SOFT COMPUTING
Subject Code : 18MIT24C
Prepared by Dr. N.Thenmozhi

UNIT-III: Adaptive Resonance Theory: Introduction: Cluster Structure, Vector Quantization, Classical ART Networks, Simplified ART Architecture. ART1: Architecture of ART1–Special features of ART1 Models-ART1 Algorithms. ART2: Architecture of ART2–ART2 Algorithms.

TEXT BOOK

1. S.Rajasekaran & G.A.Vijayalakshmi Pai, "Neural Networks, Fuzzy logic, and Genetic Algorithms Synthesis and Applications, PHI, 2005.

REFERENCE BOOKS

1. James A. Freeman, David M.Skapura, "Neural Networks-Algorithms, Applications, and Programming Techniques", Pearson Education.
2. Fredric M. Ham, Ivica Kostanic, "Principles of Neuro computing for science of Engineering", TMCH.

3. Adaptive Resonance Theory

Introduction:

Adaptive resonance theory (ART) is a **theory** developed by Stephen Grossberg and Gail Carpenter on aspects of how the brain processes information. It describes a number of neural **network** models which use supervised and unsupervised learning methods, and address problems such as pattern recognition and prediction.

3.1 Cluster Structure

Hierarchical **clustering** results in a **clustering structure** consisting of nested partitions. If the nesting occurs in the other direction, that is, the **clustering** begins with one large **cluster** and breaks down into smaller **clusters**, it is known as a divisive **clustering** algorithm.

Clustering is a fundamental data analysis method. It is widely used for pattern recognition, feature extraction, vector quantization (VQ), image segmentation, function approximation, and data mining. **Clustering** methods can be based on statistical model identification (McLachlan & Basford, 1988) or competitive learning.

Clustering is a fundamental data analysis method. It is widely used for pattern recognition, feature extraction, vector quantization (VQ), image segmentation, function approximation, and data mining. As an unsupervised classification technique, clustering identifies some inherent structures present in a set of objects based on a similarity measure. Clustering methods can be based on statistical model identification (McLachlan & Basford, 1988) or competitive learning. In this paper, we give a comprehensive overview of competitive learning based clustering methods. Importance is attached to a number of competitive learning based clustering neural networks such as the self-organizing map (SOM), the learning vector quantization (LVQ), the neural gas, and the ART model, and clustering algorithms such as the C-means, mountain/subtractive clustering, and fuzzy C-means (FCM) algorithms. Associated topics such as the under-utilization problem, fuzzy clustering, robust clustering, clustering based on non-Euclidean distance measures, supervised clustering, hierarchical clustering as well as cluster validity are various clustering methods.

3.2 Vector Quantization

- Vector quantization (VQ) is a classical quantization technique from signal processing that allows the modeling of probability density functions by the distribution of prototype vectors.
- It was originally used for data compression.
- It works by dividing a large set of points (vectors) into groups having approximately the same number of points closest to them.
- Each group is represented by its centroid point, as in k-means and some other clustering algorithms

Vector quantization (VQ) is a classical method for approximating a continuous probability density function (PDF) p(x) of the vector variable $x \in R$ n by using a finite number of prototypes. A set of feature vectors x is represented by a finite set of prototypes $\{c1, \ldots, cK\} \subset R$ n , referred to as the codebook. Codebook design can be performed by using clustering. Once the codebook is specified, approximation of x is to find the reference vector c from the codebook that is closest to x (Kohonen, 1989, 1997). This is the nearest-neighbor paradigm, and the procedure is actually simple competitive learning (SCL). The codebook can be designed by minimizing the expected squared quantization error

$$E = Z\ kx - ck\ 2\ p(x)dx \quad \text{--------}(1)$$

where c is a function of x and ci . Given the sample xt , an iterative approximation scheme for finding the codebook is derived by Kohonen (1997)

$$ci(t + 1) = ci(t) + \eta(t)\delta wi\ [xt - ci(t)] \quad \text{-------}\ (2)$$

where the subscript w corresponds to the winning prototype, which is the prototype closest to xt , δwi is the Kronecker delta.

3.3 Classical ART Networks

ART system has been utilized to clarify different types of cognitive and brain data.The Adaptive Resonance Theory addresses the **stability-plasticity**(stability can be defined as the nature of memorizing the learning and plasticity refers to the fact that they are flexible to gain new information) dilemma of a system that asks how learning can proceed in response to huge input patterns and simultaneously not to lose the stability for irrelevant patterns. Other than that, the stability-elasticity dilemma is concerned about how a system can adapt new data while keeping what was learned before. For such a task, a feedback mechanism is included among the ART neural network layers.

In this neural network, the data in the form of processing elements output reflects back and ahead among layers. If an appropriate pattern is build-up, the resonance is reached, and then adaption can occur during this period. It can be defined as the formal analysis of how to overcome the learning instability accomplished by a competitive learning model, let to the presentation of an expended hypothesis, called **adaptive resonance theory** (ART). This formal investigation indicated that a specific type of top-down learned feedback and matching mechanism could significantly overcome the instability issue. It was understood that top-down attention mechanisms, which had prior been found through an investigation of connections among cognitive and reinforcement mechanisms, had similar characteristics as these code-stabilizing mechanisms.

In other words, once it was perceived how to solve the instability issue formally, it also turned out to be certain that one did not need to develop any quantitatively new mechanism to do so. One only needed to make sure to incorporate previously discovered attentional mechanisms. These additional mechanisms empower code learning to self-stabilize in response to an essentially arbitrary input system.

- Stability: system behaviour doesn't change after irrelevant events
- Plasticity: System adapts its behaviour according to significant events
- Dilemma: how to achieve stability without rigidity and plasticity without chaos?
    - Ongoing learning capability
    - Preservation of learned knowledge

ART Types

- ART1: Unsupervised Clustering of binary input vectors.
- ART2: Unsupervised Clustering of real-valued input vectors.
- ART3: Incorporates "chemical transmitters" to control the search process in a hierarchical ART structure.
- ARTMAP: Supervised version of ART that can learn arbitrary mappings of binary patterns.
- Fuzzy ART: Synthesis of ART and fuzzy logic.
- Fuzzy ARTMAP: Supervised fuzzy ART
- dART and dARTMAP: Distributed code representations in the F2 layer (extension of winner take all approach).
- Gaussian ARTMAP

3.4. Simplified ART Architecture

- Adaptive Resonance Theory(ART) is a type of neural network technique developed by stephen Grossberg and Gail Carpenter in 1987.
- The basic ART uses unsupervised learning technique.
- The term "adaptive" and "resonance" used in this suggests that they are open to new learning(i.e.,adaptive)without discarding the previous or the old information(i.e., resonance).

ART Architecture

- Bottom-up weights b_{ij}
- Top-down weights t_{ij}
    - Store class template
- Input nodes
    - Vigilance test
    - Input normalisation
- Output nodes
    - Forward matching
- Long-term memory
    - ANN weights
- Short-term memory
    - ANN activation pattern

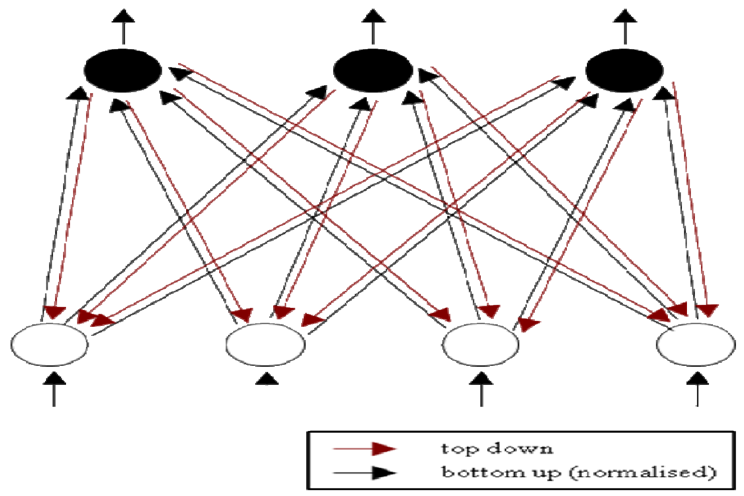


Figure 1 : Simplified ART architecture

3.5 ART1

It is the simplest and the basic ART architecture. It is capable of clustering binary input values.

ART1 is an unsupervised learning model primarily designed for recognizing binary patterns. It comprises an attentional subsystem, an orienting subsystem, a vigilance parameter, and a reset module, as given in the figure given below. The vigilance parameter has a huge effect on the system. High vigilance produces higher detailed memories. The ART1 attentional comprises of two competitive networks, comparison field layer L1 and the recognition field layer L2, two control gains, Gain1 and Gain2, and two short-term memory (STM) stages S1 and S2. Long term memory (LTM) follows somewhere in the range of S1 and S2 multiply the signal in these pathways.

It is a type of ART, which is designed to cluster binary vectors. We can understand about this with the architecture of it.

3.5.1.Architecture of ART1

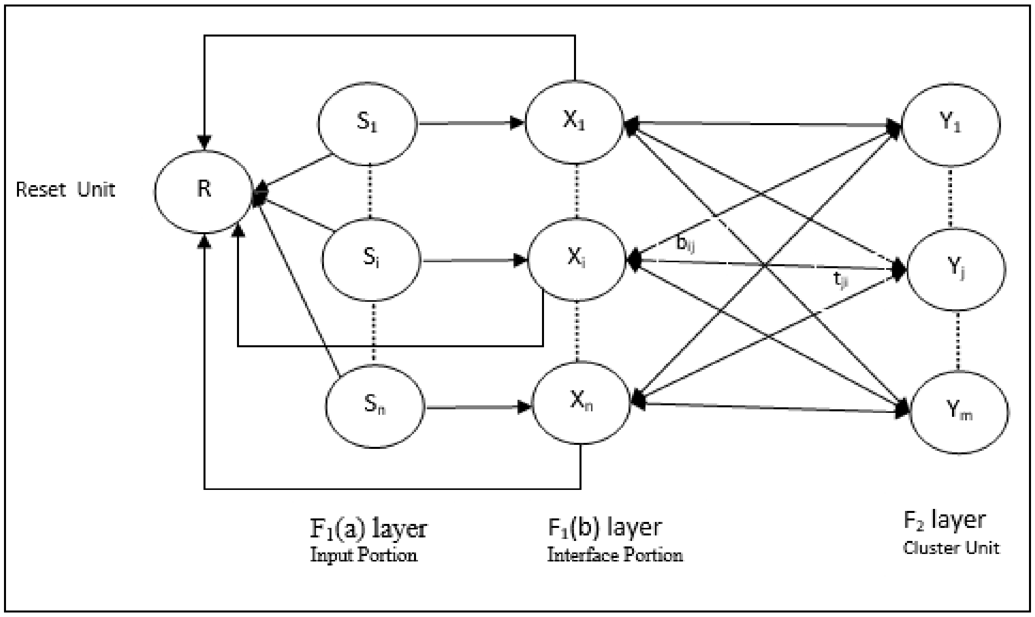
It consists of the following two units –

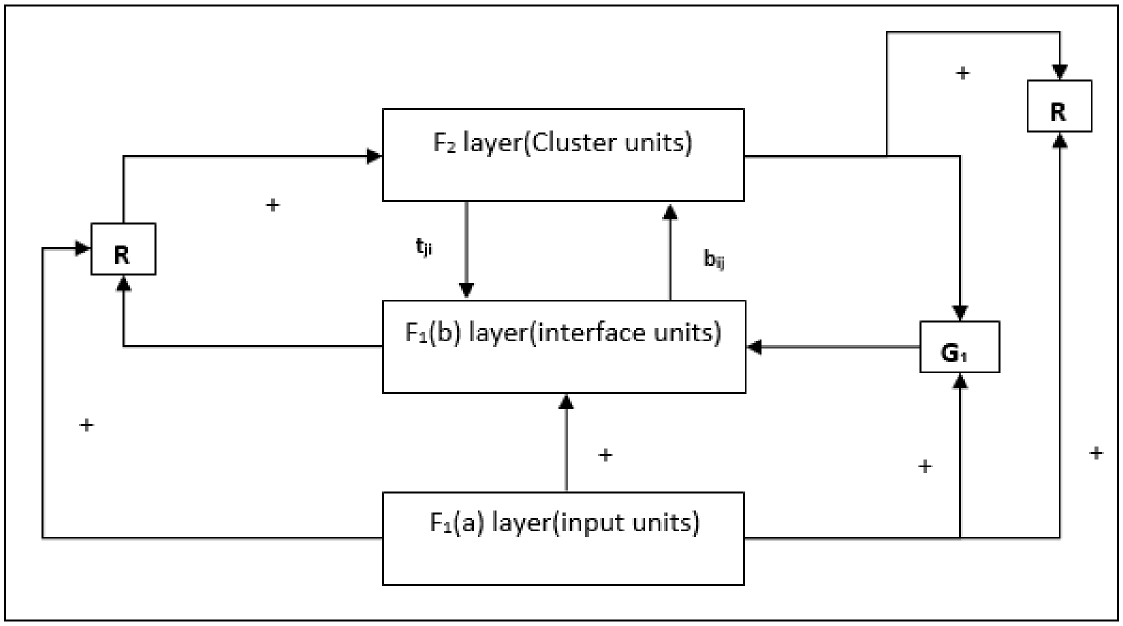
Computational Unit – It is made up of the following –

- **Input unit (F_1 layer)** – It further has the following two portions –
    - **F_1**aa **layer** InputportionInputportion – In ART1, there would be no processing in this portion rather than having the input vectors only. It is connected to F_1bb layer interfaceportioninterfaceportion.
    - **F_1**bb **layer** InterfaceportionInterfaceportion – This portion combines the signal from the input portion with that of F_2 layer. F_1bb layer is connected to F_2 layer through bottom up weights $\mathbf{b_{ij}}$ and F_2 layer is connected to F_1bb layer through top down weights $\mathbf{t_{ji}}$.
- **Cluster Unit (F_2 layer)** – This is a competitive layer. The unit having the largest net input is selected to learn the input pattern. The activation of all other cluster unit are set to 0.

- **Reset Mechanism –** The work of this mechanism is based upon the similarity between the top-down weight and the input vector. Now, if the degree of this similarity is less than the vigilance parameter, then the cluster is not allowed to learn the pattern and a rest would happen.

Supplement Unit – Actually the issue with Reset mechanism is that the layer **F_2** must have to be inhibited under certain conditions and must also be available when some learning happens. That is why two supplemental units namely, **G_1** and **G_2** is added along with reset unit, **R**. They are called **gain control units**. These units receive and send signals to the other units present in the network. **'+'** indicates an excitatory signal, while **'–'** indicates an inhibitory signal.





Parameters Used

Following parameters are used –

- **n** – Number of components in the input vector
- **m** – Maximum number of clusters that can be formed
- **b_{ij}** – Weight from F_1bb to F_2 layer, i.e. bottom-up weights
- **t_{ji}** – Weight from F_2 to F_1bb layer, i.e. top-down weights
- **ρ** – Vigilance parameter
- **||x||** – Norm of vector x

3.5.2. Special features of ART1 Models

Recognition Phase

- Forward transmission via bottom-up weights
- Input pattern matched with bottom-up weights (normalised template) of output nodes
- Inner product $x \bullet b_i$
- Best matching node fires (winner-take-all layer)
- Similar to Kohonen's SOM algorithm, pattern associated to closest matching template
- ART1: fraction of bits of template also in input pattern

Comparison Phase

- Backward transmission via top-down weights
- Vigilance test: class template matched with input pattern
- If pattern close enough to template, categorisation was successful and "resonance" achieved
- If not close enough reset winner neuron and try next best matching
- Repeat until
  - vigilance test passed
  - Or (all committed neurons) exhausted

Vigilance Threshold

- Vigilance threshold sets granularity of clustering
- It defines amount of attraction of each prototype
- Low threshold
  - Large mismatch accepted
  - Few large clusters
  - Misclassifications more likely
- High threshold
  - Small mismatch accepted
  - Many small clusters
  - Higher precision

Adaptation

- Only weights of winner node are updated
- Only features common to all members of cluster are kept
- Prototype is intersection set of members

3.5.3 ART1 Algorithms

Step 1 – Initialize the learning rate, the vigilance parameter, and the weights as follows –

α>1and0<ρ≤1α>1and0<ρ≤1

0<bij(0)<αα−1+nandtij(0)=10<bij(0)<αα−1+nandtij(0)=1

Step 2 – Continue step 3-9, when the stopping condition is not true.

Step 3 – Continue step 4-6 for every training input.

Step 4 – Set activations of all F_1aa and F_1 units as follows
F_2 = 0 and F_1aa = input vectors
Step 5 – Input signal from F_1aa to F_1bb layer must be sent like
si=xisi=xi

Step 6 – For every inhibited F_2 node

yj=∑ibijxiyj=∑ibijxi the condition is $\mathbf{y_j \neq -1}$

Step 7 – Perform step 8-10, when the reset is true.

Step 8 – Find **J** for $\mathbf{y_J \geq y_j}$ for all nodes **j**

Step 9 – Again calculate the activation on F_1bb as follows
xi=sitJixi=sitJi

Step 10 – Now, after calculating the norm of vector **x** and vector **s**, we need to check the reset condition as follows –

If **||x||/ ||s||** < vigilance parameter **ρ**, then inhibit node **J** and go to step 7

Else If **||x||/ ||s||** ≥ vigilance parameter **ρ**, then proceed further.

Step 11 – Weight updating for node **J** can be done as follows –

bij(new)=αxiα−1+||x||bij(new)=αxiα−1+||x||

tij(new)=xitij(new)=xi

Step 12 – The stopping condition for algorithm must be checked and it may be as follows –

- Do not have any change in weight.
- Reset is not performed for units.
- Maximum number of epochs reached.

3.6 ART2

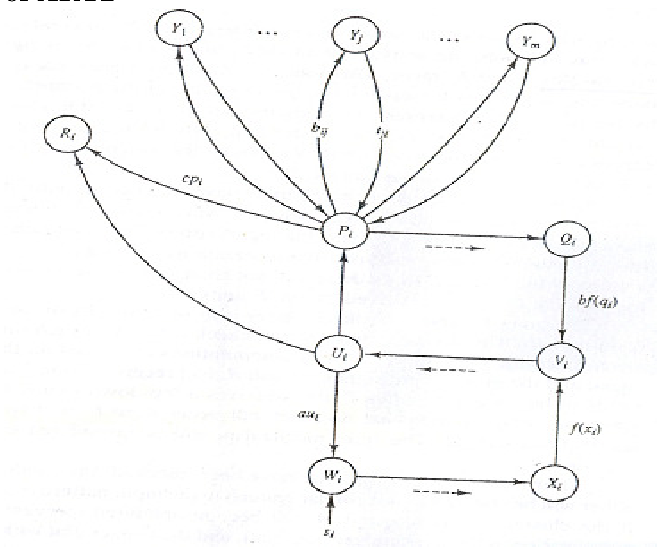
- There are no fundamental differences in learning and recognition processes with such structure of a network. The only difference is that there is a change in the training weights of connections *zij*, made by recognizing neurons. And in recognizing process the change in weights does not happen.
- The algorithm of the network working consists of the following stages.

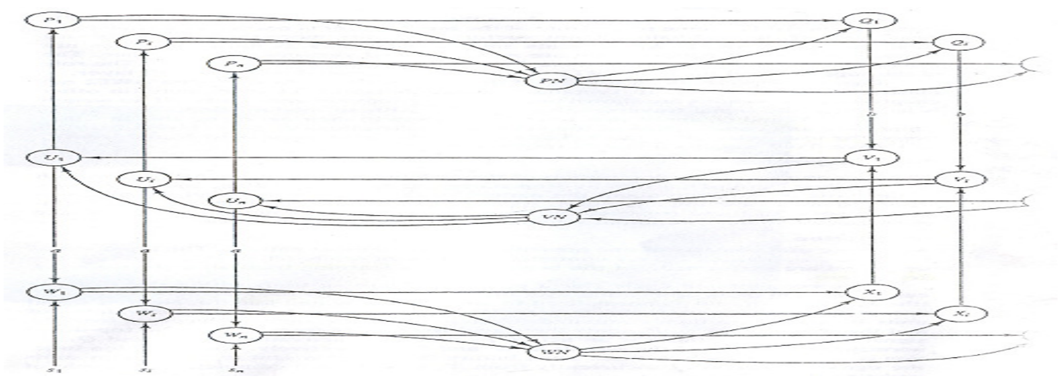
  Unsupervised Clustering for :
  - Real-valued input vectors
  - Binary input vectors that are noisy
  - Includes a combination of normalization and noise suppression

3.6.1 ART2 Learning Mode

- Fast Learning
  - Weights reach equilibrium in each learning trial
  - Have some of the same characteristics as the weight found by ART1
  - More appropriate for data in which the primary information is contained in the pattern of components that are 'small' or 'large'
- Slow Learning
  - Only one weight update iteration performed on each learning trial
  - Needs more epochs than fast learning
  - More appropriate for data in which the relative size of the nonzero components is important

3.6.2 Architecture of ART2





3.6.3 ART2 Algorithms

Step 0: Initialize parameters:

a, b, ɵ, c, d, e, α, ρ.

Step 1: Do Steps 2-12 N-EP times.

(Perform the specified number of epochs of training.)

Step 2: For each input vector s, do steps 3-11.

Step 3: Update F1 unit activations:

Update F1 unit activations again:

$$u_i = 0$$
$$w_i = s_i$$
$$p_i = 0$$
$$x_i = \frac{s_i}{e + \|s\|}$$
$$q_i = 0$$
$$v_i = f(x_i)$$

$$u_i = \frac{v_i}{e + \|v\|}$$
$$w_i = s_i + a u_i$$
$$p_i = u_i$$
$$x_i = \frac{w_i}{e + \|w\|}$$
$$q_i = \frac{p_i}{e + \|p\|}$$
$$v_i = f(x_i) + b f(q_i)$$

Step 4: Compute signals to F2 units:

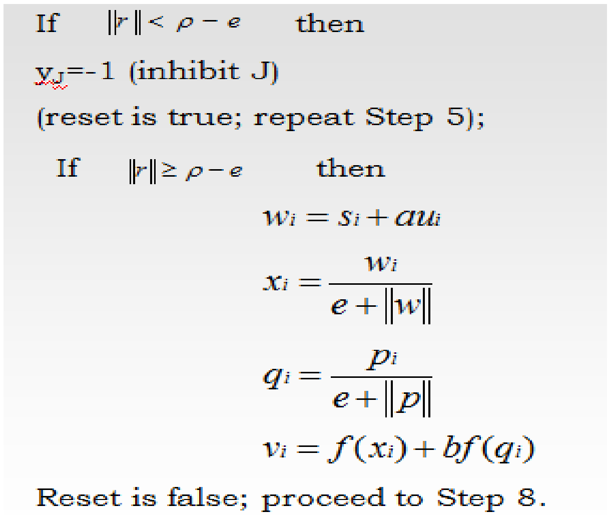
$$y_j = \sum_i b_{ij} p_i$$

Step 5: While reset is true, do Steps 6-7.

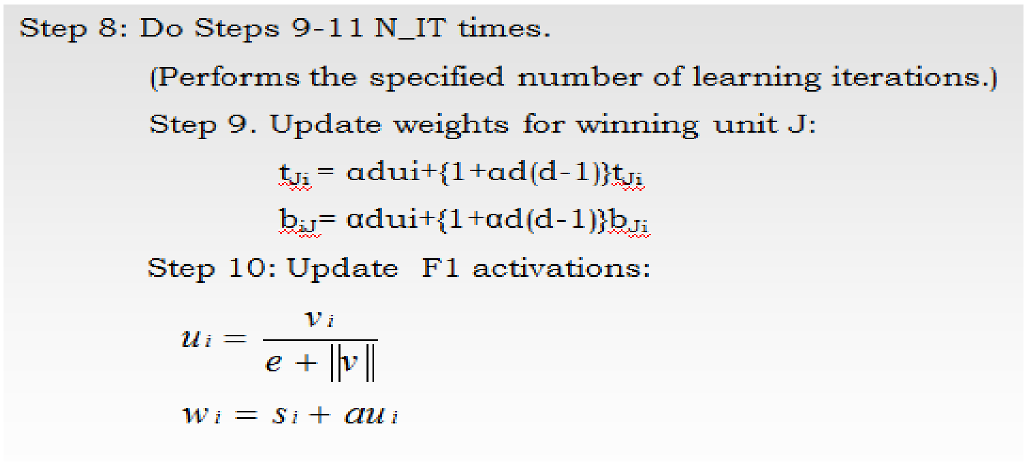
Step 6: Find F2 unit Y_J with largest signal .(Define J such that $y_J \geq y_j$ for j=1…m.)

Step 7: Check for reset:

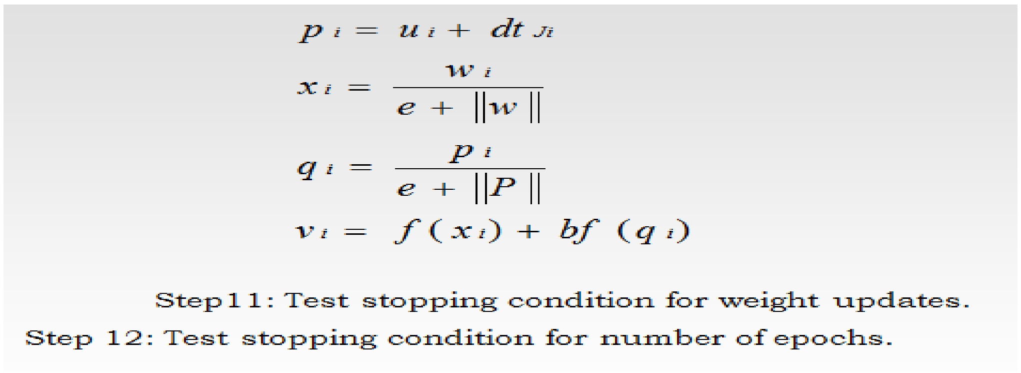
$$u_i = \frac{v_i}{e + \|v\|}$$
$$p_i = u_i + d t_{Ji}$$
$$r_i = \frac{u_i + c p_i}{e + \|u\| + c\|p\|}$$



```
If   $\|r\| < \rho - e$   then
$y_J$=-1 (inhibit J)
(reset is true; repeat Step 5);
  If   $\|r\| \geq \rho - e$   then
        $w_i = s_i + au_i$
        $x_i = \frac{w_i}{e + \|w\|}$
        $q_i = \frac{p_i}{e + \|p\|}$
        $v_i = f(x_i) + bf(q_i)$
Reset is false; proceed to Step 8.
```



```
Step 8: Do Steps 9-11 N_IT times.
        (Performs the specified number of learning iterations.)
        Step 9. Update weights for winning unit J:
                $t_{Ji}$ = αdui+{1+αd(d-1)}$t_{Ji}$
                $b_{iJ}$= αdui+{1+αd(d-1)}$b_{Ji}$
        Step 10: Update  F1 activations:
          $u_i = \frac{v_i}{e + \|v\|}$
          $w_i = s_i + au_i$
```



```
                $p_i = u_i + dt_{Ji}$
                $x_i = \frac{w_i}{e + \|w\|}$
                $q_i = \frac{p_i}{e + \|P\|}$
                $v_i = f(x_i) + bf(q_i)$

        Step11: Test stopping condition for weight updates.
Step 12: Test stopping condition for number of epochs.
```

Final Cluster Weight Vector

- In fast learning : $t_{Ji} = \frac{1}{1-d} u_i$
- In slow learning :
  - After adequate epochs , top-down weights converge to average of learned patterns by that cluster