



Using genetic algorithm to select the presentation order of training patterns that improves simplified fuzzy ARTMAP classification performance

Ramaswamy Palaniappan ^{a,*}, Chikkanan Eswaran ^b

^a Department of Computing and Electronic Systems, University of Essex, Colchester, United Kingdom

^b Faculty of Information Technology, Multimedia University, Cyberjaya, Malaysia

ARTICLE INFO

Article history:

Received 9 May 2007

Received in revised form 27 February 2008

Accepted 11 March 2008

Available online 18 March 2008

Keywords:

Fuzzy ARTMAP

Genetic algorithm

Individual identification

Min–max ordering

Visual evoked potential

Voting strategy

ABSTRACT

The presentation order of training patterns to a simplified fuzzy ARTMAP (SFAM) neural network affects the classification performance. The common method to solve this problem is to use several simulations with training patterns presented in random order, where voting strategy is used to compute the final performance. Recently, an ordering method based on min–max clustering was introduced to select the presentation order of training patterns based on a single simulation. In this paper, another single simulation method based on genetic algorithm is proposed to obtain the presentation order of training patterns for improving the performance of SFAM. The proposed method is applied to a 40-class individual classification problem using visual evoked potential signals and three other datasets from UCI repository. The proposed method has the advantages of improved classification performance, smaller network size and lower training time compared to the random ordering and min–max methods. When compared to the random ordering method, the new ordering scheme has the additional advantage of requiring only a single simulation. As the proposed method is general, it can also be applied to a fuzzy ARTMAP neural network when it is used as a classifier.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Fuzzy ARTMAP (FAM) [1] and simplified fuzzy ARTMAP (SFAM) [2] belong to a special class of neural networks (NNs) which are capable of incremental learning. In the fast learning mode, these networks have lower training time compared to other NN architectures like Multilayer Perceptron. SFAM and FAM have been used in numerous classification problems [1–6]. FAM structure has three modules: Fuzzy ARTa, Fuzzy ARTb and Inter ART. SFAM differs from FAM in that its main purpose is for classification and as such, does not have Fuzzy ARTb, which becomes redundant for this purpose.

The presentation order of training patterns affects the classification (i.e. generalisation) performance of SFAM. To solve this problem, SFAM is trained several times using training patterns presented in random order (i.e. permutations of the training patterns) and then the predicted class of the test patterns are stored. Majority votes are used to determine the final class prediction for the test patterns [1]. It is also customary to state the average classification of test patterns from all the simulations in

addition to the voting results. To solve the problem of having to run many simulations, a single simulation method based on min–max clustering was proposed [3]. For a c -class problem, the method works by ordering the c training patterns that are maximally distant in the training feature space. Next, for the rest of the patterns, the method orders training patterns that are minimally distant from these c patterns. Hence, it is known as min–max ordering.

In this paper, a method that uses genetic algorithm (GA) [7] to select the presentation order of training patterns is proposed. The method works by using the selection, mutation and inversion operators in GA to select the presentation order of training patterns that maximises the SFAM classification performance. Once the order is selected, only a single SFAM training simulation (similar to min–max ordering) will be required for classification of test patterns. The performance of the proposed technique is compared with training patterns ordered by min–max and random ordering using classifications of visual evoked potential (VEP) patterns to identify individuals [8]. In addition, three data sets from UCI repository [9] are also used to measure the performances of these ordering methods. From this point onwards, these three methods will be denoted simply as GA method, random ordering method and min–max method. All the discussions in this paper on SFAM could be equally applied to FAM with the condition that FAM is used as a classifier.

* Corresponding author.

E-mail addresses: rpalan@essex.ac.uk (R. Palaniappan), eswaran@mmu.edu.my (C. Eswaran).

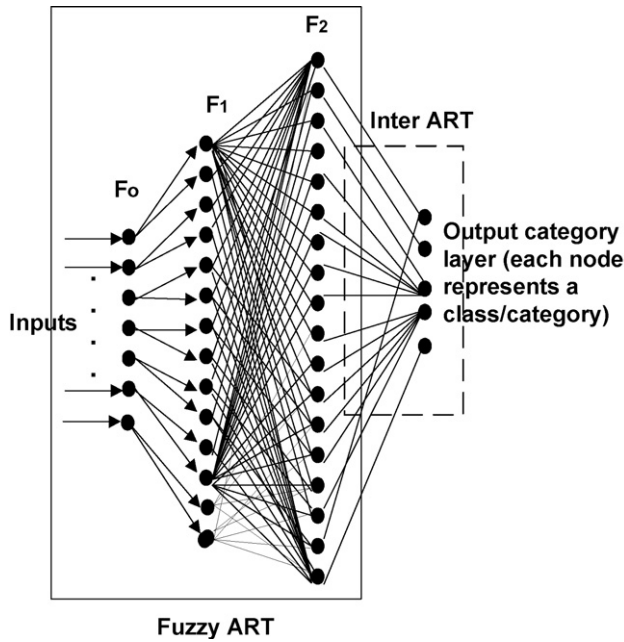


Fig. 1. Architecture of SFAM.

2. Simplified fuzzy ARTMAP

Fig. 1 shows the architecture of SFAM. It consists of two modules (Fuzzy ART and Inter ART) that create stable recognition categories in response to sequence of input patterns. During supervised learning, Fuzzy ART receives a stream of input features representing the pattern that map to the output classes in the category layer. Inter ART module works by increasing the vigilance parameter (VP) of Fuzzy ART by a minimal amount to correct a predictive error in the output category layer. VP calibrates the minimum confidence that Fuzzy ART must have in an input vector in order for Fuzzy ART to accept that category, rather than search for a better one through an automatically controlled process of hypothesis testing. Lower values of VP denote increased generalisation ability and lower category formations, while larger values of VP induce more categories [1].

2.1. Fuzzy ART

Fuzzy ART module has three different layers: \$F_0\$, \$F_1\$ and \$F_2\$ consisting of nodes. Input layer \$F_0\$ nodes represent a current input vector and \$F_0\$ activity vector is denoted by \$I = (I_1, \dots, I_M)\$, with each component \$I_i\$ in the interval \$[0,1]\$, \$I = 1, \dots, M\$.

Proliferation of categories in Fuzzy ART is avoided if the inputs are normalized using the method of complement coding. Therefore, the complement coded input \$I\$ to the field \$F_1\$ is the \$2M\$ dimensional vector

$$I = (a, a^c) \quad \text{where} \quad a_i^c = 1 - a_i \quad (1)$$

\$F_1\$ layer nodes are connected to output layer \$F_2\$ nodes through a weight vector. For each \$F_2\$ category node \$j\$ (\$j = 1, \dots, N\$), there is a weight vector associated with layer of \$F_1\$ nodes, \$w_j = (w_{j1}, \dots, w_{j2M})\$ of adaptive weights. The initial condition is

$$w_{j1}(0) = \dots = w_{j2M}(0) = 1 \quad (2)$$

which means that each category is uncommitted.

For each input \$I\$ and \$F_2\$ node \$j\$, the choice function \$T_j\$ is defined by

$$T_j(I) = \frac{|I \wedge w_j|}{\alpha + |w_j|} \quad (3)$$

where the fuzzy AND operator \$\wedge\$ is defined by

$$(p \wedge q)_i = \min(p_i, q_i) \quad (4)$$

and the norm \$|\cdot|\$ is defined by

$$|p| = \sum_{i=1}^M |p_i| \quad (5)$$

for any \$M\$-dimensional vectors \$p\$ and \$q\$. For simplicity, let, \$T_j(I)\$ in (3) be denoted as \$T_j\$ when the input \$I\$ is fixed. A category choice is made when one \$F_2\$ node becomes active at a given time. The category choice is indexed by \$J\$, where

$$T_J = \max\{T_j : j = 1, \dots, N\} \quad (6)$$

If more than one \$T_j\$ is maximal, the category with a smaller index is chosen.

Resonance occurs if the match function, \$|I \wedge w_J|/|I|\$ of the chosen category meets the vigilance criterion:

$$\frac{|I \wedge w_J|}{|I|} \geq \rho, \quad (7)$$

where \$\rho\$ is VP. With resonance, learning starts, as explained below. Mismatch reset occurs if condition in (7) is not met and then the value of the choice function \$T_j\$ is set to 0 and a new index \$J\$ is chosen by (6).

The search process continues until the chosen \$J\$ satisfies (7). Once the search is completed, the weight vector \$w_j\$ is updated according to the equation

$$w_j^{(\text{new})} = (I \wedge w_j^{(\text{old})}) \quad (8)$$

assuming that fast learning is used.

2.2. Inter ART

The Inter ART module will create mappings between the Fuzzy ART layer \$F_2\$ and output category layer to correctly learn to predict the classification patterns. For all the input patterns presented, it creates a dynamic weight link that consists of a many to one or one to one mapping between the \$F_2\$ and output category layers. Whenever a one to many mapping from the \$F_2\$ and output category layers happens, an error correcting mechanism called match tracking occurs which will increase the VP to a value slightly higher than

$$\frac{|I \wedge w_J|}{|I|} \quad (9)$$

where \$J\$ is the index of the active \$F_2\$ node.

Match tracking avoids confusing in predictions. When this occurs, Fuzzy ART search leads either to another category that correctly predicts the target or to an uncommitted new category and the dynamic weight link between the Fuzzy ART and Inter ART modules (\$F_2\$ and output category layers) are updated. After this, VP is set back to the earlier (baseline) vigilance parameter value. This process is continued until all the training patterns have been presented.

2.3. Testing stage and effect of input pattern order during training

The testing stage works in the same principle except that there will be no match tracking. This is since the input presented to Fuzzy ART will output values at \$F_2\$ layer. The node at \$F_2\$ with highest value will trigger the link in the Inter ART module to establish the appropriate node in output category layer, which will denote the predicted class.

The order of input patterns during training affects the formation of the nodes at \$F_2\$ layer of Fuzzy ART module, even for a fixed VP.

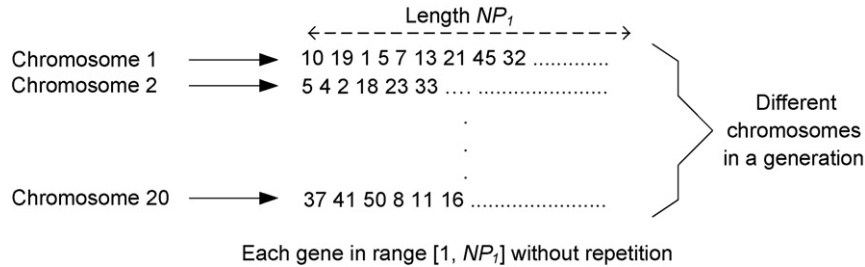


Fig. 2. Initial GA population.

Hence, the different presentation order of training input patterns will affect the SFAM network size (i.e. the number of F_2 nodes) and as a result, the classification performance will vary too. Random ordering and min–max methods have been suggested to solve this problem [1]. In this paper, we propose another superior method using GA to select the pattern ordering.

3. Pattern ordering using GA method

The overall methodology could be divided into two separate phases. In the first phase, either GA or min–max (as comparison) was used to order the input training patterns. The second phase involved testing the performance of SFAM for all the ordering methods (i.e. our proposed method, min–max and random). This second phase was conducted to show the improvement in SFAM performances when trained by training patterns ordered by GA as compared to min–max and random ordering.

Initially, the available dataset was split into three sets: datasets 1, 2 and 3. In the pattern ordering phase, GA was used with datasets 1 and 2 where GA was run for 100 generations with the fitness of 20 chromosomes given by SFAM training and testing for each chromosome (hence 20 times for a generation). SFAM was trained with VP value of 0. When this was completed, the presentation order of training patterns has been selected, and GA was not used anymore. Similarly, dataset 1 was used by the min–max method in the pattern ordering phase to order the presentation of training patterns. After order selection, the min–max method was not used anymore.

In the next phase of performance testing, the presentation orders selected by GA and min–max methods were used. For random ordering, since there was no pattern ordering phase, the simulation was repeated 20 times with random permutations of the training patterns and voting strategy¹ as suggested in [1] was used to predict the final class of the test patterns. For this second phase, we conducted classification experiments with dataset 1 for SFAM training and dataset 3 for SFAM testing. Dataset 2 was not used here (to be fair) as only GA has used this dataset before in the earlier phase. In other words, all the different ordering methods (proposed, min–max and random) were trained and tested with the same datasets.

This may seem confusing as SFAM training and testing were involved in both phases for GA method but it should be noted that the SFAM training and testing used in pattern ordering phase was different from the SFAM training and testing used in performance testing phase. The SFAM training (conducted for 2000 times as we had 100 generations and population size of 20) in pattern ordering phase was conducted with the aim to order the presentation of training patterns. But the SFAM training in performance testing phase was conducted with the aim to test the classification performance of SFAM with the selected presentation order of training patterns. So, in the second performance testing phase, for the presentation order selected by GA and min–max methods, only

one SFAM training and testing was completed. Neither GA nor min–max methods was used in this second phase, only the presentation order selected earlier by GA or min–max methods was used. For random ordering, 20 SFAM training and testing were conducted in the performance testing phase.

The steps involved in the GA are as follows.

3.1. Initialisation

The number of genes in each chromosome was set to the number of patterns, NP_1 in dataset 1. Each gene was randomly set from integer values of 1 to this number, without repetition (as shown in Fig. 2). Twenty similar chromosomes were generated, which represent the population.

3.2. Fitness value

To calculate the fitness value of each chromosome, SFAM was trained (using VP set to zero to speed training and minimise overfitting) by patterns in dataset 1 using the presentation order given by the chromosome. The trained SFAM was tested with patterns from dataset 2 and the fitness value was computed as the percentage of correctly classified patterns over the total tested patterns.

The fitness function for each chromosome was therefore

$$\text{fitness} = \frac{\text{correctly_recognised}}{NP_2}, \quad (10)$$

where NP_2 is the total number of pattern in dataset 2.

3.3. Selection, mutation and inversion operators

Two selection (reproduction) methods, namely, roulette wheel and tournament selection methods were used to select the chromosomes for the next generation. Half of the population (i.e. 10 chromosomes) were selected by each method. Here, the tournament selection method works by selecting the best from three randomly chosen chromosomes. This step was repeated 10 times to obtain 10 chromosomes. Tournament selection would be useful to retain the chromosomes with high fitness values, but roulette wheel selection was necessary to avoid premature convergence, i.e. to avoid GA from converging too quickly with suboptimal chromosomes. The roulette wheel method works by selecting chromosomes with a higher probability of survival. In general, higher fitness chromosomes will have a higher chance of survival [7].

For mutation, two genes in one chromosome were randomly chosen and they were swapped if

$$r < p_m, \quad (11)$$

where r is a random number generated in the range [0,1] and p_m is the mutation rate. This mutation procedure used here is different from the common method of applying the mutation operation because of the nature of the problem, where genes in every

¹ Each random ordering will predict a certain class. The final output was based on the majority vote of the different classes.

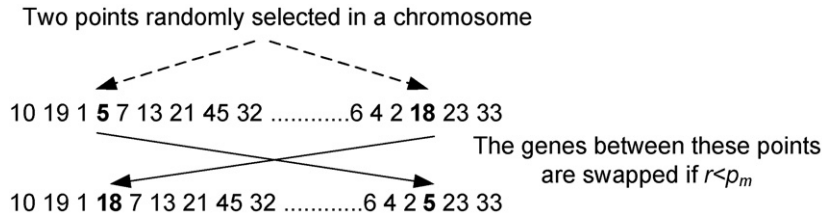


Fig. 3. Mutation operation as used here.

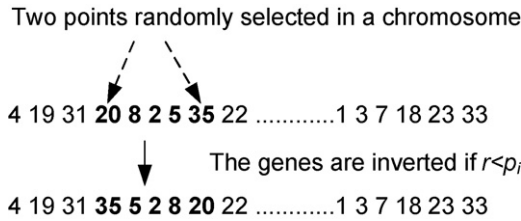


Fig. 4. Inversion operation as used here.

chromosome must be permutations of genes in other chromosomes. This is also the reason why crossover operator was not used here. Fig. 3 shows an example of this mutation operation.

Inversion operator was used to inverse the genes in the chromosomes. Here, a two-point inversion operator was used. Two points were randomly chosen and genes from a randomly chosen chromosome were inverted between these two points if

$$rr < p_i, \tag{12}$$

where rr is a random number generated in the range $[0,1]$ and p_i is the inversion rate. An example of the inversion operation is shown in Fig. 4.

Mutation and inversion operators were applied for certain number of times based on the probability, p , which was initially set at 0.9. The high initial probability was chosen because of the lack of crossover operator. The probability was gradually reduced with increasing number of generations using

$$p(n) = 0.9 * \left(\frac{1 - n}{\text{max generation}} \right), \tag{13}$$

where n is the current generation.

3.4. Iterate

Steps 2 and 3 were repeated until a maximum generation number of 100 was reached. The overall best chromosome (with highest fitness value) was then stored. Since the best chromosome selected by the GA depends on the initial search space, GA simulation was repeated five times and the chromosome that has the fitness value closest to the average of the five best chromosomes' fitness values was the one that was actually stored.

Table 1
Summary of GA parameters

Coding of genes	Integer coding in the range $[1, NP_1]$
Fitness function	SFAM classification performance (training dataset 1, testing dataset 2)
Population size	20
Number of genes	NP_1 (depending on the number of pattern in dataset 1)
Reproduction (selection)	Roulette wheel (50% of population) and tournament selection (50% of population)
Mutation type and rate	Two bit swap in one chromosome, initial probability = 0.9 (decreases with generations)
Inversion type and rate	Inversion between two randomly selected points, initial probability = 0.9 (decreases with generations)
Convergence	100 generations

This chromosome represents the GA selected presentation order of training patterns for SFAM. Fig. 5 shows the steps involved in the GA method, while Table 1 summarizes the GA parameters.

4. Experimental study

An experimental study is conducted to show the superior performance of the GA method compared to the random ordering and min–max methods. For this purpose, the dataset used in an earlier work to identify individuals [8] was used. In addition, three datasets, namely, wine, glass and iris, from UCI repository [9] were also used.

4.1. Individual classification data

The details of this data set will be briefly repeated here. VEP signals were extracted from 61 channels from 40 subjects while seeing a picture. These pictures were objects chosen from Snodgrass and Vanderwart picture set [10]. These pictures represent common black and white objects, such as, for instance, airplane, banana, and ball (a few examples are shown in Fig. 6). These were chosen according to a set of rules that provides consistency of pictorial contents and have been standardized based on the variables of central relevance to memory and cognitive processing. These objects had definite verbal labels, i.e. they could be named.

The subjects were asked to remember or recognize the stimulus. Stimulus duration of every picture was 300 ms with an inter-trial interval of 5.1 s. All the stimuli were shown using a display located

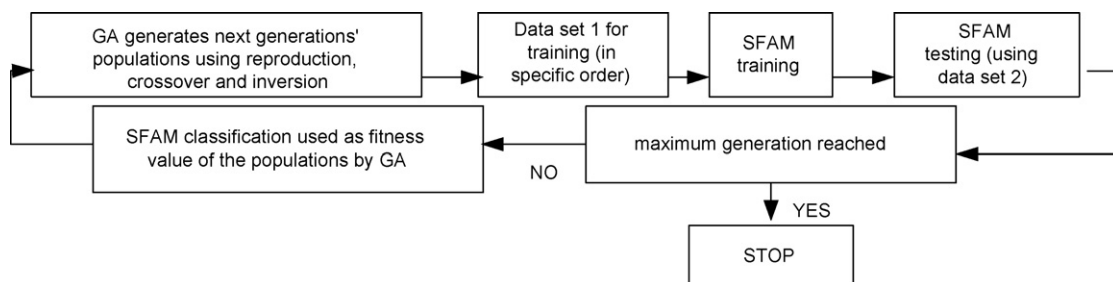


Fig. 5. GA method to select the presentation order.

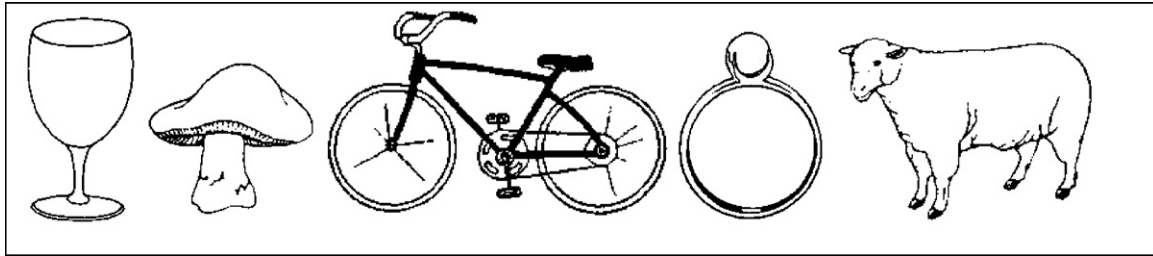


Fig. 6. Examples of pictures shown.

1 m away from the subjects. Forty VEP measurements (each with length 1 s after each stimulus onset) were stored.

Gamma band energy in the range of 30–50 Hz for each VEP signal was computed using zero-phase forward and reverse Butterworth filter and Parseval's time-frequency equivalence theorem. Each VEP pattern consisted of these energy features from 61 channels. These VEP patterns were then classified into the 40 categories representing the different subjects.

The data consisting of 1600 VEP patterns were divided exclusively into three sets: datasets 1, 2 and 3. Datasets 2 and 3 each consisted of 13 VEP patterns from each subject, while dataset 1 consisted of 14 VEP patterns from each subject. Therefore, in total, dataset 1 consisted of 560 VEP patterns, while datasets 2 and 3 consisted of 520 VEP patterns.

4.2. UCI repository data sets

4.2.1. Wine

The wine dataset is a three-class classification problem. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The original dataset consisted of 178 patterns each with 13 features representing the level of alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavanoids, non-flavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, and proline. For the purpose of the experimental study here, dataset 1 consisted of 60 patterns while datasets 2 and 3 each consisted of 59 patterns.

4.2.2. Glass

The glass dataset is a six-class classification problem, defined in terms of their oxide content (i.e. Na, Fe, K, etc.). The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence assuming that it is correctly identified. The original dataset consisted of 214 patterns each with nine features, namely refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron. The chemicals were measured in weight percent in corresponding oxide. The six classes are building windows float processed, building windows non-float processed, vehicle windows float processed, containers, tableware and headlamps. For the purpose of the experimental study here, dataset 1 consisted of 78 patterns while datasets 2 and 3 each consisted of 68 patterns.

4.2.3. Iris

The iris dataset is a three-class classification problem. The original dataset contained 3 classes of 50 instances each, where each class refers to a type of iris plant: Setosa, Versicolor, and Virginica. One class is linearly separable from the other two; the latter are not linearly separable from each other. The four features are (in cm): sepal length, sepal width, petal length, and petal width. For the purpose of the experimental study here, datasets 1, 2 and 3 each consisted of 50 patterns.

The selections of patterns for each dataset were done randomly. As could be seen from the described division, each dataset consists of approximately equal number of patterns from each class. The left over patterns (if any) after dividing the datasets into three equal portions were pushed into dataset 1.

For all the data in this experimental study, datasets 1 and 2 were used by the GA to select the presentation order of training patterns for the SFAM, while dataset 3 was used to test the performance of the trained SFAM.

4.3. Performance testing—results and discussion

The classification was carried out for VP values ranging from 0.1 to 0.9 (in steps of 0.1) but to save space, only the averaged results for classification performances, averaged training times (for a single pattern) and averaged SFAM network sizes (i.e. number of Fuzzy ART F_2 nodes) are given in Table 2. Note that the SFAM training times reported in Table 2 were the average time to train a single pattern in the performance testing phase. Actually, the random-voting method would require 20 as many weights or 20 training times as shown in Table 2. But in reality, either the training time or the weights would be 20 times more, but not both the training time and weights. This is a notable fact, though the average of both training time and weights are reported here. In addition, the SFAM training time for random-average method was not from 20 SFAM trainings but averaged from 20, which was done to approximate one SFAM training time. Fig. 7 shows the classification performances for varying VP values using different ordering methods for the four different datasets.

From Table 2 and Fig. 7, it can be seen that the GA method gave superior classification performance over both random ordering and the min–max methods for all the VP values. This is true for all the studied four data sets. For example, for the individual classification data, the averaged performances over all the VP values were 93.48% (GA method), 89.17% (random ordering – averaged), 91.84% (random ordering with voting) and 90.26% (min–max method). It can also be seen that GA based presentation order of training patterns required lower training times and smaller SFAM sizes when compared to the random ordering and min–max methods. This was also true for all the VP values and for all the four data sets. The GA method is also advantageous over random ordering since it requires only one simulation. Note that the classification performances of the min–max method reported here for wine, class and iris data are different from [3] because of the difference in the sizes of the data used.

Another interesting fact that can be concluded from Fig. 7 is that the VP values do not affect the classification performance of the GA method as significantly as they do for the random ordering and min–max methods. As such, if the GA method is used, the value of VP can be fixed at 0. The two main parameters that require tuning for SFAM are presentation order of training patterns and VP. By using the GA method, the SFAM does not require tuning of either of these parameters.

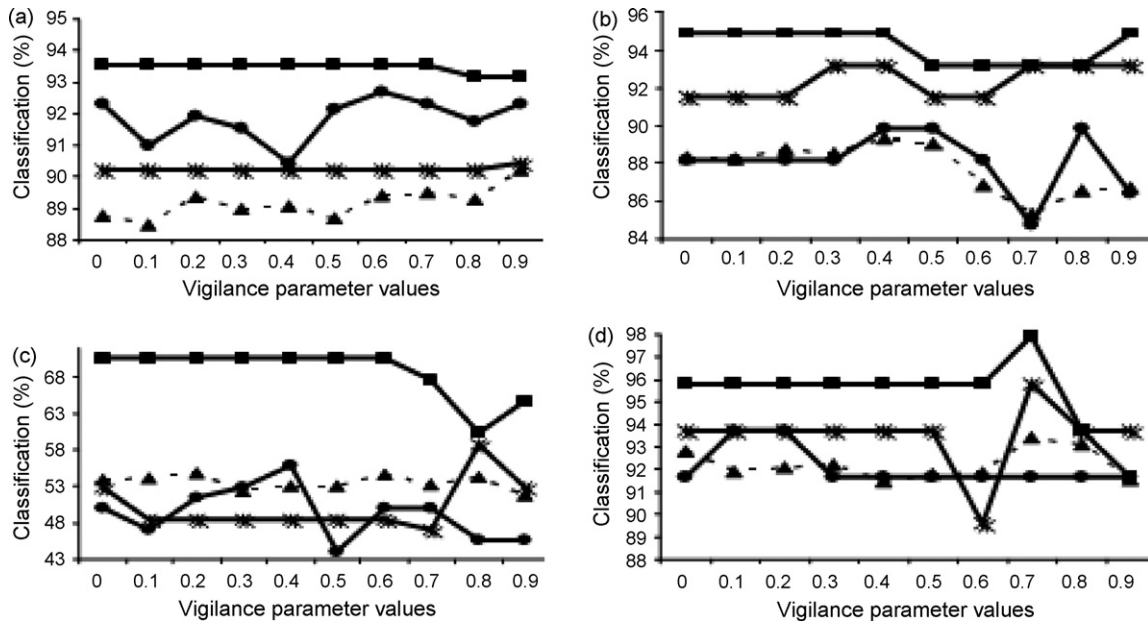


Fig. 7. Classification performances using different ordering methods for the four data sets: (a) individual; (b) wine; (c) glass; and (d) iris. Legend: GA method (—■—), random-averaged (---▲---), random-voting (---●---), and min-max (—◆—).

Table 2
Summary of averaged results (VP=0.1–0.9) for the different datasets using different ordering methods

Dataset	Ordering method	Training time (s)	SFAM size	Classification (%)
Individual	GA method	0.011134	94.10	93.48
	Random – average	0.012028	98.96	89.17
	Random – voting	0.012028	98.96	91.84
	Min–max	0.011150	94.20	90.26
Wine	GA method	0.001033	10.90	94.24
	Random – average	0.001201	11.37	87.77
	Random – voting	0.001201	11.37	88.14
	Min–max	0.001283	12.60	92.38
Glass	GA method	0.000731	18.80	68.68
	Random – average	0.001122	19.10	53.57
	Random – voting	0.001122	19.10	49.26
	Min–max	0.001397	21.30	50.29
Iris	GA method	0.000945	7.60	95.42
	Random – average	0.000949	7.87	92.22
	Random – voting	0.000949	7.87	92.08
	Min–max	0.001037	8.10	93.54

In the pattern ordering phase, our simulations indicate that GA selects the pattern order is much lesser time than min–max though the GA has to run 100 iterations! This difference becomes more evident as the numbers of training patterns become higher due to the increase of the min–max ordering complexity with increasing pattern size. However, exact time comparisons are not a matter of concern as there would be no such comparison possible with the random ordering method which does not have pattern ordering phase. Furthermore, the pattern ordering phase is generally conducted ‘offline’ and the important issues are actual performances addressed in the second phase (like training time, size, accuracy) once the presentation order of training patterns have been selected.

5. Conclusion

This paper has proposed the use of GA to select the presentation order of training patterns for SFAM. The new method could also be applied to FAM. The performances of the proposed method have

been compared with the performances of the random ordering with a voting strategy and the min–max method for solving an individual classification problem using VEP signals and three data sets from UCI repository. Though there are computational overheads for the proposed method, it will be only during the pattern ordering phase and once ordered, the method performed the fastest.

It has been shown that SFAM classification performances were better for the GA based method when compared to the random ordering and the min–max methods. Further the GA based method required lower training times and smaller SFAM sizes when compared to the other methods. An additional advantage of the proposed method over random ordering method was that it required only a single simulation. The SFAM classification performances, when training patterns were ordered by the GA method showed only a small variance for different VP values. SFAM could be used with a VP of 0 for both the pattern ordering and performance testing phases. This means that the two parameters in SFAM, namely, presentation order of training patterns and VP do not require tuning for this method.

Acknowledgment

We thank Prof. Henri Begleiter at the Neurodynamics Laboratory at the State University of New York Health Centre at Brooklyn, USA who generated the raw VEP data and Mr. Paul Conlon, of Sascos Hill Research, USA for sending us the database. A part of this work was funded by University of Essex Research Promotion Fund (DDQP40).

References

- [1] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, D.B. Rosen, Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps, *IEEE Trans. Neural Netw.* 3 (5) (1992) 698–713.
- [2] T. Kasuba, Simplified fuzzy ARTMAP, *AI Expert* 8 (11) (1993) 19–25.
- [3] I. Dagher, M. Georgiopoulos, G.L. Heileman, G. Bebis, An ordering algorithm for pattern presentation in Fuzzy ARTMAP that tends to improve generalization performance, *IEEE Trans. Neural Netw.* 10 (4) (1999) 768–778.

- [4] R. Palaniappan, P. Raveendran, S. Nishida, N. Saiwaki, A new brain-computer interface design using fuzzy ARTMAP, *IEEE Trans. Neural Syst. Rehabil. Eng.* 10 (3) (2002) 140–148.
- [5] P. Raveendran, R. Palaniappan, S. Omatu, Fuzzy ARTMAP classification of invariant features derived using angle of rotation from a neural network, *Inf. Sci.: Int. J.* 130 (2000) 67–84.
- [6] M. Vakil-Baghmisheh, N. Pavesic, A fast simplified fuzzy ARTMAP network, *Neural Process. Lett.* 17 (2003) 273–316.
- [7] R.L. Hought, S.E. Haupt, *Practical Genetic Algorithms*, John Wiley & Sons, 1998.
- [8] R. Palaniappan, New method to identify individuals using VEP signals and neural network, *IEEE Proc. Sci. Meas. Technol. J.* 151 (1) (2004) 16–20.
- [9] P. Murphy, D. Ana, UCI Repository of Machine Learning Databases, Department of Computer Science, University of California, Irvine, CA, Technical Report. Available <http://www.ics.edu/mlearn/MLRepository.html> (1994).
- [10] J.G. Snodgrass, M. Vanderwart, A standardized set of 260 pictures: norms for name agreement, image agreement, familiarity, and visual complexity, *J. Exp. Psychol.: Hum. Learn. Mem.* 6 (2) (1980) 174–215.