

神经网络

Neural Networks

第四章

自组织网络

— 自适应谐振理论

史忠植

中国科学院计算技术研究所
<http://www.intsci.ac.cn/>

内容提要

1. ART的结构

2. ART的初始化

3. ART的实现

识别、比较、查找、训练

4. ART2

自适应谐振理论

- 自适应共振理论ART(Adaptive Resonance Theory)模型是美国Boston大学的S. Grossberg和A. Carpenter在1976年提出的。
- ART是一种自组织神经网络结构，是无教师的学习网络。当在神经网络和环境有交互作用时，对环境信息的编码会自发地在神经网络中产生，则认为神经网络在进行自组织活动。ART就是这样一种能自组织地产生对环境认识编码的神经网络理论模型。网络的可塑性需要的4项功能

自适应谐振理论

Grossberg一直对人类的心理和认识活动感兴趣，他长期埋头于这方面的研究并希望用数学来刻画人类这项活动，建立人类的心理和认知活动的一种统一的数学模型和理论。

ART模型是基于下列问题的求解而提出的：

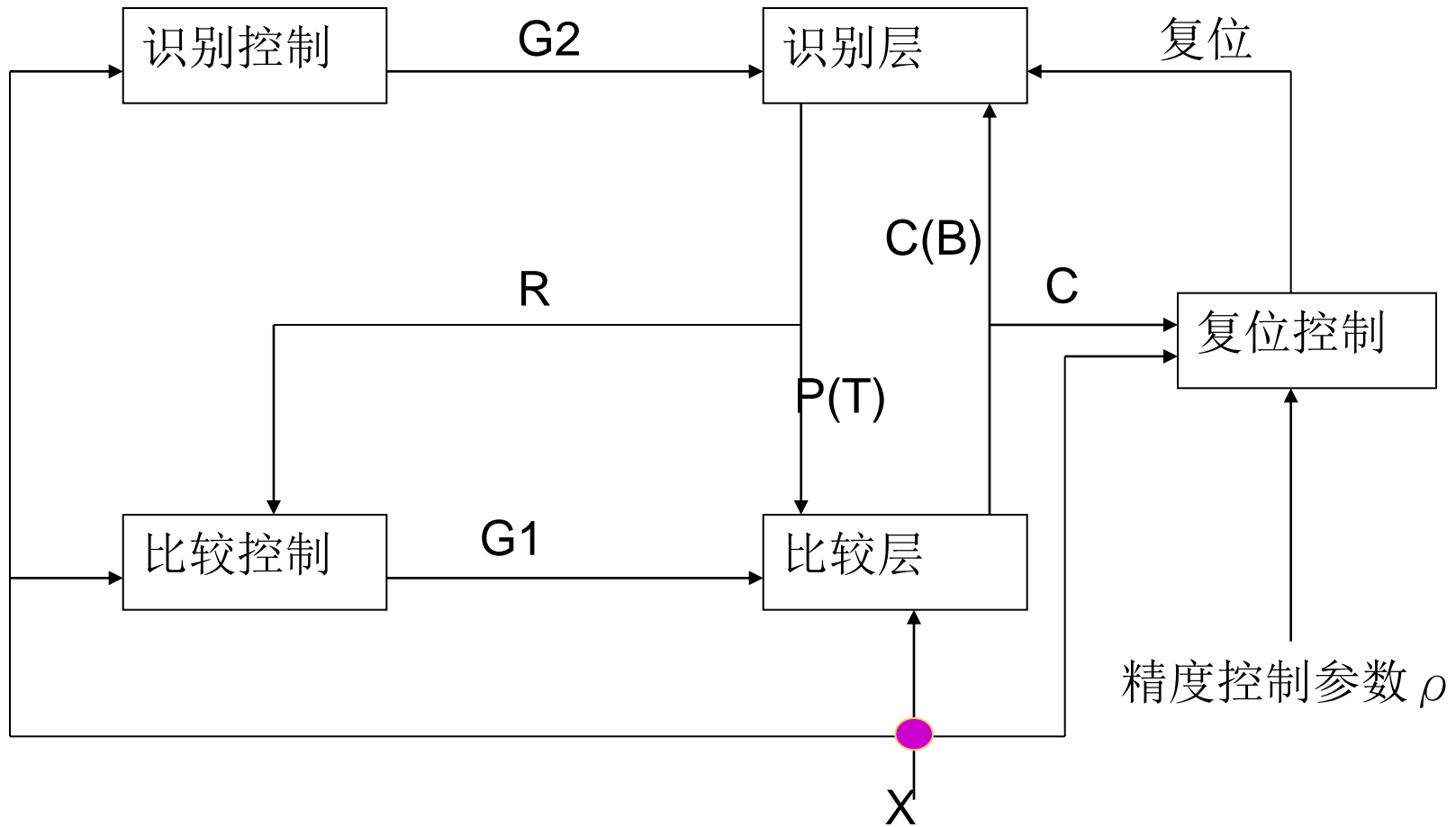
- 1. 对于一个学习系统，要求它有适应性及稳定性，适应性可以响应重要事件，稳定性可以存储重要事件。这种系统的设计问题。
- 2. 学习时，原有的信息和新信息如何处理，保留有用知识，接纳新知识的关系如何及解决的问题。
- 3. 对外界信息与原存储的信息结合并决策的问题。

自适应谐振理论

ART理论已提出了三种模型结构，即ARTt, ART2, ART3。

- **ART1**用于处理二进制输入的信息；
- **ART2**用于处理二进制和模拟信息这两种输入；
- **ART3**用于进行分级搜索。
- **ART理论**可以用于语音、视觉、嗅觉和字符识别等领域。

ART总体结构图



ART的结构

$$\mathbf{X}=(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$$

$$\mathbf{R}=(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m)$$

$$\mathbf{C}=(\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n)$$

$$\mathbf{P}=(\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n)$$

$$\mathbf{T}_i=(\mathbf{t}_{i1}, \mathbf{t}_{i2}, \dots, \mathbf{t}_{in})$$

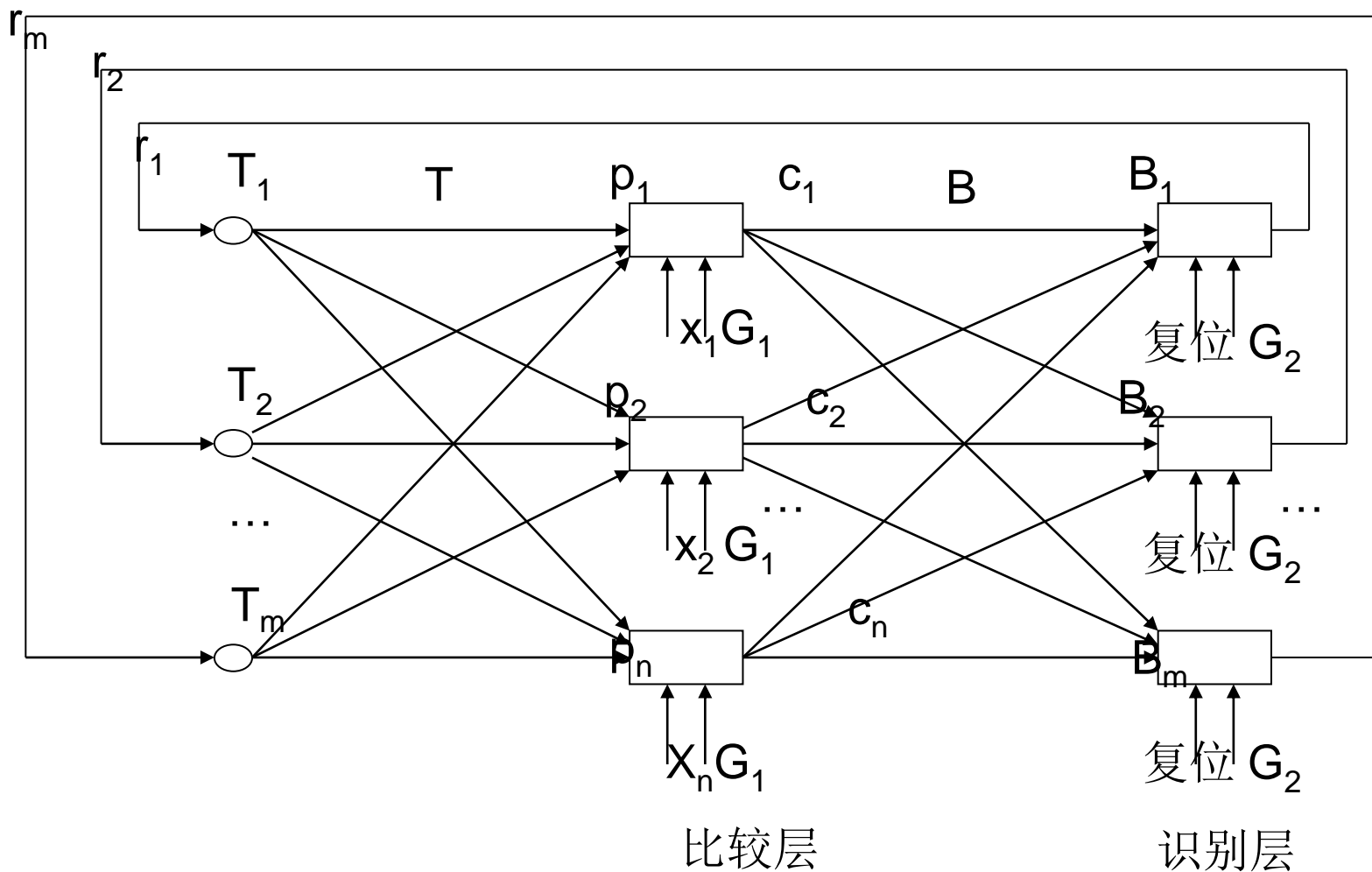
$$\mathbf{B}_i=(\mathbf{b}_{1i}, \mathbf{b}_{2i}, \dots, \mathbf{b}_{ni})$$

ART的结构

- t_{ij} 表示识别层的第*i*个神经元到比较层的第*j*个神经元的联接权
- b_{ij} 表示比较层的第*i*个神经元到识别层的第*j*个神经元的联接权
- p_i 为比较层的第*i*个神经元的网络输入

$$p_i = \sum_{j=1}^m r_j t_{ji}$$

比较层和识别层



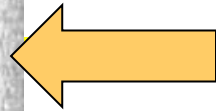
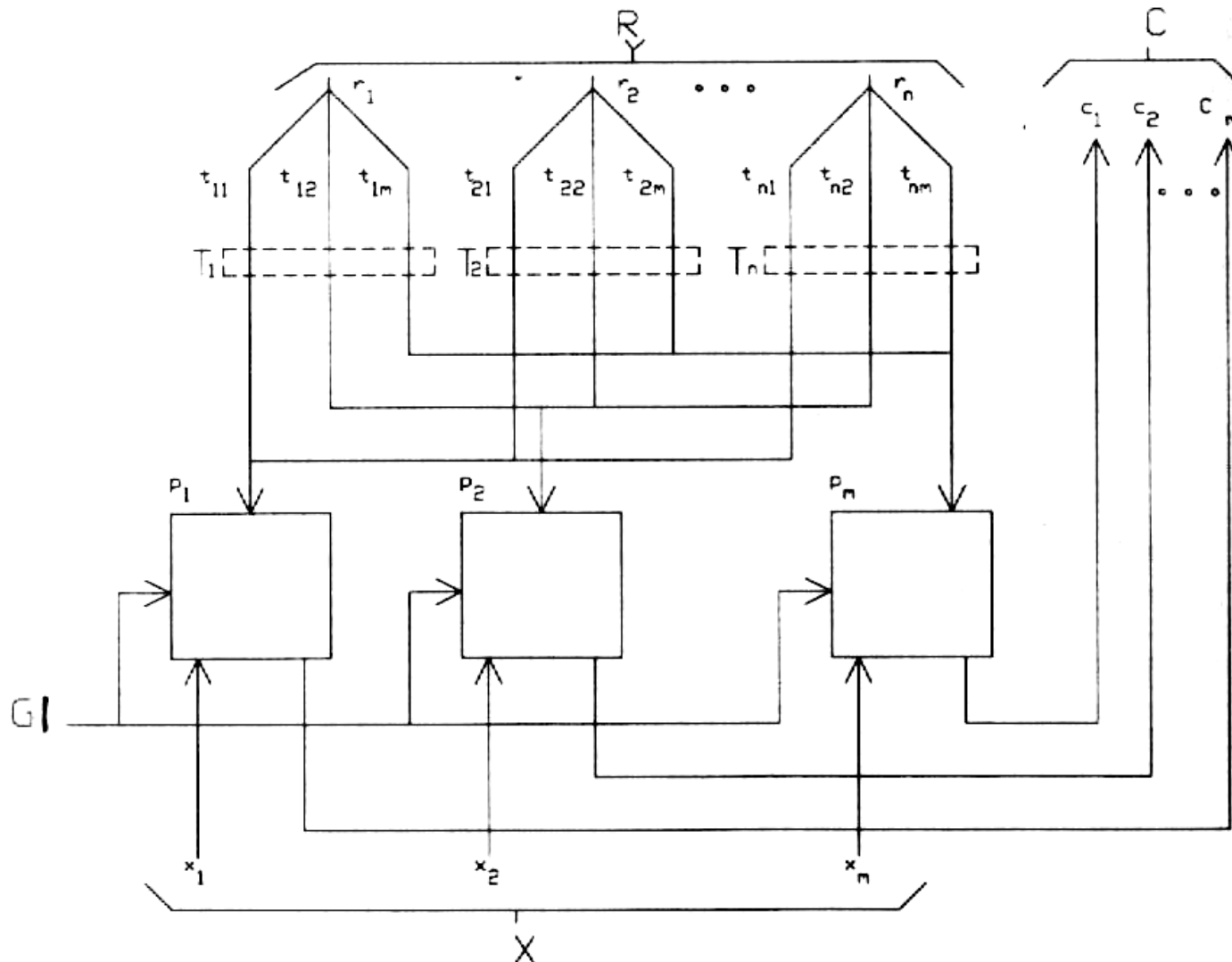
ART Architecture

Functions of each models:

Comparison layer:

- accept binary input \underline{X}
- initially pass \underline{X} to \underline{C} . So $\underline{C} = \underline{X}$
- binary vector \underline{R} is produced from recognition layer to modify \underline{C}

ART Architecture



ART Architecture

-each neuron in the comparison layer has 3 inputs:

\underline{X} : input vector

\underline{P}_j : weighted sum of recognition layer output

Gain 1: same signal to all neurons

-use “two-third” rule

=> at least two of a neuron’s three inputs must be one, otherwise, the output is zero.

-initially, *Gain 1* is set to one and \underline{R} are set to 0.

ART Architecture

Recognition layer:

- compute **dot product** of B and C
- the neuron with largest output wins
- the winning neuron is set to one others
are set to zero.

Gain 2

- OR** of components of X

ART Architecture

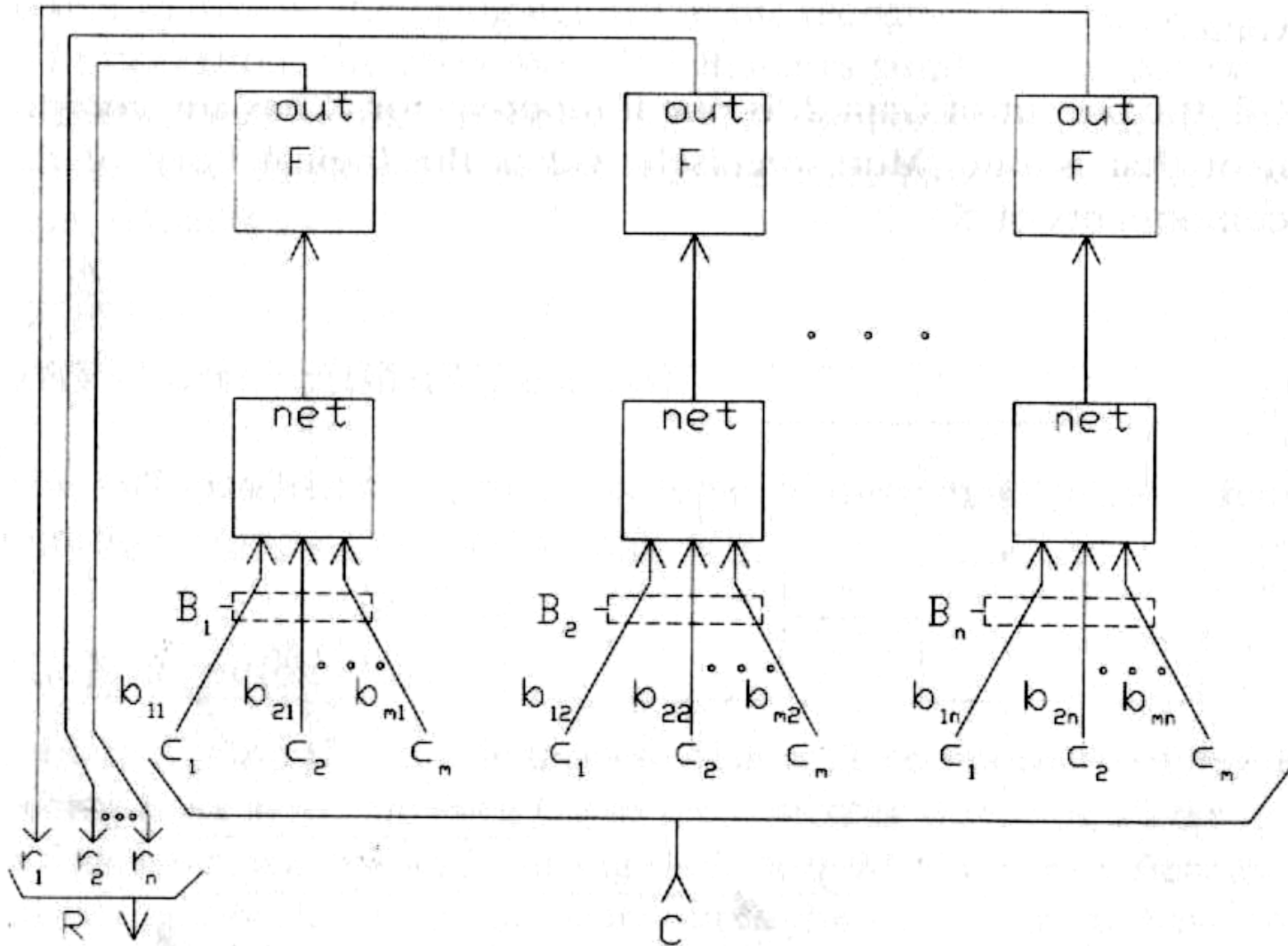
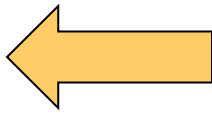


Figure 8-3. Simplified Recognition Layer

ART Architecture

Gain 1

OR of X

components G2 OR of R G1

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 |

比较层输出信号控制

$$G1 = \neg (r_1 \vee r_2 \vee \dots \vee r_m) \wedge (x_1 \vee x_2 \vee \dots \vee x_n)$$

识别层输出信号控制

$$G2 = x_1 \vee x_2 \vee \dots \vee x_n$$

比较层

- 执行2/3规则

$$c_i = 1$$

$$x_i + p_i + G1 \geq 2$$

$$c_i = 0$$

$$x_i + p_i + G1 < 2$$

- 待命期

- 工作周期

$$p_i = \sum_{j=1}^m r_j t_{ji}$$

$$C=X$$

$$= r_k t_{ki}$$

$$P=T_k$$

$$= t_{ki}$$

$$c_i = x_i \wedge p_i$$

识别层

- 识别层实现竞争机制

- \mathbf{B}_k 与 \mathbf{C} 有最大的点积

$$\sum_{i=1}^n b_{ik} c_i = \max \left\{ \sum_{i=1}^n b_{ij} c_i \mid 1 \leq j \leq m \right\}$$

- X 的“暂定”代表 \mathbf{RN}_k 所获得的网络输入为

$$\sum_{i=1}^n b_{ik} c_i$$

与 $\mathbf{RN}_1, \mathbf{RN}_2, \dots, \mathbf{RN}_m$ 相对应

向量 $\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_m$ 代表不同分类

系统复位控制

X与C的相似度

$$S = \frac{\sum_{i=1}^n c_i}{\sum_{i=1}^n x_i}$$

$s \geq \rho$ ，当前处于激发态的 RN_k 所对应的 B_k 、 T_k 为X的类表示；

$s < \rho$ ，此 RN_k 所对应的 B_k 、 T_k 不能很好地代表X，需要重新寻找

ART的初始化

- **T的初始化**
 - 矩阵T的所有元素全为1
- **B的初始化**
 - $b_{ij} < L / (L - 1 + n)$
 - n 为输入向量的维数； L 为一个大于1的常数，其值应该与输入向量的位数相关
 - T_k 、 B_k 是 RN_k 对应类的两种不同表示
- **ρ 的初始化**
 - $\rho \in [0, 1]$

ART的实现

- 四个阶段：识别、比较、查找、训练
- 一、识别
 - \mathbf{X} (非0向量) 未被加在网上时
 - $G_2=0$
 - $R=(r_1, r_2, \dots, r_m)=(0, 0, \dots, 0)$
 - \mathbf{X} (非0向量) 被加在网络上时
 - $G_1=G_2=1$
 - $R=0$ 导致 $P=(p_1, p_2, \dots, p_m)=(0, 0, \dots, 0)$

ART的实现

- 在识别层，每个 RN_k 完成的操作
 - 计算 $\sum b_{ik}c_i$
 - 接收来自其它RN的抑制信号，并向其它的RN发出抑制信号
 - 确定自己的输出状态
 - 完成输出
- RN之间的抑制连接与抑制信号
- 如果 RN_k 输出1，则表明，在本轮识别中， X 暂时被认为是属于该 RN_k 所对应的类

比较

- X 归于 RN_k ， RN_k 的输出值1被分别以权重 t_{kj} 传送到比较层
- 向量 P 就是向量 T_k
- T 的初始化及训练保证了 T 的每个元素取值为0或者1
- B_k 与 T_k 根据 RN_k 进行对应，互为变换形式
- 如果对于所有的 j ， $1 \leq j \leq n$ ， $p_j = x_j$ ，则表示 X 获得良好的匹配。如果存在 j ，使得 $p_j \neq x_j$ ，则表明 X 与相应的“类”的代表向量并不完全一致

比较

- 当系统复位控制模块计算X和C的相似度s
- 如果 $s \geq \rho$ ，表明本轮所给出的类满足精度要求。查找成功，系统进入训练周期
- 如果 $s < \rho$ ，表明本轮所给类不满足精度要求。
 - 复位模块要求识别层复位，使所有RN输出0
 - 系统回到开始处理X的初态，重新进行搜索
 - 复位信号屏蔽本次被激发的RN，在下一轮匹配中，该RN被排除在外，以便系统能够找到其它更恰当的RN

查找

- 如果 $s \geq \rho$ ，认为网络查找成功，此时分类完成，无需再查找
- 如果 $s < \rho$ ，表明本轮实现的匹配不能满足要求，此时需要寻找新的匹配向量
- 查找过程

查找

- 1 复位模块向识别层发出复位信号
- 2 所有RN被抑制： $\mathbf{R}=(r_1, r_2, \dots, r_m)=(\mathbf{0}, \mathbf{0}, \dots, \mathbf{0})$ ，上轮被激发的RN被屏蔽
- 3 G1的值恢复为1
- 4 X的值再次被从比较层送到识别层：
 $\mathbf{C}=\mathbf{X}$
- 5 不同的RN被激发，使得不同的 $\mathbf{P}(T_k)$ 被反馈到比较层
- 6 比较层进行相应的比较，并判定本次匹配是否满足要求

查找

7 如果本次匹配不成功，则重复1~6直到如下情况之一发生

7.1 本轮匹配成功。表明已找到一个与X匹配较好的模式，此时，网络进入训练期，对这个匹配的模式进行适当的修改，使它能更好地表示X

7.2 网络中现存的模式均不匹配。因此，网络需要重新构造一个新模式表达此

查找

- 网络用一个还未与任何类关联的RN来对应X所在的类
 - 根据X修改与此RN对应的 T_k 、 B_k
 - 被网络选中的 RN_k 对应的 $T_k = (1, 1, \dots, 1)$
 - $P = (1, 1, \dots, 1)$ 被送入比较层。
 - $C = X \wedge P = X$, 被送入系统复位控制模块, $s=1$ 。而 $\rho \leq 1$, 所以, $s \geq \rho$ 。匹配获得成功

查找

- 首先被选中的RN不一定对应X属于的类
 - 受B取法的影响，有时候，获得最大激励值的RN对应的类不一定是X所属的类
- 例如：设 $n=5$ ，三个输入向量为：
$$X_1 = (1, 0, 0, 0, 0)$$
$$X_2 = (1, 0, 0, 1, 1)$$
$$X_3 = (1, 0, 0, 1, 0)$$

查找

- 假定用初始化B，当 X_1 、 X_2 被输入时， RN_1 、 RN_2 分别被激发
- T_1 、 T_2 、 B_1 、 B_2 分别取如下值
 - $T_1=(1,0,0,0,0)$ ， $B_1=(1,0,0,0,0)$
 - $T_2=(1,0,0,1,1)$ ， $B_2=(0.5,0,0,0.5,0.5)$
- 当 X_3 被输入系统时， RN_1 、 RN_2 获得的激励值都是1
 - RN_2 被选中，则成功

查找

- RN_1 被选中，则出现问题
 - 比较层输出向量 $C = (1, 0, 0, 0, 0)$ ，使得 $s = 0.5$ ，当 $\rho > 0.5$ 时，选择 RN_1 就不能满足精度要求，此时网络就需要进入查找工作阶段

1、 RN_1 获胜

2、 C 取值 $(1, 0, 0, 0, 0)$

3、

$$s = \frac{\sum_{i=1}^5 c_i}{\sum_{i=1}^5 x_i} = 0.5$$

查找

4、 $s < \rho$

5、 RN_1 被屏蔽

6、网络进入第二个查找周期， RN_2 获胜

7、C取值 (1, 0, 0, 1, 0)

8、

$$s = \frac{\sum_{i=1}^5 c_i}{\sum_{i=1}^5 x_i} = 1.0$$

查找

- 9、满足精度要求，停止查找，进入训练期
- 当L取其它的值时，将会有不同的结果
 - 当RN被系统认为是不能满足精度要求后，在继续查找过程中，一直被屏蔽
 - “查找周期”：网络的五个功能模块之间互相影响，加上信号的反馈，使得网络中的信号较为复杂

训练

- T_k 、 B_k 的修改

$$b_{ik} = \frac{Lc_i}{L - 1 + \sum_{j=1}^n c_j}$$

$$t_{ki} = c_i$$

训练

- L是常数
- T的元素只可能从1变成0，不可能从0变成1：用1初始化T的所有元素
- 如果RN_k对应的模式代表类{X₁, X₂, ..., X_d}，则有 $T_k = X_1 \wedge X_2 \wedge \dots \wedge X_d$
- 网络将向量共有的东西作为它的类表示，这也符合一般意义下的“共同特征”的要求

训练

$$b_{ik} = \frac{Lc_i}{L-1 + \sum_{j=1}^n c_j}$$

中含有重要因子

$$\sum_{j=1}^n c_j$$

训练

- 设 X_1 、 X_2 分别使 RN_1 、 RN_2 激发
- 设 $T_1 = X_1$ 、 $T_2 = X_2$
- 如果相应式子中没有该因子，则此时 $B_1 = T_1$ 、 $B_2 = T_2$
- 当 X_1 再一次被输入时， RN_1 、 RN_2 因为获得的网络输入相同而都有被选中的可能
- 如果 RN_2 被选中，则会导致网络运行错误，使得原有的分类被严重破坏

训练

- ΣC_j 可以看成向量 C 的一个度量
 - 越大，产生的权值就越小；
 - 越小，产生的权值就越大。
 - 当一个向量是另一个向量的子集时，能够获得较好的操作

- 例如

$$X_1 = (1, 0, 0, 0, 0)$$

$$X_2 = (1, 0, 0, 1, 1)$$

$$X_3 = (1, 0, 0, 1, 0)$$

训练

- ① X_1 被再次输入，导致 RN_2 被选中；
- ② 识别层将 T_2 送入比较层： $P = T_2$ ；
- ③ 此时， $C = P \wedge X_1 = X_1$ ；
- ④ 复位控制模块根据 C 与 X_1 计算出 $s = 1$ ；
- ⑤ 因为 $s > \rho$ ，所以对网络进行训练：
 $T_2 = C$ 。

显然，其原值被破坏了。而当我们选择一个适当的 L ，同时在调整 B 时保留，这个问题就可以避免了。

训练

- 网络的分类并不是一成不变的
 - 继续使用上面例子中的输入向量，取 $L=6$ ，初始化使 B 的所有元素均取值 0.6
- 1、 X_1 的输入导致 RN_1 被激发； B_1 被训练后取值为 $(1, 0, 0, 0, 0)$
 - 2、输入 X_2 时， RN_1 、 RN_2 所获得的网络输入分别为 1 和 1.8 ，这导致 RN_2 被激发； B_2 被训练后取值为 $(0.6, 0, 0, 0.6, 0.6)$

训练

- 3、如果 X_1 再次被输入， RN_1 、 RN_2 所获得的网络输入分别为1和0.6，从而正确的神经元被激发；如果 X_2 再次被输入， RN_1 、 RN_2 所获得的网络输入分别为1和1.8，从而也仍然有正确的神经元被激发
- 4、当 X_3 被输入时， RN_1 、 RN_2 所获网络输入分别为1和1.2，从而 RN_2 被激发，此时， $T_2 = (1, 0, 0, 1, 1)$ 被送入比较层，使得 $C = T_2 \wedge X_3 = X_3$ 。从而导致 $s = 1 > \rho$

训练

5、网络进入训练： T_2 、 B_2 被修改

$$T_2 = (1, 0, 0, 1, 0)$$

$$B_2 = (6/7, 0, 0, 6/7, 0)$$

6、当再次输入 X_2 时， RN_1 、 RN_2 所获得的网络输入分别为：1和 $12/7$ ，这再次导致 RN_2 被激发。但是，此时识别层送给比较层的 $T_2 = (1, 0, 0, 1, 0)$ 。从而有 $s = 2/3$ ，如果系统的复位控制参数 $\rho > 2/3$ ，此时系统会重新为 X_3 选择一个新的神经元

● 可以让ART在训练完成后，再投入运行

ART Implementation

1. Initialization
2. \underline{T}_j , \underline{B}_j , Vigilance level ρ
3. \underline{B}_j :
4. $0 < b_{ij} < L/(L-1+m)$ for all i, j
5. m : # of components in the input vector
6. L : as constant > 1 ($L=2$, typically)
7. all b_{ij} are the same value
8. \underline{T}_j : $t_{ij}=1$ for all i, j
9. ρ : $0 < \rho < 1$ coarse distinction at first, fine distinction at last.

ART Implementation

- 2. Recognition

- $\underline{NET}_j = (\underline{B}_j \bullet \underline{C})$

- $\underline{OUT}_j = 1 \quad \underline{NET}_j > \underline{T}$

- $0 \quad \text{otherwise}$

- 3. Comparison

$$S = \frac{\# \text{ of } 1 \text{ in } \underline{C}}{\# \text{ of } 1 \text{ in } \underline{X}}$$

- 4. Search

Searching process is going until a pattern is

matched or no pattern is matched.

ART Implementation

- 5. Training
- If the input \underline{X} is matched,

$$b_{ij} \equiv (L \bullet c_i) / (L - 1 + \sum_k c_k)$$

c_i : the i^{th} component of \underline{C}

j : firing neuron

$$L \equiv 2$$

for newly stored \underline{T}_j :

$$t_{ji} \equiv c_i \text{ for all } i$$

ART Algorithm

1. Initialization
2. $L > 1, 0 < \rho < 1$
3. $0 < b_{ij} < L / (L - 1 + m)$ for all i, j
4. $t_{ij} = 1$ for all i, j
- 2. While stopping condition is false, do 3-12:
- 3. For each training pattern(\underline{X}),
 - do 4-11:
 - 4. Set $\underline{R} = 0, \underline{C} = \underline{X}$
 - 5. For each node in recognition layer that is not inhibited ($r \neq -1$):

ART Algorithm

•

10.

$$\text{if } S \equiv \frac{\# \text{ of } 1 \text{ in } \underline{C}}{\# \text{ of } 1 \text{ in } \underline{X}} < \rho, \underline{r}_J \equiv -1$$

$$\text{if } S \equiv \frac{\# \text{ of } 1 \text{ in } \underline{C}}{\# \text{ of } 1 \text{ in } \underline{X}} \geq \rho, \text{ proceed to 11.}$$

11. Update $\underline{B}_J, \underline{T}_J = \underline{C}$

12. Test for stopping conditions

a. no weight change

b. maximum number of epochs reached.

ART Algorithm

- ** Stopping conditions:
 - a. no weight change
 - b. maximum number of epochs reached.

ART2

1. Several sublayers replace F1 layer
2. Update functions:

$$u_i = \frac{v_i}{e + \|\underline{V}\|}, \quad x_i = \frac{w_i}{e + \|\underline{W}\|}, \quad q_i = \frac{p_i}{e + \|\underline{P}\|}$$

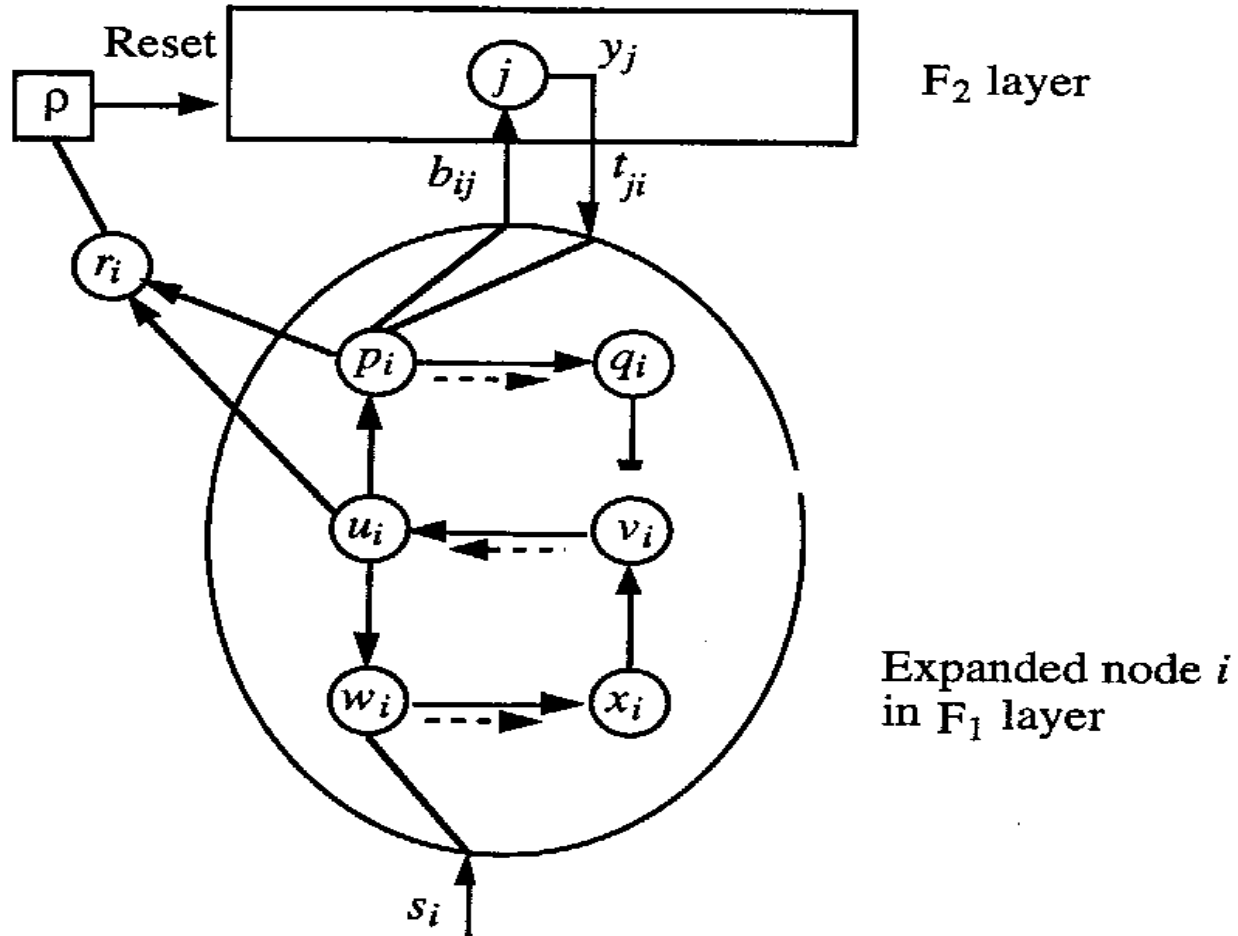
$$w_i = s_i + au_i, \quad p_i = u_i + dt_{ji}, \quad v_i = f(x_i) + bf(q_i)$$

3. Parameters:

m : dimension of input vector

n : number of cluster units

ART2



ART2

3. Parameters(continue):

4. a, b : fixed weight in F1, cannot be zero.

5. c : fixed weight used in testing for Reset, $0 < c < 1$. $\frac{cd}{1-d} \leq 1$

6. d : output activation of F2,

7. e : a small parameter to prevent division by zero when norm of a vector is zero.

8. θ : noise suppression parameter

9. α : learning rate

10. ρ : vigilance level

ART2

3. Algorithm:
4. a. Initialization:
5. a, b, c, d, e, θ , α , ρ
6. b. perform # of epochs of training
7. c. For each input pattern, \underline{S} ,
8. do d. to m.
9. d. update F1

$$u_i \equiv \mathbf{0}, \quad x_i \equiv \frac{s_i}{e + \|\underline{S}\|}, \quad w_i \equiv s_i,$$

$$q_i \equiv \mathbf{0}, \quad p_i \equiv \mathbf{0}, \quad v_i \equiv f(x_i)$$

ART2

d. update F1 again

$$\underline{u}_i \equiv \frac{v_i}{e + \|\underline{V}\|}, \quad \underline{x}_i \equiv \frac{w_i}{e + \|\underline{W}\|},$$

$$\underline{q}_i \equiv \frac{p_i}{e + \|\underline{P}\|}, \quad w_i \equiv s_i + a u_i,$$

$$p_i \equiv u_i, \quad v_i \equiv f(x_i) + b f(q_i)$$

e. compute signal to F2

$$y_j \equiv \sum_i b_{ij} p_i$$

ART2

- f. While Reset is true, do g. to h.
- g. Find winning neuron J in F2
- h. Check for Reset:

$$u_i = \frac{v_i}{e + \|\underline{V}\|}, \quad p_i = u_i + dt_{Ji},$$

$$r_i = \frac{u_i + cp_i}{e + \|\underline{U}\| + c\|\underline{P}\|}$$

ART2

- -
 -
 -
- i) If $\|\underline{R}\| < \rho - e$,
then $y_j = -1$ (inhibit J)
Reset is true; repeat
f.
- ii) If $\|\underline{R}\| \geq \rho - e$,
then

$$\underline{w}_i = \underline{s}_i + a\underline{u}_i, \quad x_i = \frac{w_i}{e + \|\underline{W}\|},$$

$$\underline{q}_i = \frac{\underline{p}_i}{e + \|\underline{P}\|}, \quad v_i = f(x_i) + bf(q_i).$$

ART2

- i. Do j. to 1. for # of learning iterations.
- j. update weight for neuron J

$$t_{Ji} \equiv \alpha d u_i + \{1 + \alpha d(d-1)\} t_{Ji}$$

$$b_{iJ} \equiv \alpha d u_i + \{1 + \alpha d(d-1)\} b_{iJ}$$

k. Update F1

$$u_i \equiv \frac{v_i}{e + \|\underline{V}\|}, \quad x_i \equiv \frac{w_i}{e + \|\underline{W}\|},$$

$$q_i \equiv \frac{p_i}{e + \|\underline{P}\|}, \quad w_i \equiv s_i + \alpha u_i,$$

$$p_i \equiv u_i + dt_{ji}, \quad v_i \equiv f(x_i) + bf(q_i)$$

ART2

- - l. Test stopping condition for weight updates.
- - m. Test stopping condition for # of epochs.

ART2

- - l. Test stopping condition for weight updates.
- - m. Test stopping condition for # of epochs.

Thank You

Question!

Intelligence Science

<http://www.intsci.ac.cn/>

