Pergamon

Pattern Recognition, Vol. 27, No. 8, pp. 1103–1110, 1994
Elsevier Science Ltd

0031-3203/94 $7.00 + .00

SELF-ORGANIZING NEURAL-NETWORK-BASED PATTERN CLUSTERING METHOD WITH FUZZY OUTPUTS

D. T. PHAM and E. J. BAYRO-CORROCHANO

Intelligent Systems Research Laboratory, School of Electrical, Electronic and Systems Engineering, P.O. Box 904, Cardiff CF1 3YH, U.K.

(*Received 7 January* 1993; *received for publication 23 September* 1993)

Abstract—A pattern clustering method based on the Kohonen feature mapping algorithm and the backpropagation multilayer perceptron is described. The method comprises two phases. First, the Kohonen algorithm and a simple cluster labelling procedure is applied to the training data set to divide it into labelled clusters. The data clusters are then employed to train a three-layer perceptron using the error backpropagation technique. Thus the method is self-organizing by virtue of the Kohonen algorithm and naturally produces fuzzy outputs as a consequence of the backpropagation network. The results of using the proposed method on two standard clustering problems are presented. These show that the method has superior performance compared to crisp clustering networks such as the Kohonen feature map and the ART-2 network.

Fuzzy logic   Neural computing   Self-organization   Unsupervised learning   Pattern recognition

1. INTRODUCTION

Many real pattern recognition or classification problems are fuzzy: a given pattern does not necessarily belong exclusively to one class or another but can have varying degrees of membership to several classes. It is useful to employ pattern classifiers with continuous valued outputs which indicate the extent to which a pattern is a member of different classes. These fuzzy pattern classifiers match the nature of real problems more closely and thus tend to be more reliable than ordinary crisp classifiers.

In a sizeable number of practical situations, the available classes into which patterns should be grouped are not known in advance. For those situations, there are pattern clustering methods which automatically assemble similar patterns together to form classes in a self-organized manner. However, although most self-organizing pattern classifiers can handle noisy or fuzzy input patterns and, as such, are able to generalize, they only have binary outputs, that is, they only act as crisp classifiers.

This paper presents a pattern classification method which is self-organizing and produces fuzzy outputs. The proposed method is based upon two well-known neural network models, the self-organizing feature map by Kohonen(1) and the backpropagation multilayer perceptron by Werbos,(2) Parker(13) and Rumelhart *et al.*(4) The paper is organized as follows. Section 2 briefly surveys previous work on fuzzy unsupervised and self-organizing neural classifiers. Section 3 reviews the main aspects of the Kohonen and backpropagation networks. Section 4 describes the proposed pattern classification method. Section 5 gives the results obtained for two classification problems using the proposed method, the original Kohonen network and the Adaptive Resonance Theory (ART) network, another well-known self-organizing crisp neural pattern classifier developed by Carpenter and Grossberg.(5)

2. PREVIOUS WORK

There are two main types of neural fuzzy self-organizing classification or clustering methods. They are based on two major unsupervised neural network paradigms, the above-mentioned ART network and its family and the self-organizing feature map.

Fuzzy ART(6) and the Fuzzy Min–Max neural network,(7) both developed by Simpson, are examples of fuzzy clustering networks related to ART. These networks, divide the space of input patterns into fuzzy sets. The fuzzy Min–Max network, which is an improved version of Fuzzy ART, is trained by allowing the fuzzy sets automatically to expand to cover the input space. The expansion is controlled so that the crisp cores of these sets do not overlap one another. Like the original ART network, the fuzzy versions can incorporate new input data and create additional clusters without retraining. A previously unseen input pattern is assigned to existing clusters with different degrees of membership if it is related to patterns in those clusters, in other words located sufficiently close to the cores of the clusters. Otherwise, a new cluster is generated to accommodate the given pattern. Note that there is another clustering network also named Fuzzy ART.(8) However, that network only produces crisp outputs rather than provide information regarding degrees of membership.

Fuzzy clustering networks which integrate the fuzzy-c-means clustering technique(9,10) and the Kohonen algorithm for self-organizing feature map construction have been proposed by Huntsberger and Ajjimarangsee(11) and by Bezdek *et al.*(12) Essentially these networks have an additional layer of neurons located at the output of a Kohonen network. The task of the additional neurons is to compute the cluster membership values of the input pattern as a function of the distance between that pattern and the different cluster centres. They also feedback these membership values to the Kohonen network to determine the learning rates for the latter: the larger the membership value for a cluster, the higher the learning rate for the associated neuron in the Kohonen network.

3. KOHONEN SELF-ORGANIZING FEATURE MAP AND BACKPROPAGATION NETWORK

For completeness, this section briefly reviews the main aspects of the two neural network paradigms making up the proposed self-organizing classification scheme.

3.1. *Kohonen self-organizing feature map*

A Kohonen self-organizing feature map (see Fig. 1) consists of a 1D or 2D array of nodes (or neurons). Associated with each node is a feature vector of the same dimension as the patterns to be classified. The components of a feature vector are the weights of the connections between its node on the map and the input neurons where the components of the patterns to be classified are presented. Initially these weights are given small random values. As the feature map undergoes training, they are gradually modified so that neighbouring nodes on the map have similar feature vectors, the euclidean distance being used as the measure of similarity.

The training of a feature map, described in more detail in Appendix A, consists of feeding patterns taken from a training set or directly from an on-line process to the map via its input neurons. Upon presentation of a training pattern, the node on the map with the feature vector closest to that pattern is identified. That feature vector and the vectors belonging to nodes in the neighbourhood of its associated node are modified slightly to bring them closer to the training pattern. The amount of weight change within the neighbourhood is inversely related to the distance from the identified node. A different pattern is then presented and the training procedure is repeated. The size of the neighbourhood is reduced with each training iteration. At the end of this process, the feature map is automatically organized into regions where nodes have similar feature vectors as mentioned above. Usually an additional labelling operation is then needed to identify the different regions with the natural data clusters in the training set. An unknown pattern is subsequently recognized as belonging to a particular data cluster if it activates a node in the region labelled as representing that cluster. A node is activated by an input pattern if that pattern is closer to its feature vector than to the feature vectors of other nodes. The classification decision is purely binary.

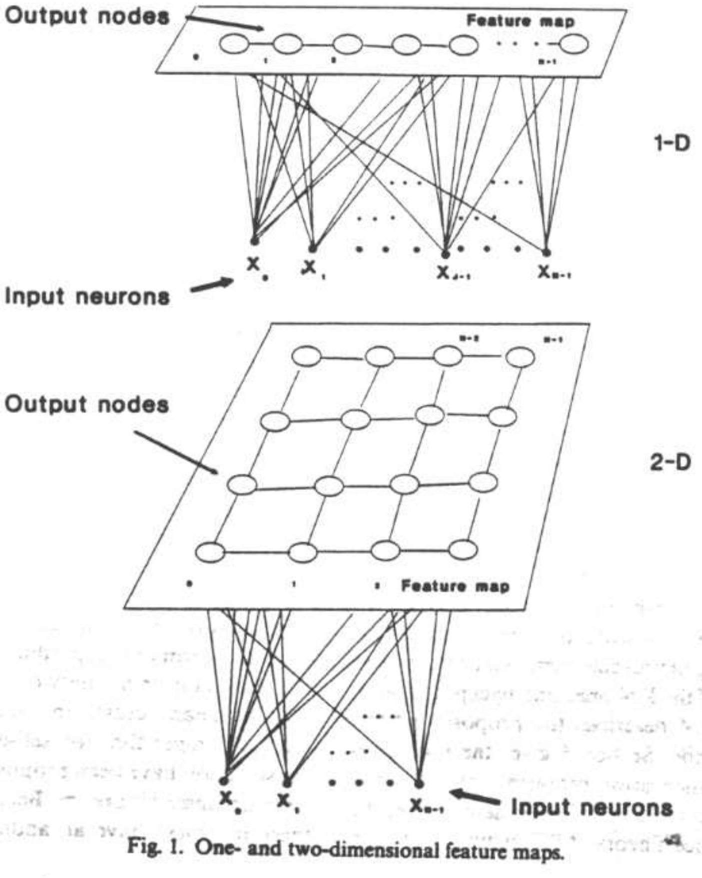

Fig. 1. One- and two-dimensional feature maps.

Each data cluster, like its corresponding feature map region, is regarded as a crisp set. Either a pattern belongs to a given cluster or it does not. Partial membership is not permitted.

3.2. *Backpropagation multilayer perceptron*

A backpropagation multilayer perceptron or backpropagation network (Fig. 2) consists of a layer of input neurons (the input layer), one or more intermediate layers of hidden neurons (the hidden layers) and a layer of output neurons (the output layer). Normally, the input neurons only act as buffers for the input data and do not perform any processing. Neurons in the hidden and output layers carry out simple operations such as summing up the signals at their inputs and perhaps passing the result through a "squashing" activation function. There are no connections between neurons in the same layer. Neurons in consecutive layers are linked together such that signals propagate in one direction from the input layer to the output layer via the hidden layers. The influence of one neuron on a neuron linked to it in the next layer is determined by the weight of the connection between them. The objective of training in a backpropagation network is to determine the values of all the weights for it to produce the correct output signals in response to the patterns in a training set. The backpropagation algorithm used in the training operation employs a type of gradient descent technique. It first obtains the difference between the actual output of the network and the target output for a training pattern. Then it propagates that difference from the output layer backwards through to the input layer to compute the amount of weight changes to reduce the difference. Thus, unlike the feature mapping algorithm, the backpropagation algorithm requires a priori knowledge of the target output for each training pattern. Furthermore, the backpropagation algorithm produces a network with the ability to interpolate and thus generate outputs indicating the degrees of membership of different clusters for a pattern that does not exactly fit into a single cluster.

4. SELF-ORGANIZING FUZZY PATTERN CLASSIFICATION METHOD

The proposed classification method is implemented in two phases. In the first phase, the Kohonen feature mapping algorithm is employed together with a simple automatic clustering procedure to divide the training data set into labelled classes. In the second phase, a backpropagation multilayer perceptron is configured and trained to recognize the known data clusters obtained in the first phase. Thus the proposed classification method has both the self-organizing characteristic of a Kohonen network and the interpolation capability of a backpropagation multilayer perceptron. Consequently, it is not necessary to know in advance the number of clusters present in the training data set nor to predetermine the detailed structure of the final neural classifier. At the same time, it is possible to classify patterns into somewhat fuzzy clusters, attributing to each pattern varying degrees of membership to the different clusters.

4.1. *Phase 1: self-organized determination of data clusters and target output patterns*

This phase comprises the following steps:

Step 1: *formation of topologic feature map.*

The Kohonen feature mapping algorithm is applied to the training data set to construct a topologic feature map and associated feature vectors for the given problem.

Step 2: *identification of clusters of nodes on feature map.*

A simple clustering procedure based on euclidean distances[1] is used to group the feature vectors obtained in Step 1, and thus their corresponding nodes on the feature map, into individually labelled clusters. The centroid of each cluster is also found.

Step 3: *identification of data clusters and determination of target output patterns.*

The labelled feature map of Step 2 is employed to subdivide the training data set into crisp clusters. Training patterns which excite nodes within a specified

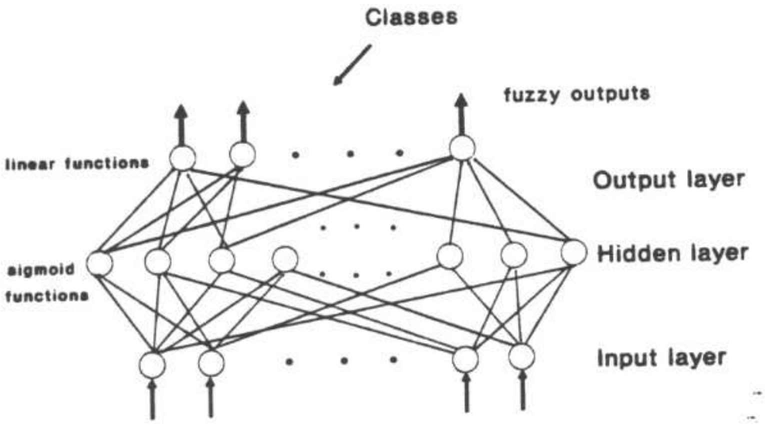

Fig. 2. Backpropagation three-layer perceptron.

distance of the centroid of a cluster are selected from each of the main data clusters for use in the second phase to train the backpropagation network. A target output pattern is defined for each training pattern depending on the data cluster to which it belongs. According to the problem, the same target output pattern could be assigned to training patterns in the same cluster or (a small number of) different target output patterns could be used for training patterns corresponding to nodes at different distances from the centroid.

4.2. *Phase II: construction of fuzzy classifier*

Step 1: configuration of the backpropagation network.

A backpropagation network consisting of three layers is formed. The number of neurons in the input layer is equal to the dimension of the input training patterns. The number of neurons in the output layer is taken as the dimension of the target output patterns, which is the same as the number of main data clusters obtained in Step 3 of Phase I. The number of neurons in the hidden layer is made equal to at least twice that of output neurons. The input and output neurons have linear activation functions and the hidden neurons, sigmoidal activation functions.

Step 2: training of backpropagation network.

The backpropagation algorithm is employed to teach the network configured in Step 1 to classify the selected training data set using the input–output pattern pairs identified in Step 3 of Phase I. Upon successful training the resulting network should reproduce the target output patterns when fed with the corresponding training patterns. As previously mentioned, due to its interpolative nature, the network will yield slightly different output patterns if the input patterns do not exactly match the training patterns. This gives it the desired fuzzy behaviour.

5. RESULTS

The proposed self-organized classification method was evaluated on two standard clustering problems. These are described in this section together with the results obtained. For comparison, the performance of the original Kohonen self-organized feature mapping algorithm and the ART-2 classifier on the same problems is also presented. As detailed below, different configurations (1D, 2D) of the Kohonen feature maps were experimented with. The ART-2 model used in the tests was similar to the architecture depicted in Fig. 10 of reference (5) except that the F0 layer was made isomorphic to the F1 layer (see also Fig. B1 in Appendix B).

5.1. *Clustering problem 1: Iris data classification*

The Iris data set, assembled by Anderson,[13] consists of measurements of the flowers of 50 plants from each of the three species: Iris Setosa, Versicolor and Virginica. Four measurements (sepal length, sepal width, petal length, petal width) were taken for each flower and normalized to have real values between 1.0 and 0.0. Thus there are three natural clusters in the data set according to the measurement statistics. The cluster corresponding to the Iris Setosa is well separated from the Iris Versicolor and Virginica clusters which overlap each other slightly.

Of the 150 data samples (4-tuples) in the data set, 100 were used to train the various classifiers and 50 were retained to test their generalization ability. The percentages of test samples accurately recognized by the different classifiers are given in Table 1. Nine classifiers were evaluated in total: 1 ART-2 classifier, 4 Kohonen feature maps (a 1D map with 5 nodes, a 1D map with 12 nodes, a 2D map with 4×4 nodes and a 2D map with 10×10 nodes) and 4 backpropagation networks all with the same structure (4 input neurons, 6 hidden neurons and 3 output neurons). The backpropagation networks were obtained using the method described in Section 4. The four Kohonen feature maps employed to create the training data clusters for these networks were the same as those used in the evaluation. For this problem, all data samples belonging to the same cluster were assigned the same target pattern. Table 2 gives examples of outputs produced by the third backpropagation network trained using the 2D Kohonen network with 4×4 nodes as detailed in Table 1.

Table 1. Results for Iris clustering problem

Neural classifier	Accuracy (%)
ART-2	92
Kohonen	
(i) (1D, 5 neurons)	80
(ii) (1D, 12 neurons)	92
(iii) (2D, 4 × 4 neurons)	92
(iv) (2D, 10 × 10 neurons)	92
Backpropagation (4/6/3)	
(i) (1D, 5 neurons)	82
(ii) (1D, 12 neurons)	96
(iii) (2D, 4 × 4 neurons)	96
(iv) (2D, 10 × 10 neurons)	96

Table 2. Sample outputs from backpropagation network (iii) for Iris clustering problem

Class	*O*(0) Setosa	*O*(1) Versicolor	*O*(2) Virginica
Setosa	0.91	0.01	0.00
Setosa	0.92	0.01	0.00
Setosa	0.92	0.02	0.00
Versicolor	0.01	0.92	0.01
Versicolor	0.01	0.88	0.03
Versicolor	0.01	0.65	0.14
Virginica	0.00	0.033	0.85
Virginica	0.00	0.029	0.87
Virginica	0.00	0.01	0.92

5.2. *Clustering problem 2: butterfly data set*

A bi-dimensional data set was created that is similar to the popular butterfly data set commonly used to test clustering algorithms, but with more data samples (51 instead of only 15 as in the original butterfly data set). Figure 3 depicts the data set and clearly shows that there is much overlapping between the two clusters forming the "wings" of the butterfly. Three classifiers were evaluated: an ART-2 classifier, a 1D Kohonen feature map with 10 nodes, and a backpropagation network with 2 input neurons, 4 hidden neurons and 2 output neurons. The backpropagation network was again configured according to the procedure detailed in Section 4. The clustering and labelling of the data used to train it was carried out with the help of the 1D Kohonen feature map. As with the Iris clustering problem, all data samples in the same cluster were assigned the same target pattern. Table 3 gives the overall classification accuracies achieved by the different classifiers (that is, the accuracies obtained for the complete data set, 40 of which had been used to train the classifiers). Table 4 presents the decisions of the classifiers regarding individual samples in the data set.

Table 3. Results for butterfly data set clustering problem

Neural classifiers	Errors in 40 training samples	Errors in 11 test samples
ART-using complement coding: $I = (\mathbf{x}, \mathbf{x}^c)$	0	5
Kohonen (1D, 10 neurons)	4	4
Backpropagation (2/4/2) (1D, 10 neurons)	0	0

6. DISCUSSION AND CONCLUSION

Table 1 shows that provided there are sufficient nodes in the Kohonen feature map used to pre-cluster

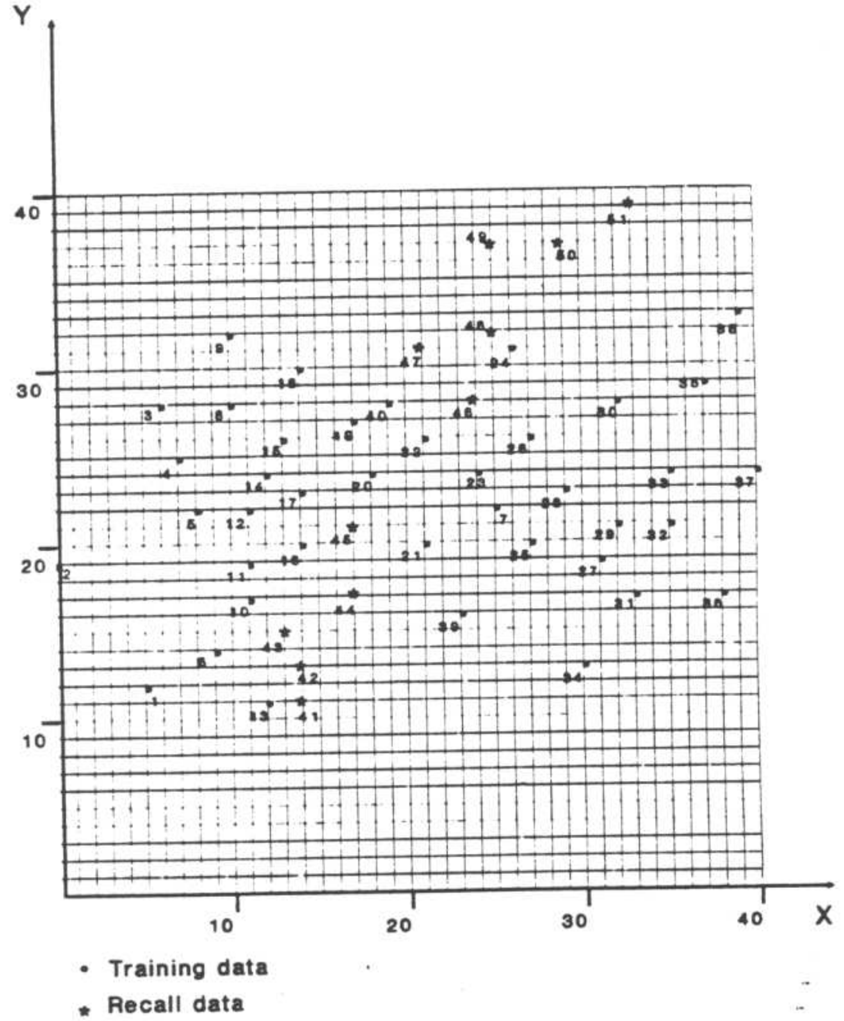

Fig. 3. Butterfly type data set.

Table 4. Outputs from neural classifiers for butterfly data set clustering problem

No.	Coordinates x	y	ART-2 $I=(\mathbf{x},\mathbf{x}^c)$	Neural classifiers Kohonen (1D, 10 neurons)	Backpropagation (1D, 10 neurons) $O(0)$	$O(1)$
1t	5	12	0	0	0.94	0.01
2t	0	19	0	0	0.94	0.01
3t	6	28	0	0	0.95	0.01
4t	7	25	0	0	0.95	0.01
5t	8	22	0	0	0.95	0.01
6t	9	14	0	0	0.94	0.01
7t	25	22	1	1	0.02	0.91
8t	10	28	0	0	0.95	0.01
9t	10	32	0	0	0.95	0.01
10t	11	17	0	0	0.94	0.01
11t	11	19	0	0	0.94	0.01
12t	11	22	0	0	0.94	0.01
13t	12	11	0	1	0.93	0.01
14t	12	24	0	0	0.94	0.01
15t	13	26	0	0	0.94	0.01
16t	14	20	0	1	0.93	0.01
17t	14	23	0	0	0.94	0.01
18t	14	30	0	0	0.94	0.01
19t	17	27	0	0	0.91	0.02
20t	18	24	0	0	0.85	0.04
21t	21	20	1	1	0.21	0.45
22t	21	26	0	1	0.39	0.28
23t	24	24	1	1	0.04	0.85
24t	26	31	1	1	0.02	0.91
25t	27	20	1	1	0.01	0.93
26t	27	26	1	1	0.02	0.93
27t	31	19	1	1	0.01	0.94
28t	29	23	1	1	0.01	0.94
29t	32	21	1	1	0.01	0.94
30t	32	28	1	1	0.01	0.95
31t	33	17	1	1	0.01	0.94
32t	35	21	1	1	0.01	0.94
33t	35	24	1	1	0.01	0.94
34t	30	13	1	1	0.01	0.94
35t	37	29	1	1	0.01	0.95
36t	38	17	1	1	0.01	0.94
37t	40	24	1	1	0.01	0.94
38t	39	33	1	1	0.01	0.95
39t	23	16	1	1	0.03	0.85
40t	19	28	0	1	0.82	0.05
41r	14	11	0	1	0.91	0.01
42r	14	13	0	1	0.92	0.01
43r	13	15	0	1	0.93	0.01
44r	17	17	?	1	0.84	0.03
45r	17	21	?	1	0.88	0.03
46r	24	28	0	1	0.05	0.82
47r	21	31	0	1	0.56	0.17
48r	25	32	0	1	0.04	0.86
49r	25	37	0	1	0.07	0.81
50r	29	37	1	1	0.02	0.94
51r	33	39	1	1	0.01	0.95

t, training data; r, recall (test) data; ?, unidentified class.

the training data set, the proposed neural classifier will have superior performance compared to the crisp Kohonen and ART networks. For the given problem there was no difference in performance between backpropagation networks built using 1D and 2D Kohonen networks. Also, there was no merit in employing large Kohonen networks. The results obtained for the case of the Kohonen network with 10×10 nodes used on its own and in tandem with a backpropagation network were not better than for smaller Kohonen networks.

Table 2 reveals the fuzzy nature of the outputs from the backpropagation network. As expected, because of the clear separation between the Setosa and Versicolor clusters the network outputs were distinct for samples of these two clusters (the same applies to the outputs corresponding to samples of the Setosa and

Virginica clusters). The overlapping of the Versicolor and Virginica clusters is reflected by less distinct outputs for certain samples of these clusters.

Table 3 further illustrates the improvements achieved with the proposed clustering technique. The crisp Kohonen network was not able to learn the training data set completely. It misclassified 4 out of 40 training samples. However, the backpropagation network, trained using a selected data set preclustered by the same Kohonen network, was able to learn to classify correctly all samples from the training data set. Moreover, it could classify 100% of the test set without error whereas there was a high proportion of misclassifications with both the ART and Kohonen networks.

The robust performance of the backpropagation network is even more evident from Table 4. For those patterns in the region between the two butterfly wings, the crisp classifiers failed in a brittle manner because they had to make binary decisions regarding the cluster to which these patterns belonged. Due to its inherent "elasticity", that is its ability to interpolate, the backpropagation network proved to be much more resilient.

In conclusion, the neural-network-based clustering technique proposed in this paper has the dual advantage of being able to organize itself and produce outputs indicating the degrees of membership of the different clusters. In two typical pattern classification problems, it has performed better than non-fuzzy pattern clustering networks, in particular, the Kohonen network on which it is partly modelled. Lack of detailed information has precluded a direct comparison with existing fuzzy pattern clustering networks. However, a clear merit of the proposed method is that it results in a simple and compact backpropagation network which can be readily implemented in hardware for high speed real time applications.

Acknowledgement—The authors would like to thank the ACME Directorate of the SERC, Seal Technology Systems Ltd, Performance Vision Ltd, the British Council and the Bolivian Science Academy for supporting this work.

REFERENCES

1. T. Kohonen, *Self-organization and Associative Memory*. Springer, Heidelberg (1984).
2. P. J. Werbos, Beyond regression: new tools for prediction and analysis in the behavioral sciences, Ph.D. Thesis, Harvard University, Cambridge, Massachusetts (1974).
3. D. B. Parker, Learning logic, Technical Report TR-47, Center for Computational Research in Economics and Management Science, MIT, April (1985).
4. D. E. Rumelhart, G. E. Hinton and R. J. Williams, Learning internal representation by error propagation, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, D. E. Rumelhart and J. L. McClelland, eds, Vol. 1, pp. 318–362. MIT Press, Cambridge, Massachusetts (1986).
5. G. A. Carpenter and S. Grossberg, ART-2: Self-organization of stable category recognition codes for analog input patterns, *Appl. Optics* **26**(23), 4919–4930 (1987).
6. P. Simpson, Fuzzy adaptive resonance theory, Southern Illinois University Neuroengineering Workshop, Carbondale, Illinois, 6–7 September (1990).
7. P. K. Simpson, Fuzzy min–max neural networks: Part 2 clustering *IEEE Trans. Fuzzy Syst.* **1**(1), 32–45 (1993).
8. G. A. Carpenter, S. Grossberg and D. Rosen, Fuzzy ART: an adaptive resonance algorithm for rapid, stable classification of analog patterns, *Proc. 1991 Int. Joint Conf. on Neural Networks*, Seattle, Vol. II, pp. 411–416 (1991).
9. J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York (1981).
10. J. Bezdek, Fuzzy models and computational neural networks, Tutorial at IEEE Conf. on Neural Networks for Ocean Engineering, Washington, DC, 15 August (1991).
11. T. Huntsberger and P. Ajjimarangsee, Parallel self-organising feature maps for unsupervised pattern recognition, *Int. J. General Syst.* **16**, 357–372 (1989).
12. J. C. Bezdek, E. C. Tsao and N. R. Pal, Fuzzy Kohonen clustering networks, *IEEE Int. Conf. on Fuzzy Systems*, San Diego, California, pp. 1035–1043, 8–12 March (1992).
13. E. Anderson, The irises of the Gaspe peninsula, *Bull. Am. Iris Soc.* **59**, 2–5 (1936).

APPENDIX A. KOHONEN'S FEATURE MAPPING ALGORITHM

Kohonen's feature mapping algorithm can be summarized as follows:

Step (0). Set $t = 0$.

Step (1). Initialize the components of the feature vectors $\mathbf{w}_j$, i.e. the weights w_{ij} of the connections between the ith input neuron ($i = 1$ to I) and the jth feature map node ($j = 1$ to N), to small random values.

Step (2). Present a new input pattern $\mathbf{x}(t)$ to the input neurons.

Step (3). Measure the Euclidean distance d_j between the input pattern $\mathbf{x}(t)$ and each of the feature vectors $\mathbf{w}_j(t)$, where

$$d_j^2 = \sum_{i=1}^{I-1} (x_i(t) - w_{ij}(t))^2.$$

Step (4). Select the output node corresponding to the smallest d_j. Call it node s. Modify the feature vectors of all nodes according to the following equation:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t) \cdot NE_{s,j}(t)(\mathbf{x}(t) - \mathbf{w}_j(t))$$

where $\alpha(t)$ is a gain factor and $NE_{s,j}(t)$ a "neighbourhood" function that usually varies inversely with the distance measured on the feature map between the jth node and node s. Both $\alpha(t)$ and $NE_{s,j}(t)$ are decaying functions of t. For example

$$\alpha(t) = K_1 e^{(-t/T_1)} \text{ and } NE_{s,j}(t) = K_{2s,j} + K_{3s,j} e^{(-t/T_2)}$$

where K_1, T_1 and T_2 are constants. K_1 determines the maximum gain and T_1 the rate of decay of the gain. $K_{2s,j}$ and $K_{3s,j}$ define the limits of the neighbourhood of node s and T_2 ($\ll T_1$) determines the rate of shrinking of that neighbourhood. The effect of the above weight adaptation method is to pull feature vectors towards a small number of clusters and thus partition the input vector space into smaller subspaces.

Step (5). Increment t by 1. If ($t < t_{max}$) then go to step (2).

APPENDIX B. ART-2

The ART-2 network used in this work is depicted in Fig. B1. Note that the F_0 layer was made isomorphic with the F_1 layer in order to decouple the input pattern $\mathbf{I}$ from the top-down influence of the attentional subsystem. The input data was formatted using the complement coding technique ($\mathbf{I} = (\mathbf{x}, \mathbf{x}^c)^{(8)}$ for the butterfly data set. For the Iris data set, only $\mathbf{x}$ was supplied to the network. The setting parameters used in both clustering problems were: $\rho = 0.8$, $\theta = 0.001$, $a = b = 5.0$, $c = 0.1$, $d = 0.9$. For the Iris data problem, 5 iterations of the whole data set were required for the network to stabilize and for the butterfly data problem, 4 iterations were needed.

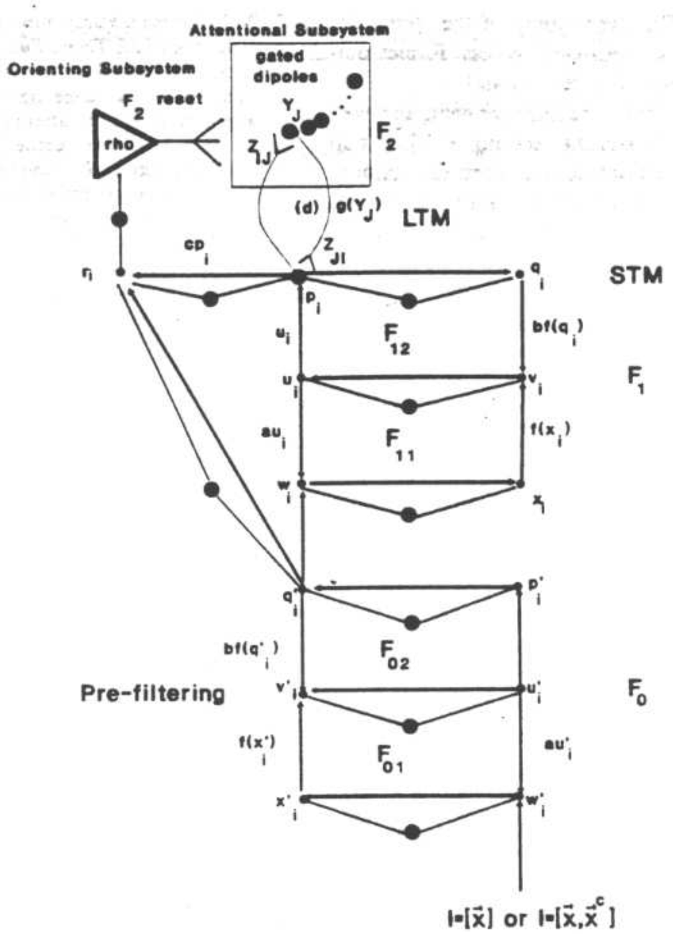

Fig. B1. Alternative ART-2 architecture.

About the Author—DUC TRUONG PHAM is Professor and Head of the Systems Engineering Division in the School of Electrical, Electronic and Systems Engineering at the University of Wales, Cardiff. He obtained his Bachelor of Engineering degree (First Class Honours) and his Ph.D. degree from the University of Canterbury, New Zealand. He worked as a lecturer in the Department of Mechanical Engineering at the University of Birmingham between 1979 and 1988, prior to his present appointment. His research interests include pattern recognition, neural networks and expert systems.

About the Author—EDUARDO BAYRO-CORROCHANO received his M.S. degree in 1978 in electronic engineering from the University Tecnica Federico Santa Maria, Valparaiso, Chile, and Ph.D. degree in 1993 in systems engineering from the University of Wales, Cardiff, U.K. From 1981 to 1982 he was a postgraduate research student at the Elektrische Nachrichten Technik Institut of the Technische Hochschule of Aachen in Germany. He was a lecturer during the periods 1979–1980 and 1983–1986 in the Electronic Engineering Department of Bolivian University in La Paz. He became the director of the Tarija Instituto de Ciencias and Tecnologia Apropiada of Bolivian University in 1987 where he remained until 1989. In parallel with his academic activities he also worked actively as a consultant engineer in areas such as applications of microprocessors, computing, telephone and telecommunication engineering and solar energy. His research interests include pattern recognition, image processing, neural networks and fuzzy logic.