

# A Cooperative Learning Model for the Fuzzy ARTMAP-Dynamic Decay Adjustment Network with the Genetic Algorithm

Shing Chiang Tan<sup>1</sup>, M.V.C. Rao<sup>2</sup>, Chee Peng Lim<sup>3</sup>

Faculty of Information Science & Technology<sup>1</sup>, Faculty of Engineering & Technology<sup>2</sup>  
Multimedia University  
Melaka Campus, Jalan Ayer Keroh Lama, Bukit Beruang, 75450 Melaka, Malaysia  
{sctan, machavaram.venkata}@mmu.edu.my  
School of Electrical & Electronic Engineering<sup>3</sup>  
University of Science Malaysia  
Engineering Campus, 14300 Nibong Tebal, Penang, Malaysia  
cplim@eng.usm.my

**Abstract.** In this paper, combination between a Fuzzy ARTMAP-based artificial neural network (ANN) model and the genetic algorithm (GA) for performing cooperative learning is described. In our previous work, we have proposed a hybrid network integrating the Fuzzy ARTMAP (FAM) network with the Dynamic Decay Adjustment (DDA) algorithm (known as FAMDDA) for tackling pattern classification tasks. In this work, the FAMDDA network is employed as the platform for the GA to perform weight reinforcement. The performance of the proposed system (FAMDDA-GA) is assessed by means of generalization on unseen data from three benchmark problems. The results obtained are analyzed, discussed, and compared with those from FAM-GA. The results reveal that FAMDDA-GA performs better than FAM-GA in terms of test accuracy in the three benchmark problems.

**Keywords:** Fuzzy ARTMAP, Dynamic Decay Adjustment, Genetic Algorithms, Cooperative Learning, Classification.

## 1 Introduction

Over the last few years, combination between artificial neural networks (ANNs) and evolutionary algorithms (EAs) has attracted a lot of attention. In fact, both ANNs and EAs cross-fertilize each other; ANNs provide a framework of accurate and exact computation whereas EAs provide a robust and efficient approach for undertaking complex optimization problem (Fogel 1995; Yao 1999). The focus of the combination between ANNs and EAs is highlighted on the accuracy rate produced by the resulting hybrid system (Cantú-Paz and Kamath 2005).

EAs, which encompass genetic algorithms (GAs), genetic programming, evolutionary programming and evolution strategies, are stochastic in nature, and they are based on populations made up of individuals with a specific behaviour similar to

biological phenomenon. They are search algorithms that are less likely to be trapped in a local minimum. Among existing EAs, the most well-known branch is GAs, which is essentially a mechanism of natural selection, genetic and evolution (Goldberg 1989). Compounded by a capability of evaluating many points in the search space simultaneously, GAs are likely to find near-optimum solution of a given problem. They are general-purpose optimization methods that require no derivative information for solving search problems. GAs have been employed primarily in two main research areas; optimization and machine learning. In machine learning, GAs can be used to aid the learning of ANNs (Whitley 1995). On the other hand, learning of ANNs is actually a process of compressing codes and refining the connection weights from a large pool of incoming data within a specific structure. In this regards, the role of GAs (or general EAs) is to aid in the learning process of ANNs by searching for a set of “optimized” weights of ANNs. Another way of hybridization between GAs and ANNs is that GAs are used to design the structure of ANNs. However, the second approach is not the focus of this paper. Instead, the main objective of our work is to improve the learning process of ANNs by searching for the weights that would eventually lead to a better network generalization performance.

Among many ANN models, the multilayer perceptron (MLP) network, which is trained with the backpropagation algorithms, has often been used as the numerical model in a hybrid system with the GAs (Lam and Leung 2004; Tsai et al. 2006). The GAs have been used to search for the number of hidden nodes as well as for the connection weights of the network. As pointed out in Taha and Ghost (1999), one major limitation of the standard MLP network is that it is difficult to adopt an incremental learning scheme in its structure. On the other hand, one ANN model that is able to overcome the issue of incremental learning and has demonstrated good performance especially for pattern classification problems is Fuzzy ARTMAP (FAM) (Carpenter et al. 1992). Indeed, FAM is a supervised model of Adaptive Resonance Theory (ART) (Carpenter and Grossberg 1987). The FAM network is an incremental learning model that has overcome the stability-plasticity dilemma (Carpenter and Grossberg 1987), i.e., it is stable enough to preserve information learned from previous data and, at the same time, flexible enough to learn new information from new data. Combination between GA and FAM is not new. However, most of the work leans towards the use of GA in selecting relevant features for assisting the learning of the FAM network (Palaniappan and Raveendran 2002; Palaniappan et al. 2002), and there is relatively little work in the literature that reports fusion between GA and FAM in machine learning and optimization.

In our work, we aim to develop a neuro-genetic system that can perform machine learning for undertaking classification problems. Unlike MLP, FAM performs incrementally learning for which it does not require handcrafting of network architecture prior to training process. Nevertheless, like MLP, FAM might likely to produce sub-optimal solutions upon the completion of the training process. Meanwhile, GAs require the initialization of population, which is predefined with arbitrary size and is of random basis, before performing the search procedure. The combination of FAM and GA amplifies the strengths and, at the same time, ameliorates the disadvantages of each constituent.

In this work, FAM learns incrementally the information by means of prototypes and the resulting templates are employed by the GA as a set of sub-optimal weights

that would guide the future search and adaptation. Such a cooperative learning scheme can improve the search efficiency of the hybrid system for obtaining good solutions, as will be demonstrated in the later section of this paper.

Learning in the original FAM network involves the formation of new prototypes and/or invariable boundary expansion of the existing prototypes. In this regards, FAM does not impart a learning scheme that resolves conflicts resulting from overlapping among the prototypes of different classes in the attribute space. In our previous work (Tan et al. 2004a), FAM is integrated with the Dynamic Decay Adjustment (DDA) algorithm for resolving conflicts during training period. In Tan et al. (2004b), an investigation of using an *ordering algorithm* (Dagher et al. 1999) to identify a fixed order of training data presentation for network learning is reported. The sequence of the training patterns is determined prior to network learning, and it is independent of any permutations of the training patterns. The resulting network (i.e., Ordered FAMDDA) learns the *ordered* training patterns for improving generalization performance.

In this paper, the focus is on the weight reinforcement of the FAMDDA network, for better generalization. A GA (Baskar et al. 2001) is proposed to co-assist in searching and adapting the weights that are learned by FAMDDA. To show the effectiveness of the hybrid system FAMDDA-GA, a performance comparison between FAM and FAMDDA, which are trained respectively with the GA (i.e., FAM-GA and FAMDDA-GA), is made. The main focus of the investigation is to ascertain that the cooperative machine learning scheme of FAMDDA-GA is more effective than that of FAM-GA towards good solutions.

This paper is organized as follows. Section 2 briefly describes FAM, FAMDDA and the GA that are applied in this work. The fusion of FAMDDA-GA is explained in section 3. To demonstrate the effectiveness of the proposed FAMDDA-GA, its performance is evaluated with three benchmark datasets and the results are compared with those from FAM-GA. All of the experimental results are analyzed and discussed in section 4. A summary of the work is given in section 5.

## 2 Fuzzy ARTMAP-Based Networks

The section presents a brief description on the operations of FAM and FAMDDA. For full description, readers can refer to Carpenter and Grossberg (1987), Carpenter et al. (1991), and Carpenter et al. (1992) for FAM; and Tan et al. (2004a) for FAMDDA.

### 2.1 Fuzzy ARTMAP (FAM)

FAM is a neural network that is constructed by a pair of Fuzzy ART modules (i.e., Fuzzy ART<sub>a</sub> and Fuzzy ART<sub>b</sub>) interconnected through a mapping field. On presentation of input patterns, FAM self-organizes and self-stabilizes information and network configuration. Such ability of FAM is reliant on the recruitment of a feedforward pass (that is usually employed in most ANN models) and a feedback pass for pattern matching. On the other hand, the network growth is also controlled by the vigilance parameter (Carpenter et al. 1992), which is a user-defined threshold. The

vigilance test evaluates the similarity between the prototype patterns of the network and the input pattern. If none of the existing neurons could satisfy the vigilance test, a new neuron is introduced, and the input is coded as its prototype pattern. Hence, subject to the vigilance criterion, the number of prototype patterns grows with time and this would engender the formation of an autonomous network topology for a problem in hand.

## 2.2 Fuzzy ARTMAP with the Dynamic Decay Adjustment Algorithm (FAMDDA)

The architecture of FAMDDA is actually similar to that of FAM. The difference is that FAMDDA incorporates a conflict-resolving facility for dealing with overlapping among prototypes of different classes. Such conflict-resolving facility is provided by the DDA algorithm (Huber and Berthold 1995), which comprises a width shrinking procedure for rectangular basis prototypes. Width adjustment of the prototype is class dependent, which distinguishes the prototype from different neighbours. Several modifications have been made in FAMDDA to perform width shrinking. In order to avoid the formation of a very thin rectangle of a prototype in FAMDDA, a user-defined minimum width threshold,  $\varepsilon_{\min}$  (Huber and Berthold, 1995), is enforced on each dimension. The learning procedure of FAMDDA is attached in the Appendix.

## 3 Reinforcement Learning of FAMDDA with GA

GAs are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic (through crossover and mutation operations). The basic concept of GAs is designed to simulate processes in natural system necessary for evolution. The combination of FAMDDA and GA (i.e., FAMDDA-GA) amplifies the strengths of both constituents and at the same time, it complements the weaknesses of each individual entity. In our work, the GA participates in enhancing the learning capability of the FAMDDA network, by searching and adapting its weights. Fig. 1 shows the overall procedure of the cooperative learning model of FAMDDA-GA. Training starts with the weight learning process of FAMDDA within a predefined number of maximum epochs. FAMDDA performs supervised learning based on the given training patterns. The architecture as well as the weights of the network are identified autonomously. It is noted that all weights of Fuzzy ART<sub>a</sub> that represent the essence of input data space are grouped as a Pittsburg chromosome  $z_{\theta}$  for search mechanism and adaptation in the GA environment. In fact, a chromosome in the GA environment represents the weights of different classes in the data space in the network environment. On the other hand, both weights of Fuzzy ART<sub>b</sub> and mapping field are secured from evolution. Such arrangement is intended to reinforce weights of Fuzzy ART<sub>a</sub> in the GA environment without affecting its behavior link with Fuzzy ART<sub>b</sub>, upon returning to the network environment. A finite number of chromosomes are generated in the GA environment according to a modified function (Baskar et al. 2001)

$$z_i = z_\theta + \text{RMF} \cdot \text{rand}(1, n) \quad (1)$$

where  $z_i$  is the  $i$ -th replicated chromosome of  $z_\theta$ ;  $\text{RMF} \in [0,1]$  stands for the range multiplication factor;  $\cdot$  represents scalar multiplication operation; and,  $\text{rand}(1, n) \in [0,1]$  is a uniformly distributed random vector.

1. Weight initialization and nodes self-organization using FAMDDA.
  2. Generate chromosomes (Pittsburgh Approach).  
Repeat
  3. Compute fitness value of each chromosome.
  4. Apply Roulette-Wheel selection.
  5. Generate new generation through crossover and mutation.
- Until terminating condition has been satisfied.

**Fig. 1.** The overall training procedure of FAMDDA-GA.

Each chromosome is converted into the Fuzzy  $\text{ART}_a$  module and the fitness of the network is evaluated in terms of accuracy, by presentation of training patterns. Once the fitness of all chromosomes is calculated, the GA applies Roulette-Wheel to select chromosomes. The genetic operators (i.e., crossover and mutation) are applied to process the selected chromosomes. In this study, the arithmetic crossover (Baskar et al. 2001) that can deal with a pair of selected real-coded chromosomes (i.e.,  $z_a, z_b$ ) is applied, as follows.

$$z' = \lambda z_a + (1 - \lambda) z_b \quad (2)$$

$$z'' = (1 - \lambda) \lambda z_a + \lambda z_b \quad (3)$$

where  $\lambda \in [0,1]$  is a uniformly distributed random variable;  $z'$  and  $z''$  are the offsprings.

A dynamic mutation as described in Baskar et al. (2001) is adopted in our work. For a given chromosome  $z$ , if the element  $z_k$  is selected for mutation, the resulting chromosome is  $z' = [z_1 \cdots z_k' \cdots z_n]$ , where  $z_k'$  is randomly selected from two possible choices

$$z_k' = z_k + (z_k^U - z_k) \cdot r \cdot (1-t/T)^d \quad \text{or} \quad z_k' = z_k - (z_k - z_k^L) \cdot r \cdot (1-t/T)^d \quad (4)$$

where  $z_k^U / z_k^L$  are the upper/lower bounds of the element  $z_k$ ;  $r \in [0,1]$  is a random number;  $T$  is the maximum generation number; and,  $d$  is a parameter that determines the degree of non-uniformity (usually assumed as 2).

After crossover and mutation have been applied, new set of chromosomes (offspring) are obtained. They are new set of individuals that forms a new population. The process of fitness evaluation, selection, and genetic operations on a population of chromosomes is repeated until one of the stopping criteria has been satisfied, i.e., either an achievement of 100% recognition rate of training patterns by one of the chromosome or a maximum setting of 10 generations (i.e.,  $T=10$ ), has been reached.

If second stopping criterion is fulfilled, the chromosome of the current population that has the highest fitness value is identified as the weights of the Fuzzy ART<sub>a</sub> module.

## 4 Experimental Results and Discussion

To demonstrate the effectiveness of the proposed FAMDDA-GA model, a series of experiments were conducted using three benchmark datasets; pima Indian diabetes (PID), Australian credit approval (AUS), and heart (HEA). All these datasets are available from the UCI machine-learning repository (Blake and Merz 1998). The PID, AUS and HEA datasets are binary-class problems with 768, 690, and 270 samples, and have 8, 14, and 13 input attributes, respectively.

In each experiment, the dataset is divided into training and test sets. The FAMDDA-GA model was trained using 50% randomly selected data while the remaining data were used for evaluating the network’s performance. The FAMDDA-GA model underwent two different modes of training: FAMDDA and GA. In the FAMDDA environment, the system was trained at 1 epoch with the “default” parameter settings: fast learning rate,  $\beta = 1$ ; minimum width,  $\varepsilon_{\min} = 0.1$ ; and ART<sub>a</sub> baseline vigilance,  $\bar{\rho}_a = 0.0$ . On the other hand, in the GA environment, search and adaptation of the system are driven by parameters which include the number of replicate chromosomes (i.e.,  $n=10$ ), RMF=0.30,  $T=10$ , crossover rate 0.30, and mutation rate 0.01. For each classification task, eight independent experiments were carried out. The network performances, in terms of the test accuracy rate and the number of nodes, were averaged. To compare the performance of FAMDDA-GA with that of its counterpart (FAM-GA) as well as with the ordinary classifiers FAMDDA and FAM, the same experimental procedure was adopted. Tables 1 and 2 summarize the overall results obtained from the three datasets.

**Table 1.** Performance comparison between FAM-GA and FAMDDA-GA, in terms of test accuracy and number of nodes. The results are averages of eight runs, and their respective standard deviation is given in parenthesis. The  $p$ -values are the results from the bootstrap hypothesis tests between FAM-GA and FAMDDA-GA.

Dataset	Accuracy (%)			Size (Nodes)		
	FAMDDA-GA	FAM-GA	$p$ -value	FAMDDA-GA	FAM-GA	$p$ -value
PID	72.55 (2.24)	68.53 (2.45)	0.006	15.6 (1.4)	13.9 (1.9)	0.03
AUS	78.89 (2.57)	72.61 (2.27)	0	15.2 (3.1)	13.0 (2.6)	0.081
HEA	77.02 (2.33)	71.50 (4.63)	0.010	9.3 (1.5)	8.5 (1.4)	0.151

**Table 2.** Performance comparison between the FAM and FAMDDA classifiers, in terms of test accuracy and number of nodes. The results are averages of eight runs, and their respective standard deviation is given in parenthesis. The  $p$ -values are the results from the bootstrap hypothesis tests between the classifier and its GA-based network ( $a$  denotes FAM vs. FAM-GA; and  $b$  denotes FAMDDA vs. FAMDDA-GA).

Dataset	Accuracy (%)			Size (Nodes)		
	FAMDDA	FAM	$p$ -value	FAMDDA	FAM	$p$ -value
PID	70.11	64.36	0.008 <sup>a</sup>	16.2	13.9	0.172 <sup>a</sup>
	(2.26)	(3.35)	0.007 <sup>b</sup>	(2.6)	(1.9)	0.403 <sup>b</sup>
AUS	72.25	67.61	0.006 <sup>a</sup>	15.2	13.0	0.513 <sup>a</sup>
	(2.42)	(3.58)	0 <sup>b</sup>	(3.1)	(2.6)	0.496 <sup>b</sup>
HEA	69.28	62.97	0.003 <sup>a</sup>	8.7	9.0	0.685 <sup>a</sup>
	(5.33)	(4.46)	0.018 <sup>b</sup>	(1.9)	(2.0)	0.271 <sup>b</sup>

From Table 1, we can see that FAMDDA-GA produces higher accuracy rate than FAM-GA. The comparison between the generalization performance of FAMDDA-GA and FAM-GA can be made by a bootstrap hypothesis test (Efron, 1979) with a significance level of 0.05. One advantage of the bootstrap method is that it does not rely on the assumption that the samples must be drawn from a normal distribution. In this regards, the null hypothesis states no difference between the test accuracy rate of FAM-GA and FAMDDA-GA whereas the alternative hypothesis claims that the test accuracy rate of FAM-GA is lower than that of FAMDDA-GA. The  $p$ -values of the corresponding tests are tabulated Table 1. We can observe that all  $p$ -values of the test for the accuracy between FAMDDA-GA and FAM-GA are less than 0.05. This indicates that the classification performance of FAM-GA is statistically lower than that of FAMDDA-GA.

Table 1 also shows the comparison between the network size of FAMDDA-GA and FAM-GA. The alternative hypothesis of the bootstrap hypothesis test for network size claims that FAM-GA has a smaller network size than that of FAMDDA-GA in the respective benchmark studies. Note that the  $p$ -values for the network size in the AUS and HEA case studies are greater than 0.05, whereas the  $p$ -value for the PID case study is less than 0.05. This indicates that, statistically, there is no significant difference between the network size of FAM-GA and FAMDDA-GA in both AUS and HEA studies. But, it is on the contrary for the PID case study. However, this observation from the PID case study does not affect the finding that the cooperative learning of FAMDDA-GA has yielded better generalization performance (than FAM-GA) in these three studies.

The performances between the FAMDDA/FAM classifiers and their respective counterparts (FAMDDA-GA/FAM-GA) are also compared with the same claims of alternative hypothesis aforementioned. From Table 2, we can observe that all  $p$ -values of the test for the accuracy between FAMDDA (FAM) and FAMDDA-GA (FAM-GA) are smaller than 0.05 whereas all  $p$ -values of the test for comparing network size FAMDDA (FAM) and FAMDDA-GA (FAM-GA) are greater than 0.05. These results indicate that the generalization performances of individual classifiers are, respectively, lower than their GA-version counterparts, and the difference in the network sizes is statistically insignificant.

## 5 Summary

In this paper, a neuro-genetic system which integrates FAMDDA and the GA for performing classification is proposed. The performance of the propose FAMDDA-GA was accessed with three benchmark datasets. In order to show the effectiveness of FAMDDA-GA, its performance is compared with that of FAM-GA, as well as the ordinary classifier of FAMDDA and FAM. The results reveal that the generalization performance of FAMDDA-GA is better than that of FAM-GA. As for future work, a study on the possibility of integrating FAMDDA with another variant of EA-based global search algorithm, e.g. differential evolution, will be conducted. Another direction of future work is to investigate the possibility of including a local search into the existing framework for refining the solutions that have been obtained via global search.

## References

- Baskar, S., Subraraj, P., Rao, M.V.C. (2001), "Performance of hybrid real coded genetic algorithms," *International Journal of Computational Engineering Science*, vol. 2, pp. 583-601.
- Blake, C. and Merz, C. (1998), UCI Repository of Machine Learning Databases, URL <http://www.ics.uci.edu/~mlearn/MLRepository.html>
- Cantú-Paz, E., Kamath, C. (2005), "An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems," *IEEE Trans. Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 35, pp. 915-927.
- Carpenter, G. A. and Grossberg, S. (1987), "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision, Graphics and Image Processing*, vol. 37, pp. 54-115.
- Carpenter, G. A., Grossberg, S., and Rosen, D. (1991), "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, pp. 759-771.
- Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J., and Rosen, D. (1992), "Fuzzy ARTMAP: A neural network architecture for incremental learning of analog multidimensional maps," *IEEE Trans. Neural Networks*, vol. 3, pp. 698-713.
- Dagher, I, Georgiopoulos, M, Heileman, G.L., Bebis, G. (1999), "An ordering algorithm for pattern presentation in fuzzy ARTMAP that tends to improve generalization performance," *IEEE Trans Neural Networks*, vol. 10, pp. 768-778.
- Efron, B. (1979), "Bootstrap methods: another look at the jackknife," *The Annals of Statistics*, vol. 7, pp. 1-26.
- Fogel, D. B. (1995), *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Piscataway, NJ: IEEE Press.
- Goldberg, D.E. (1989), *Genetic Algorithms in Search Optimization and Machine Learning*. Reading, MA: Addison-Wesley.
- Huber, K.-P. and Berthold, M.R. (1995), "Building precise classifiers with automatic rule extraction," in *Proc. of the IEEE Int. Conf. Neural Networks*, vol. 3, pp. 1263-1268.
- Lam, H.K., and Leung, F.H.F. (2004), "Digit and command interpretation for electronic book using neural network and genetic algorithm," *IEEE Trans. Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 34, pp. 2273 – 2283.

- Lim, C.P. and Harrison, R.F. (1997), "An incremental adaptive network for on-line supervised learning and probability estimation," *Neural Networks*, vol. 10, 1997, pp. 925-939.
- Palaniappan, R., and Raveendran, P. (2002), "Individual identification technique using visual evoked potential signals," *Electronics Letters*, vol. 28, pp. 1634-1635.
- Palaniappan, R., Raveendran, P., and Omatu, S. (2002), "VEP optimal channel selection using genetic algorithm for neural network classification of alcoholics," *IEEE Trans. Neural Networks*, vol. 13, pp. 486-491.
- Taha, I.A., and Ghosh, J. (1999), "Symbolic interpretation of artificial neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 11, pp. 448-463.
- Tan, S.C., Rao, M.V.C., and Lim, C.P. (2004a), "An adaptive conflict-resolving classifier," *WSEAS Trans. on Information Science and Applications*, vol. 1, pp. 113-117.
- Tan, S.C., Rao, M.V.C., and Lim, C.P. (2004b), "An adaptive fuzzy min-max conflict-resolving classifier," in Proceedings of the 9<sup>th</sup> Online World Conference on Soft Computing in Industrial Applications, WSC9, 20 September – 8 October 2004.
- Tsai, J.-T., Chou, J.-H., and Liu, T.-K. (2006), "Tuning the structure and parameters of a neural network by using hybrid taguchi-genetic algorithm," *IEEE Trans. Neural Networks*, vol. 17, pp. 69-80.
- Whitley, D. (1995), "Genetic algorithms and neural networks," in *Genetic Algorithms Engineering and Computer Science*, G. Winter, J. Periaux, M. Galan, and P. Cuesta, Eds. New York: Wiley, pp. 191-201.
- Yao, X. (1999), *Evolutionary Computation: Theory and Applications*, Singapore: World Scientific.

## Appendix

In FAMDDA, the operation of Fuzzy ART<sub>a</sub> module is modified for resolving conflicts resultant from the overlapping among prototypes of different classes. A set of *reference vector*  $w^r$  (Lim and Harrison 1997) is introduced to each prototype  $j$  in  $F_2^a$  (Carpenter et al. 1992). Initially,  $w_j^r$  is equal to zero. In general, as learning takes place,  $w_j^r$  is updated. In addition, the width condition of all dimension of each  $F_2^a$  prototype is initialized as zero (i.e.,  $S_j = 0$ ) to indicate "infinite volume". If  $x$ , which represents the weights of the  $M$  – dimension (or  $2M$  – dimension with complement coding) winning prototype falls in the region formed by the prototype of conflicting classes, a shrinking procedure then ensues. In this regard, the width of the conflicting prototypical region at arbitrary dimension is shrunk. In this regard, shrinking of an existing finite dimension is preferred for not losing "infinite volume" of other infinite dimensions. Nevertheless, to avoid the formation of a very thin rectangle, a user-defined minimum width threshold,  $\varepsilon_{\min}$ , is enforced on each finite dimension. The shrinking procedure is applied successively between the winning prototype and other conflicting prototypes. Three cases of width shrinking as in Huber and Berthold (1995) are considered. First, if the existing finite dimensions of the conflicting prototype  $q$  can be shrunk without falling below a pre-set  $\varepsilon_{\min}$ , the one with the smallest loss in volume ( $\gamma_{best,k}$ ) is chosen, as follows.

$$\gamma_{best,k} = \min \left\{ |w_{qk}^r - x_k| : \forall 1 \leq i \leq n, i \neq k : \left( \frac{\lambda_{qk} - |w_{qk}^r - x_k|}{\lambda_{qk}} \leq \frac{\lambda_{qi} - |w_{qi}^r - x_i|}{\lambda_{qi}} \right) \right. \\ \left. \wedge (\lambda_{qk} \geq \varepsilon_{k,\min}) : S_{qi}, S_{qk} = 1 \right\} \quad (5)$$

where  $\lambda_q$  represents the current width of the prototype  $q$ . If the first option is not satisfied, the following options are considered, i.e., either one of the remaining infinite dimensions is shrunk ( $\gamma_{\max,l}$ ),

$$\gamma_{\max,l} = \max \left\{ |w_{ql}^r - x_l| : S_{ql} = 0 \right\} \quad (6)$$

or, the width of a conflicting prototype is shrunk

$$\gamma_{\min,m} = \min \left\{ |w_{qm}^r - x_{qm}| : \forall 1 \leq i \leq n, i \neq m : \left( \frac{\lambda_{qm} - |w_{qm}^r - x_{qm}|}{\lambda_{qm}} \leq \frac{\lambda_{qi} - |w_{qi}^r - x_{qi}|}{\lambda_{qi}} \right) \right. \\ \left. : S_{qi}, S_{qk} = 1 \right\} \quad (7)$$

Width shrinking using Eq. (6) is selected if  $\gamma_{\max,l} > \gamma_{\min,m}$ . The learning algorithm of FAMDDA in one epoch is described in Fig. 2.

```
% Training of one epoch
% 1. Consider all ordered complement-coded (Carpenter et al. 1992) samples (A,c) c - class
for all samples A do
  if ARTa vigilance test = TRUE & ARTb vigilance test = TRUE
    if Mapfield link established = TRUE then
      % 2. Sample covered by winning prototype J of the F2a layer (Carpenter et al. 1992)
      wJa(new) = β(A ∧ wJa(old)) + (1 - β)wJa(old);
      NJ = NJ + 1;
      (wJr)new = (wJr)old + 1/NJ (A - (wJr)old)
      % β - learning rate; wJ - the weight of prototype J
    % 3. Commit new neuron
    else
      NJ = 1; % NJ - number of sample covered by prototype J
      wJr(new) = A; (wJr)new = A;
    end
  % 4. Shrink width of conflicting neurons: (according to Eq. (5), (6), or (7))
  end
end % for all samples A
```

**Fig. 2.** The operation of the FAMDDA network.