The 7th Iranian Conference on Electrical Engineering
Iran Telecommunication Research Center
May 17-19 1999
Computer

A Method For Adaptation Of Vigilance Factor And Choice Parameter In Fuzzy ART System

P. Bahri M. R. Meybodi

Department of Computer Engineering
Amirkabir University
Tehran, Iran

Abstract. This paper presents a method to determine optimal values of vigilance factor and choice parameter in ART networks. As we have noted little effort has done in the literature in this respect. Usually vigilance factor and choice parameter is chosen by trial and error or by guess. Naturally there may be better values of the parameters than the chosen one. With this in mind we examined the abilities of the learning automata to adapt the vigilance factor and choice parameter for best performance. The results we obtained were in fact surprising. The augmented network trained with learning automaton survived advantages of both high and low values of the parameters. The network was as fast as the fastest networks and exact as the most exact one. The results of simulations on the several benchmarks prove this assertion.

Keywords. Neural Networks, Fuzzy ART, Vigilance Factor, Choice parameter, Learning Automata.

1-Introduction.

One of the most popular networks used for recognition, classification, and association tasks commonly encountered in practice is fuzzy ART. There are many versions of the ART system developed in the literature such as ART_MAP[3], ART_EMAP[2], LAPART[5]. Most of the time these networks associate input patterns with output ones. In classification tasks however, the task of the networks is associating the input to one of the specific classes(outputs). In this report, what we have done, was of classification type. In the circle in the square experiment, the network differentiates between the points inside the circle and the rest in the square. Also in spirals experiment the network differentiates between the points of two nested spirals. In noisy digit benchmark the network recognizes persian digits with different signal to noise ratios.

ART system consists of two networks, the input and the output. Both of these accept separate vectors to their input. When input is presented to ART, a comparison is done in the network to test the similarity of the input to stored prototypes. The critical entities that determine the dynamics of the network are vigilance factor and choice parameter. Most of the time these parameters are constant. Li and Zhan [6] have analysed adaptation of vigilance factor in art II by fuzzy methods, and have shown the robustness of the network in noisy environments, but they have done nothing with the choice parameter. Several learning automata based algorithms for adaptation of backpropagation parameters have been reported in the literature [7], [8], [9]. Our aim was to adapt the parameters of the fuzzy ART to the conditions of the input and use the advantages of both high and low values of those.

With this in mind, we used the learning automaton to adapt the parameters. During learning phase the reaction of the environment with respect to vigilance factor and choice parameter was evaluated. Doing this, nearly minimum number of nodes generated, as a result th network was fast, also the most exact classification resulted. The rest of the paper is organized as follows. Section two explains the learning automaton. Section three discuses the augmented fuzzy ARTMAP in some detail. Section four shows in detail our implementation method, and section five discuses simulation results.

II -The Learning Automaton

The learning automaton (LA) can be viewed as an agent that interacts with an environment. They are related by a feedback mechanism. The automaton applies a set of actions to the environment, and the environment responds probablisticly to these actions. There are three models of LA. The first one which is called the P_model is the model which is described and used in this paper. In this model, the environment's response are two, which are either zero or one. Zero is the rewarding (considered desired) and one is the penalizing response (considered undesired).

Suppose $a = (a_1, a_2, \ldots, a_r)$ is the set of actions that the automaton chooses. There are corresponding numbers $c = (c_1, c_2, \ldots, c_r)$ which are the probabilities of penalties from the environment (response equal to one). That is, with probability c_i environment will respond in the negative. So the goal of the automaton is to choose the action that will result in minimum probability of penalty, that is, it chooses a_i with minimum corresponding c_i.

The automaton has some internal states that determine its behavior. It means, it moves among internal states and takes actions according to its states. The learning automaton is classified into deterministic or stochastic. If the process of changing states and producing outputs is deterministic, that is an action is taken with probability one or zero, the automaton is called deterministic and if the probabilities are not merely zero and one, the automaton is called stochastic.

If the probabilities are constant the automaton is called fixed structure and if they are varying with respect to time it is called variable structure automaton. The automaton can be optimal if it results in minimum probability of unfavorable response. It can be ε-optimal, that has the property that there exists an N1 such that for N>N1 the probability of unfavorable response is less than *min* $\{c_i, i=1,2,..r\}$ plus an arbitrarily small number ε.

In this paper we used both stochastic variable structure and fixed structure automata. The variable structure automaton keeps certain values p_i, i=1,2,..r corresponding to a_i,i=1,2,...r with the condition that the sum of p_i's are equal to one. These numbers are the probability of choosing the corresponding action as the next action. When the automaton receives penalty from the environment the probability of the action taken will be decreased and the rest of the probabilities will be increased according to some learning algorithm. Conversely, if the environment responds favorably the taken action's probability will be increased and the rest will be decreased.

Of the fixed structure learning automata, we used the Krinskey and Krylov automata in this paper.The diagrams in figure 1 shows the state transitions of the two action N state Krinsky automaton. When a favorable response(0) is recieved from the environment, the next state of the automaton will be the deepest state corresponding to the action taken, and the same action will be performed again. But if the response is unfavorable(1), the state will move towards the surface state which is N. Or if the state is N, the automaton will change output, and go to state N of the other action.

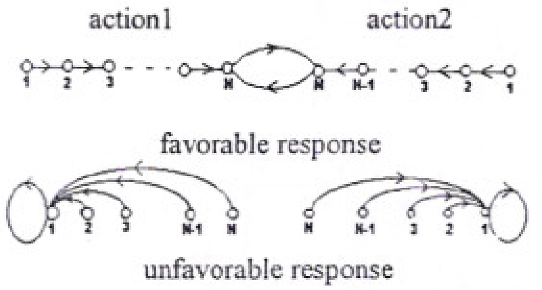

figure 1

Figure 2 shows the state diagram of the Krylov automaton. When the response of the environment

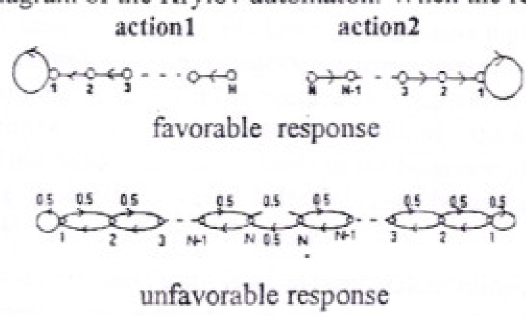

figure-2

is favorable(0), the automaton will perform the same action and move towards state one, or if it is already in state one it will remain in the same state. But when the response is unfavorable(1), for state i different than one and N, it will perform the same action, and with probability 0.5 it will move to state i +1 or with the same probability to state i -1. If it is at state one, it will remain in the same state with probability 0.5 or will move to state 2 with probability 0.5. Finally if it is in state N, it will change output and will move to state N of other action with probability 0.5, or will move to state N-1 with probability 0.5, producing the same action. For a comprehensive discussion of the learning automaton refer to [1].

III -The ART System Configuration.

Mostly, ART system consists of two subnetworks that are related to each other by anothe network as it is seen from figure 3.
The two subnetworks are called ART_a and ART_b. ART_a accepts and processes input pattern vector and ART_b recieves and processes output pattern vector. The third network shown in the figure, that is, the map field sets up relation between the two patterns. During learning phase ART_a accepts the inputs to be classified and ART_b accepts the correct prediction corresponding to the input. The two networks learn to classify their inputs through minimax rule that will be explained later on.

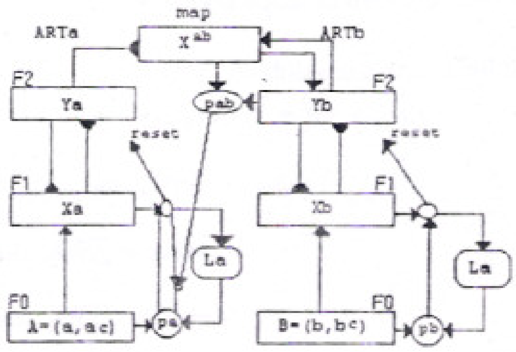

figure 3

Each ART module has an F0 layer that accepts input and the complement of the input. Each node in the complemented category is complimented with respect to one of the original nodes. This way automatically the input becomes normalized that is, it acquires norm n, where n is the number of input nodes. After complementation, the normal input is submitted to another layer called F1. This layer has central role in the behavior of the overall network. First it accepts the input coming from the layer F0. It then sends bottom up excitation signals through the weights denoted by W_{ij} to another layer called F2.

The dynamics of our fuzzy ART system is determined by three parameters that are as follows. First, the small positive number α which is also called the choice parameter. In our algorithm it changes during time to determin optimum value. Second, there is a β that is the learning rate and belongs to the interval zero and one and typically is one at the first time (corresponding to fast learning) and then gradually decreases towards zero. Third and most important of all is vigilance factor that may change and indeed in our method it changes dynamically during learning in order to better train and use advantages of both high and low vigilance values.

We said that F1 sends bottom up signals to F2. At this layer the following values are computed for each node

$$T_j(I) = \frac{|I \wedge W_j|}{a + W_j}$$

In the above formula T_j is the choice function computed at each node. W_j is the weight from layer F1 to this node. α is the choice parameter mentioned above and |.| is the norm of the vector or the sum of the absolute values of its components. The operator $\wedge$ is the fuzzy min operator. At this stage all T_j's and the maximum of them are computed. If more than one node satisfies the maximum condition the one with the lowest index is chosen. Afterwards, the top down expectation is sent to the F1 layer through top down weights W_{td}. At F1 the following inequality is examined

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

If this inequality is satisfied we are done, that is, the Jth node at F2 is the correct representation of the input. But otherwise, mismatch reset occurs and the search for the next maximum node at the

F2 layer begins with the value of the Jth node equal to zero. During training, if for all nodes mismatch occur one new node is generated with the weight equal to the input(fast learning), in the other case(when working) the input is rejected.

The map field is activated when one of the ART_a or ART_b modules is active. Call the output of ART_a, W_{Jab} and the output of ART_b, y_b. The map field output vector equals

$$
\begin{array}{ll}
W_{Jab} \wedge y_b & \text{If } ART_a \text{ and } ART_b \text{ are active} \\
W_{Jab} & \text{If only } ART_a \text{ is active} \\
y_b & \text{If only } ART_b \text{ is active} \\
0 & \text{If none is active}
\end{array}
$$

Denote the vigilance factor of the map field as ρ_{ab}. If $|X_{ab}| < \rho_{ab}|y_b|$ then ρ_a (the vigilance factor of ART_a) is increased to be slightly larger than $|I \wedge W_{Ja}|/|I|$ and search begins again in F2a (the F2 layer of ART_a) for another J with

$$|X_a| = |I \wedge W_{Ja}| > \rho_a |I|$$

and

$$|X_{ab}| = |y_b \wedge W_{Jab}| \geq \rho_{ab}|y_b|$$

The learning of the map field is such that initially $W_{jk}(0) = 1$, afterwards, when the ART_a group J learn to predict ART_b group K, that association remains one for all time and the rest become zero.

The learning automaton is shown in the figure 3 in the box labled LA. As will be explained in the sections following, it watches the network for mismatch or match tracking to occur, it then acts on the vigilance factor, and choice parameter.

IV-Implementation

We applied our automaton to determine the optimum vigilance criteria and choice parameter, and also optimize learning. That is, to use the advantage of both high and low values of the parameters. Also we had in mind to minimize training and network action durations. With this in mind we put our automaton where decision was made at the comparison point. We took the desired response of the environment as the occurrence of resonance and the undesired response as the occurrence of the generation of a new node when no match was found. We predicted that taking this strategy would result in minimum number of nodes.

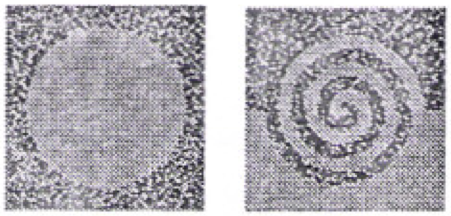

figure 4

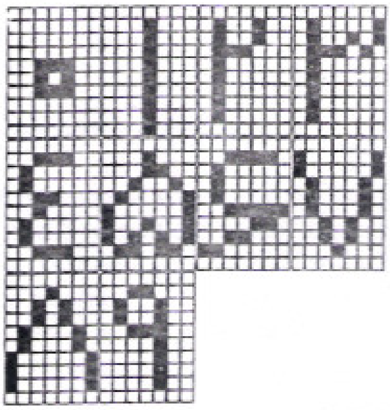

figure 5

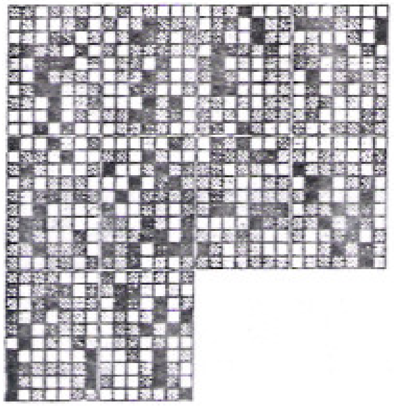

figure 6

We used three benchmarks to test the method, first the circle in the square benchmark with 400 training and 400 test points. And the other was spirals with 600 points for training and 600 other for testing. These benchmarks with different number of samples is described in [3], and shown in the figure 4. The task of the network is to separate the points belonging to the two groups. The third benchmark was noisy persian digits which are shown in figure 5, and a noisy version of which with signal to noise ratio of 2.0 is shown in figure 6..

As the actions of the learning automaton we took assignment of 11 different values to vigilance factor, and 11 different values for choice parameter. The vigilance factor values were in the increments of 0.03 between 0.66 and 0.96, and the choice parameter values were between 0.06 to 0.36 in the increments of 0.03. Let us denote $p_i(n)$ the probability of choosing action i in step n. We can summarize the proposed algorithm as follows.

```
Augmented_Art
 begin
  Apply the input.
  While(any nonzero F2 node remains)
    begin
      y=max(F2 node)
      if(vigilance criteria met for y)
        begin
          update weights
          call Automaton(vigilance factor,favorable)
          call Automaton(choice parameter,favorable)
          break out of while
        end
      else
        begin
          reset (y)
        end if
    end while
 if(vigilance criteria doe not met for any node)
  begin
   call Automaton(vigilance factor,unfavorable)
   call Automaton(choice parameter,unfavorable)
 end if
 end algorithm.
```

```
Automaton(parameter,response)
begin
    If(response=favorable)
     begin
       if(I'th action taken)
          apply learning rule to
          probabilities of actions
     end
    else
      if(I'th action taken)
        apply other learning rule to
         probabilities of actions
     end
    determine the next action
 end.
```

It should be mentioned that, in the first set of experiments we adapted only the vigilance factor thus, only the first call to the procedure **Automaton** was made.

V -Simulation Results.

1.Adapting vigilance factor only

In the first set of simulations we trained the fuzzy ART network with respect to vigilance factor only, with and without variable structure automaton. The map field was very simple as is in the classification tasks. This field learned only at the beginning and when a node was generated. The corresponding field weight was made equal to one and all irrelevant ones were made zero. At our network, ART_b had only one layer consisting of two nodes corresponding to each class.

We gathered statistics regarding the number of nodes generated, the time it took for the network to train and function, the number of inputs rejected during functioning, and the vigilance values chosen. Table 1 shows the number of nodes generated. The first column shows the vigilance values we used in the training and recognition phase, and the last entry in this column shows the network with automaton. The number of nodes generated in the network with automaton nearly equals the least of all, that is, 0.66. This shows that our method causes nearly minimum number of nodes to be generated. Although the number of nodes is minimum, the vigilance value probabilities are much more at high values that is, around 0.96, but the number of nodes generated is much less than the one for 0.96 which is the vigilance value mostly favoured by LA.

Table two shows the run times for both experiments. The run times are near the minimum for the augmented network. This is because the network has nearly the fewest nodes. Table three shows the number of times that inputs were rejected during recognition phase. Note that, at vigilance value of 0.96 the highest rate of rejection is obtained. This is natural because the network has the strictest comparison criteria that is, the narrowest selection property. Although

table 1

vigilances	nodes generated	
	circle in the square	nested spirals
0.66	27	64
0.69	27	62
0.72	29	63
0.75	32	65
0.78	37	68
0.81	42	73
0.84	49	81
0.87	61	96
0.90	80	116
0.93	111	158
0.96	184	270
LA	38	73

table 2

vigilances	run times	
	circle in the square	nested spirals
0.66	5	12
0.69	5	12
0.72	5	13
0.75	6	15
0.78	7	18
0.81	6	22
0.84	7	21
0.87	7	25
0.90	7	22
0.93	9	20
0.96	15	32
LA	5	13

table 3

vigilances	times rejected	
	circle in the square	nested spirals
0.66	0	0
0.69	1	0
0.72	0	0
0.75	1	1
0.78	1	1
0.81	3	3
0.84	6	9
0.87	16	18
0.90	32	32
0.93	70	66
0.96	173	171
LA	3	4

table 4

vigilances	nodes generated	nodes rejected
0.66	33	57
0.69	38	72
0.72	43	141
0.75	50	213
0.78	61	241
0.81	66	314
0.84	78	322
0.87	88	395
0.90	88	494
0.93	98	500
0.96	98	500
LA	78	323

the network with automaton has favoured the highest value of vigilance, it has near the minimum rejection rate which shows that the network is very accurate and elastic.

Table 4 shows the number of nodes generated and rejected for the noisy letters experiment. This time also the results confirms our previous assertions.

The second set of simulations was done with fixed structure leaning automata, introduced previously, that is, the Krinsky and Krylov automata. The results are shown in figures 7,8,9,10. We did the experiments with memory depths of 2 upto 14. Looking at the figures, we see that both automata act nearly the same, moreover, the memory depths of 5 and 11 seem best.

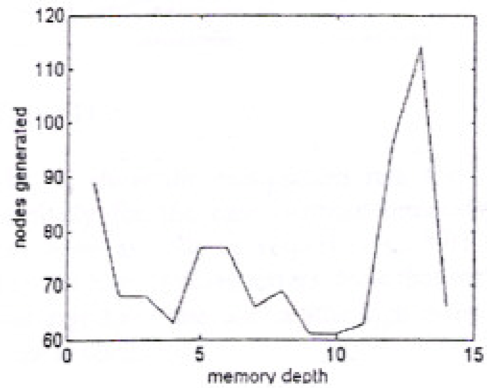

figure 7

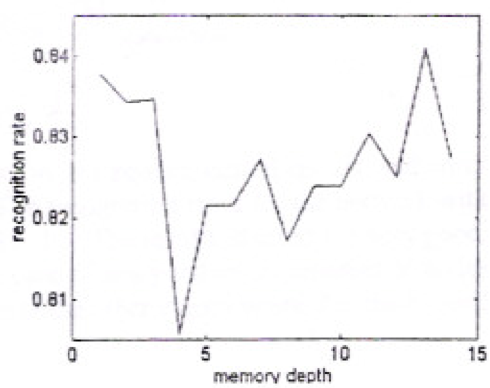

figure 8

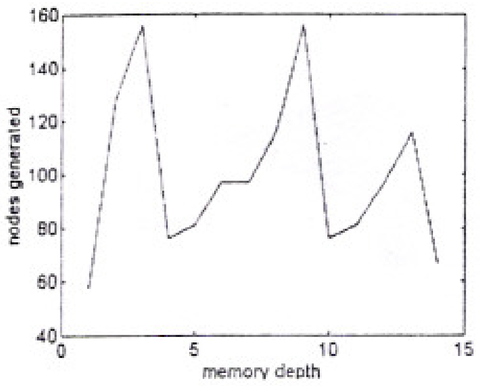

figure9

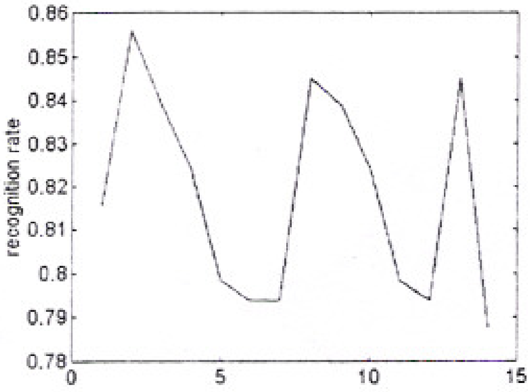

figure 10

Figures 7 and 8 are related to the Krinsky automaton, and figures 9 and 10 are related to the Krylov automaton. Comparing the results of experiments with fixed structure learning automata with that of variable structure learning automata reveales the fact that the variable structure learning automata acts more smoothly and seems a better choice.

2.Simultaneous adaptation of vigilance factor and choice parameter.

In the next set of experiments we adapted both vigilance factor and choice parameter simultaneously by variable structure automata. Thus we made totally 122 experiments for each example, 121 with choice parameter and vigilance factor constant at different values, and one with both parameters under the control of automata.

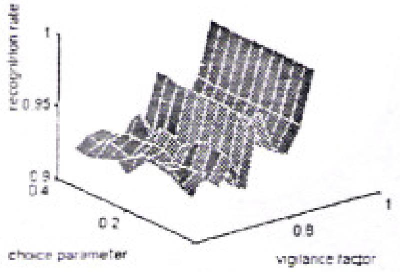

figure 11

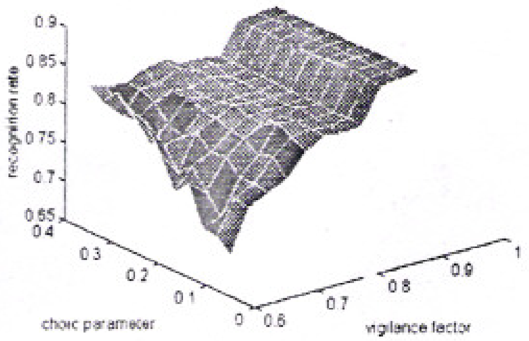

figure 12

Figures 11,12,13, show the recognition rate for circle in the square, nested spirals, and noisy letters, respectively for the case without automaton. The recognition rates for the network with two automata were as follows respectively, 0.93, 0.85, 1.0. The results obtaind are very good, especially for the case of noisy letters. Note that for the case of noisy letters the number of nodes generated and rejection rate are a little high compared to the other experiments, but the highest recognition rate is obtained.

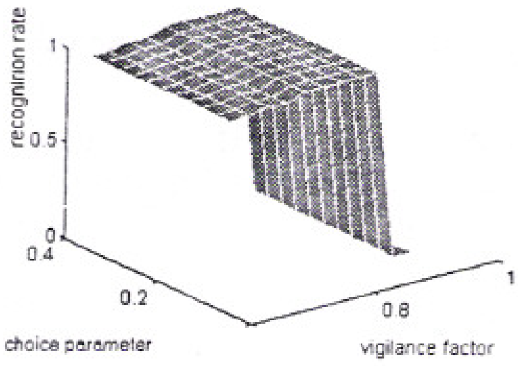

figure 13

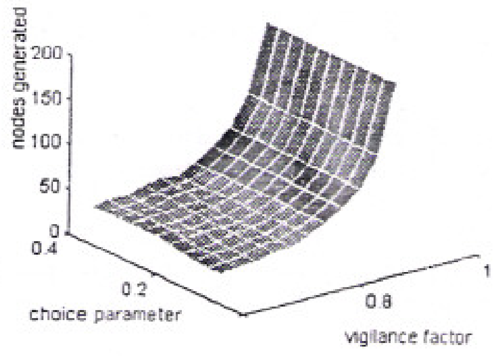

figure 14

Figures 14, 15, 16, show the nodes generated for the same experiments. The number of nodes generated for the above mentioned experiments was 41, 78, 78.

Comparing figures 11, 12, and 13 it is seen that we can not in general say what range of values give highest recognition rate. Because at the first two figures increasing vigilance factor and choice parameter increases the recognition rates, but at the third figure the recognition rate is much lower for high vigilance and choice parameter values. But at all cases the network with automaton acts excellently. Also in figures 14,15, and 16 it is seen that the number of nodes generated is sensitive to vigilance factor (the number of nodes increases with increasing vigilance factor) and it is insensitive to choice parameter.

The results of experiments show that the proposed method works very well especially in noisy environments. It generates nearly minimum number of nodes, and has nearly maximum recognition rate.

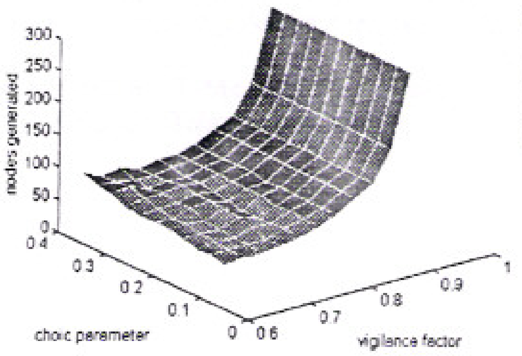

figure 15

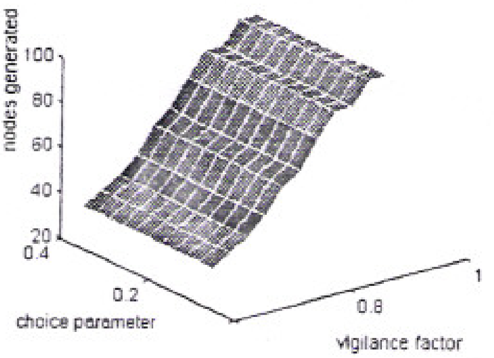

figure 16

6-Conclusion

In this paper we have augmented the ART network by learning automata in order to adapt both vigilance factor and choice parameter. The automata act at the critical point of occurrence of resonance and rejection or node generation. They dynamically change the parameters (vigilance factor and choice parameter) values and examine their actions. The simulation results show that it has inherited positive merits of both low and high values of the parameters. The new network works fastest with the most exact performance

References

1- M.A.L. Thatachar and K.S. Narendra, Learning Automata: An Introduction, Printice Hall inc, 1989.

2-G. A. Carpenter and W. D. Ross, 'ART_EMAP: A Neural Network Architecture For Object Recognition By Evidence Accumulation ', IEEE transaction on neural networks, Vol. 6, No. 4, July 1995.

3- G. A. Carpenter, S. Grossberg and J. H. Reynolds, 'A Fuzzy ARTMAP Nonparametric Probability Estimator For Nonstationary Pattern Recognition Problems' IEEE transaction on neural networks Vol.6, No. 6, November 1995.

4-G. Carpenter, S. Grossberg, N. Markuzon, J. Reynolds, and D. Rosen, 'Fuzzy ARTMAP.A Neural Network Architecture For Incremental Supervised Learning Of Analog Multidimensional Maps', IEEE transaction on neural networks, Vol.3, 1992.

5- M. J. Healy and T. P. Caudell, 'Aquiring Rule Sets As a Product Of Learning In a Logical Neural Architecture', IEEE transaction on neural networks, May 1997.

6- Fu Li and Jian Zhan,'Fuzzy Adapting Vigilance Parameter Of ARTII Neural Nets', IEEE World Congress On Computational Intelligence,Vol.3, 1994.

7- H. Beigy, M.R. Meybodi, and M. B. Menhaj, 'Adaptation Of Learning Rate In Backpropagation Using Fixed Structure Learning Automata', Proceeding of ICEE-98, Tehran, Iran.

8- M. R. Meybodi, and H. Beigy, 'New Classe Of Learning Automata Based Schemes For Adaptation Of BP Algorithm Parameters', Proceeding of 6th European Conference on Intelligent techniques and soft computing, Achen, Germany, 1998.

9- M. B. Menhaj and M. R. Meybodi, 'Application Of Learning Automata To Neural Networks', Proceeding of ICEE_95, Tehran, Iran, 1995.