectancy’ (ART)
  - detect selient features for each class seperately
  - possible to trace the weights
- many-to-one mapping
  - unsupervised clustering in feature space
  - supervised mapping to labels
  - featurally unrelated clusters can get same labels
- adjust the scale of generalization automatically (vigilance)
- fast learning in relatively less number of epochs