AGGLOMERATIVE LEARNING FOR GENERAL FUZZY MIN-MAX NEURAL NETWORK

Bogdan Gabrys

Applied Computational Intelligence Research Unit
Department of Computing and Information Systems
The University of Paisley
High Street, Paisley PA1 2BE
Scotland, United Kingdom
Tel: +44 (0) 141 848 3752, Fax: +44 (0) 141 848 3542
E-mail: gabr-ci0@paisley.ac.uk

In this paper an agglomerative learning algorithm based on similarity measures defined for hyperbox fuzzy sets is proposed. It is presented in a context of clustering and classification problems tackled using general fuzzy min-max (GFMM) neural network. The agglomerative scheme's robust behaviour in presence of noise and outliers and its insensitivity to the order of training patterns presentation are used as a complimentary features to the previously presented incremental learning scheme more suitable for on-line adaptation and dealing with large training data sets.

I. INTRODUCTION

One of the most significant features of adaptive intelligent systems is their ability to learn. This learning is usually accomplished through an adaptive procedure, known as learning rule or algorithm, which gives a formula of updating the parameters of the system (i.e. adapting weights in artificial neural networks) in such a way as to improve some performance measure [6].

All learning algorithms to be found in the neural network and machine learning literature can be classified as supervised, unsupervised or reinforcement learning. The distinctive feature in this classification is a type and presence of a target signal associated with each input/training pattern received from the environment.

There are many different learning rules falling within each of these three categories. They in turn can be further divided into incremental (also known as sequential) or batch learning. In incremental learning the parameters are updated after every presentation of an input pattern. In the batch learning, on the other hand, the parameter updating is performed only after all training patterns have been taken into consideration. Each of these two approaches has some advantages over the other. Incremental learning is often preferred because: first, it requires less storage than batch learning which can prove very useful when large volumes of data are available; second, it can be used for 'on-line' learning in real-time adaptive systems;

third, because of its stochastic character it can potentially escape from local minima and arrive at better-quality solutions. On the other hand, solutions found using incremental learning rules depend, to a greater or lesser extent, on the order of presentation of the training patterns, which in turn implies the sensitivity of such algorithms to initialization and greater sensitivity to noise and outliers. Stability and convergence of the incremental learning rules cannot be always proven which in extreme cases can mean divergence and a complete breakdown in the algorithm.

There has been a great amount of interest in the combination of the learning capability and computational efficiency of neural networks with the fuzzy sets ability to cope with uncertain or ambiguous data [7,9,11,12,14]. An example of such a combination is a general fuzzy min-max (GFMM) neural network for clustering and classification introduced in [5].

The GFMM NN for clustering and classification constitutes a pattern recognition approach that is based on hyperbox fuzzy sets. The incremental learning proposed in [5] combines the supervised (classification) and unsupervised (clustering) learning within a single training algorithm. The training can be described as a dynamic hyperbox expansion/contraction process where hyperbox fuzzy sets are created and adjusted in the pattern space after every presentation of an individual training pattern. A general strategy adopted is that of allowing to create relatively large clusters of data in the early stages of learning and reducing (if necessary) the maximum allowable size of the clusters in subsequent learning runs in order to accurately capture complex nonlinear boundaries between different classes.

This strategy have been shown to work very well in most of the cases, however it has also been found that the resulting input-output mapping depends on the order of presentation of the training patterns and the method is sensitive to noise and outliers.

Another undesired effect resulting from the dynamic nature of the algorithm are the overlapping hyperboxes. Because hyperbox overlap causes ambiguity and creates possibility of one pattern fully belonging to two or more different classes, the overlaps have to be resolved through a contraction process. This effect occurs purely because hyperbox expansion decisions have to be made on the basis of a single (current) input pattern and quite often would not have been taken in the first place had the whole training data been available at the same time.

In this paper an agglomerative learning algorithm for the GFMM neural network is proposed.

The agglomerative algorithms are part of a larger group of hierarchical clustering algorithms which due to their philosophy of producing hierarchies of nested clusterings have been popular in a wide range of disciplines from biology, medicine and archeology to computer science and engineering [1,2,3,13]. From our point of view, the main advantages of hierarchical clustering procedures are their insensitivity to the order of data presentation and initialization, their graphical representation of clusterings in form of dendrograms which can be easily interpreted even for high dimensional data and their potential resistance to noise and outliers that will be exploited and illustrated in the later sections of this paper.

Taking into account the deficiencies observed in the previously presented incremental learning algorithm and the general strong qualities of hierarchical clustering procedures, the agglomerative learning algorithm for GFMM has been

developed which can be used as an alternative or compliment to the incremental version for an off-line training performed on a finite training data sets. The mechanisms for processing labelled and unlabelled input data introduced for the incremental version are transferred into the agglomerative scheme ensuring that the hybrid clustering/classification character of GFMM is preserved.

In contrast to the incremental version, the data clustering process using the agglomerative algorithm can be described as a bottom-up approach where one starts with very small clusters (i.e. individual data patterns) and builds larger representations of groups of original data by aggregating smaller clusters. In this sense it can also be viewed as a neural network structure optimization method where an aggregation of two clusters means decreasing the number of neurons required to encode the data.

Most agglomerative algorithms to be found in the literature are based on point representatives of clusters and similarity measures defined for points [1,13]. Some other similarity measures have been used with cluster representatives in form of hyperspheres, hyperelipsoids or hyperplanes [2,3]. In the agglomerative procedure proposed here the similarity measures defined for hyperboxes used as cluster representatives are utilised. Using one of the proposed similarity measures between hyperboxes generally results in asymmetric similarity matrices which is in contrast to symmetric matrices normally encountered in other agglomerative algorithms (with exception of [8]). The potential implications and properties of the algorithm stemming from this fact will be discussed in section III.

The remaining of this paper is organised as follows. Section II provides a general description of GFMM neural network operation and definitions of hyperbox fuzzy sets used as cluster prototypes. In section III the similarity measures for hyperbox fuzzy sets and the agglomerative learning algorithm for GFMM are presented. Section IV summarises the performance and properties of the agglomerative learning for some toy and real data sets used in pattern recognition problems. Finally, conclusions are presented in the last section.

II. GFMM DESCRIPTION

GFMM neural network for clustering and classification [5] is a generalisation of and extension to the fuzzy min-max neural networks developed by Simpson [11,12]. The main changes in GFMM constitute the combination of unsupervised and supervised learning, associated with problems of data clustering and classification respectively, within a single learning algorithm and extension of the input data from a single point in n-dimensional pattern space to input patterns given as lower and upper limits for each dimension - hyperbox in n-dimensional pattern space.

The GFMM is represented by a three layer feedforward neural network shown at Fig. 1. It consists of $2*n$ input layer nodes, m second layer nodes representing hyperbox fuzzy sets and $p+1$ output layer nodes representing classes.

The basic idea of fuzzy min-max neural networks is to represent groups of input patterns using hyperbox fuzzy sets. A hyperbox fuzzy set is a combination of a hyperbox covering a part of n-dimensional pattern space and associated with it membership function. A hyperbox is completely defined by its min point and its

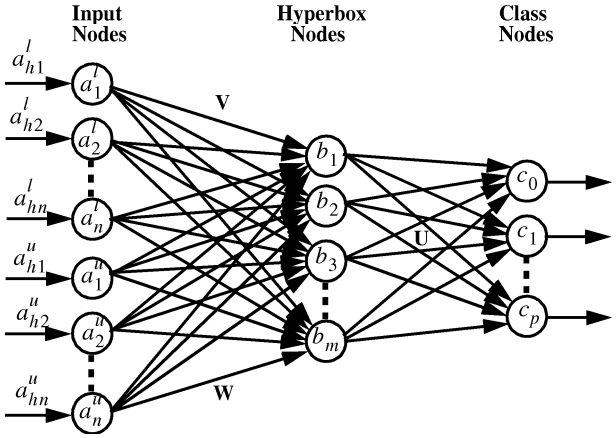

Fig. 1. GFMM neural network for clustering and classification.

max point. A membership function acts as a distance measure with input patterns having a full membership if they are fully contained within a hyperbox and the degree of membership decreasing with the increase of distance from the hyperbox. Individual hyperboxes representing the same class are aggregated to form a single fuzzy set class. Hyperboxes belonging to the same class are allowed to overlap while hyperboxes belonging to different classes are not allowed to overlap therefore avoiding the ambiguity of an input having full membership in more than one class.

The following are the definitions of input data format, hyperbox fuzzy sets, hyperbox membership function and hyperbox aggregation formula that are used within GFMM.

The input data used during the training stage of GFMM is specified as a set of N ordered pairs

$$\{\boldsymbol{A}_h, d_h\} \tag{1}$$

where $\boldsymbol{A}_h = [\boldsymbol{A}_h^l \ \boldsymbol{A}_h^u]$ is the h-th input pattern in a form of lower, $\boldsymbol{A}_h^l$, and upper, $\boldsymbol{A}_h^u$, limits vectors contained within the n-dimensional unit cube I^n; and $d_h \in \{0, 1, 2, \ldots, p\}$ is the index of one of the $p+1$ classes, where $d_h = 0$ means that the input vector is unlabelled.

The j-th hyperbox fuzzy set, B_j is defined as follows:

$$B_j = \{\boldsymbol{V}_j, \boldsymbol{W}_j, b_j(\boldsymbol{A}_h, \boldsymbol{V}_j, \boldsymbol{W}_j)\} \tag{2}$$

for all h=1,2,...,m, where $\boldsymbol{V}_j = (v_{j1}, v_{j2}, \ldots, v_{jn})$ is the min point for the j-th hyperbox, $\boldsymbol{W}_j = (w_{j1}, w_{j2}, \ldots, w_{jn})$ is the max point for the j-th hyperbox, and the membership function for the j-th hyperbox is:

$$b_j(\boldsymbol{A}_h) = \min_{i=1..n}(min([1 - f(a_{hi}^u - w_{ji}, \gamma_i)], [1 - f(v_{ji} - a_{hi}^l, \gamma_i)])) \tag{3}$$

where:

$$f(x,\gamma) = \begin{cases} 1 & if \quad x\gamma > 1 \\ x\gamma & if \quad 0 \le x\gamma \le 1 \\ 0 & if \quad x\gamma < 0 \end{cases}$$ - two parameter ramp threshold function;

$\gamma = [\gamma_1, \gamma_2, ..., \gamma_n]$ - sensitivity parameters governing how fast the membership values decrease; and $0 \le b_j(\boldsymbol{A}_h, \boldsymbol{V}_j, \boldsymbol{W}_j) \le 1$.

Hyperbox fuzzy sets from the second layer are aggregated using the aggregation formula (4) in order to generate an output which represents the degree to which the input pattern $\boldsymbol{A}_h$ fits within the class k. The transfer function for each of the third layer nodes is defined as

$$c_k = \max_{j=1}^{m} b_j u_{jk} \tag{4}$$

for each of the $p+1$ third layer nodes. Node c_0 represents all unlabelled hyperboxes from the second layer. Matrix **U** represents connections between the hyperbox and class layers of the network and the values of **U** are assigned as follows:

$$u_{jk} = \begin{cases} 1 & \text{if } b_j \text{ is a hyperbox for class } c_k \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

III. NEW LEARNING ALGORITHM

A. Similarity measures

The membership function (3) has been designed to account for maximum violation of hyperbox min and max points by an input pattern $\mathbf{A}_\mathrm{h}$. It has been dictated by the engineering application [4] where the worst possible case needed to be considered when classifying inputs in form of lower and upper limits. While (3) is interpreted as a degree of belonging of $\mathbf{A}_\mathrm{h}$ in a hyperbox fuzzy set B_j it can be easily adapted as a measure of similarity between two hyperbox fuzzy sets B_h and B_j.

For the agglomerative learning algorithm described next the following three similarity measures between two hyperboxes derived from (3) are proposed:

1) The first similarity measure between hyperboxes B_h and B_j, $s_{jh} = s(\boldsymbol{B}_j, \boldsymbol{B}_h)$ is taken directly from (3) and takes the following form:

$$s_j(\boldsymbol{B}_h) = \min_{\mathrm{i=1..n}}(min([1 - f(w_{hi} - w_{ji}, \gamma_i)], [1 - f(v_{ji} - v_{hi}, \gamma_i)])) \tag{6}$$

The characteristic features of this similarity measure are:

a) $s_{jj} = 1$

b) $0 \le s_{jh} \le 1$ - $s_{jh} = 1$ only if B_h is completely contained within B_j

c) $s_{jh} \neq s_{hj}$ - a degree of similarity of B_h to B_j is not equal to a degree of similarity of B_j to B_h (with exception when B_h and B_j are points).

The properties c) and a) lead to an asymmetrical similarity matrix used in the agglomerative algorithms with ones on its diagonal.

2) The second similarity measure between hyperboxes B_h and B_j, $\bar{s}_{jh} = \bar{s}(\boldsymbol{B}_j, \boldsymbol{B}_h)$, has been designed to find the smallest "gap" between hyperboxes and takes the following form:

$$\bar{s}_j(\boldsymbol{B}_h) = \min_{i=1..n}(min([1 - f(v_{hi} - w_{ji}, \gamma_i)], [1 - f(v_{ji} - w_{hi}, \gamma_i)])) \tag{7}$$

The characteristic features of this similarity measure are:

a) $\bar{s}_{jj} = 1$

b) $0 \leq \bar{s}_{jh} \leq 1$ - $\bar{s}_{jh} = 1$ if there is any overlap between hyperboxes B_h and B_j

c) $\bar{s}_{jh} = \bar{s}_{hj}$ - a degree of similarity of B_h to B_j is equal to a degree of similarity of B_j to B_h

The properties c) and a) lead to a symmetrical similarity matrix with ones on its diagonal.

3) The third similarity measure between hyperboxes B_h and B_j, $\hat{s}_{jh} = \hat{s}(\boldsymbol{B}_j, \boldsymbol{B}_h)$, takes into account the maximum possible distance (on every dimension basis) between hyperboxes and takes the following form:

$$\hat{s}_j(\boldsymbol{B}_h) = \min_{i=1..n}(min([1 - f(w_{hi} - v_{ji}, \gamma_i)], [1 - f(w_{ji} - v_{hi}, \gamma_i)])) \tag{8}$$

The characteristic features of this similarity measure are:

a) $0 \leq \hat{s}_{jj} \leq 1$ - $\hat{s}_{jj} = 1$ only if hyperbox B_j is a point and decreases with increasing size of B_j

b) $0 \leq \hat{s}_{jh} \leq min(\hat{s}_{jj}, \hat{s}_{hh}) \leq 1$

c) $\hat{s}_{jh} = \hat{s}_{hj}$ - a degree of similarity of B_h to B_j is equal to a degree of similarity of B_j to B_h

The properties c) and a) lead to a symmetrical similarity matrix with values less or equal one on its diagonal.

The illustration of respective similarity measures for a case of two hyperboxes in a two dimensional space is shown at Fig. 2.

The similarity measures introduced above will now be used in the agglomerative process where in each step of the procedure two most similar hyperboxes (according to one of these measures) are aggregated to form a new hyperbox fuzzy set.

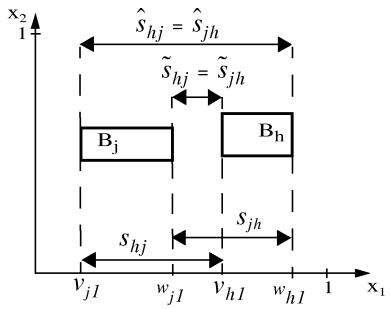

Fig. 2. Graphical illustration of hyperbox similarity measures given by (6), (7) and (8) which are proportional to the distances shown in a sense that the shorter the distance the higher the respective.hyperbox similarity value.

B. Agglomerative algorithm for GFMM

The proposed training algorithm begins with initialization of the min points matrix **V** and the max points matrix **W** to the values of the training set patterns lower $\mathbf{A}^{l}$ and upper $\mathbf{A}^{u}$ limits respectively. Labels for a set of hyperboxes generated (initialised) in this way are assigned using the labels given in the training data set $class(B_k) = d_k$, for all k=1,..,N. If the GFMM neural network was to be used at this stage the resulting pattern recognition would be equivalent to the nearest neighbour method with the distance measure represented by (3).

In the next step a similarity matrix **S** is calculated using one of the similarity measures defined above. This similarity matrix is asymmetrical for similarity measure (6) and symmetrical for similarity measures (7) and (8). This fact has implications on how to decide whether a pair of hyperboxes are to be aggregated.

For the symmetrical similarity matrix among all possible pairs of hyperboxes (B_k, B_l) the pair (B_h, B_j) with maximum similarity value s_{jh} is sought. This can be expressed as:

$$\forall_{\substack{k = 1\ldots m-1 \\ l = k+1\ldots m}} \tilde{s}_{jh} = max\ (\tilde{s}_{kl}) \tag{9}$$

or

$$\forall_{\substack{k = 1\ldots m-1 \\ l = k+1\ldots m}} \hat{s}_{jh} = max\ (\hat{s}_{kl}) \tag{10}$$

for **S** derived from (7) and (8) respectively.

For the asymmetrical similarity matrix derived using (6) the selection of a pair of hyperboxes (B_h, B_j) to be aggregated is made by finding the maximum value from a) the minimum similarity values $min(s_{kl}, s_{lk})$; or b) the maximum similarity values

$max(s_{kl}, s_{lk})$ among all possible pairs of hyperboxes (B_k ,B_l). This can be expressed as:

$$\forall_{\substack{k=1\ldots m-1 \\ l=k+1\ldots m}} s_{jh} = max\ (min(s_{kl}, s_{lk})) \tag{11}$$

or

$$\forall_{\substack{k=1\ldots m-1 \\ l=k+1\ldots m}} s_{jh} = max\ (max(s_{kl}, s_{lk})) \tag{12}$$

Once the B_h and B_j have been selected for aggregation before the aggregate B_h and B_j operation is carried out check if the following tests are passed:

a) the overlap test (see [5] for details of hyperbox overlap test)

After temporarily aggregating hyperboxes B_h and B_j check if the newly formed hyperbox does not overlap with any of the hyperboxes representing different classes. If it does take another pair of hyperboxes for potential aggregation.

b) a test for the maximum allowable hyperbox size (Θ):

$$\forall_{i=1\ldots n} (max(w_{ji}, w_{hi}) - min(v_{ji}, v_{hi})) \leq \Theta \tag{13}$$

c) a test for the class compatibility

$$\begin{array}{l} if\quad class(B_h) = 0\quad then\quad aggregate B_h \text{ and } B_j \\ else \\ \qquad if\ \ class(B_j) = \begin{cases} 0 \Rightarrow \begin{array}{c} aggregate\ B_h \text{ and } B_j \\ class(B_j) = class(B_h) \end{array} \\ class(B_h) \Rightarrow aggregate\ B_h \text{ and } B_j \\ else \Rightarrow \begin{array}{c} take \\ another\ pair\ of\ hyperboxes \end{array} \end{cases} \end{array} \tag{14}$$

If the above conditions are met the aggregation is carried out in the following way:

a) update B_j so that a new B_j will represent aggregated hyperboxes B_h and B_j

$v_{ji}^{new} = min(v_{ji}^{old}, v_{hi}^{old})$ for each i=1,..,n

$w_{ji}^{new} = max(w_{ji}^{old}, w_{hi}^{old})$ for each i=1,..,n

b) remove B_h from a current set of hyperbox fuzzy sets (in terms of neural network shown at Fig. 1 it would mean a removal of *h*-th second layer node).

c) update the similarity matrix **S** by removing *h*-th row and column and updating entries in the *j*-th row and column representing newly aggregated hyperboxes using v_{ji}^{new} and w_{ji}^{new}.

The above described process is repeated until there are no more hyperboxes that can be aggregated.

IV. SIMULATION RESULTS

The simulation experiments covering pure clustering, pure classification and hybrid clustering classification problems, have been carried out for a number of real data sets taken from the repository of machine learning databases but due to space limitation only a summary of the results is presented here.

First a potential resistance of an agglomerative scheme to noise and outliers have been tested. The two dimensional example used here consisted of two relatively dense clusters each consisting of 50 data patterns and additional 50 data patterns uniformly distributed in the input space representing noise. The data and clusters formed using (9) and (12) are shown at Fig. 3. The fact that the clusters are formed in the densest areas first with additional information about clusters cardinality has been used for filtering out outliers and the noisy data. It has been found that formulas (9) and (12) are particularly useful in recovering elongated clusters (similarly to conventional single link algorithm) and formulas (10) and (11) are particularly useful for recovering compact clusters (similarly to conventional complete link algorithm).

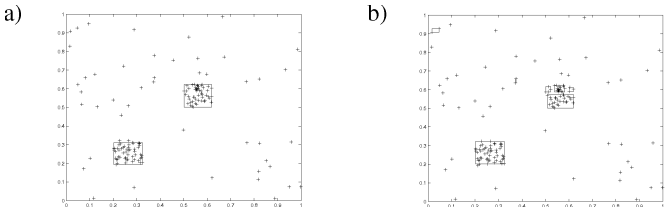

Fig. 3. Illustration of a robust behaviour of an agglomerative algorithm for noisy data. Clustering obtained using formulas a) (9) and b) (12).

In comparison to results for IRIS, Wine and Ionosphere data sets obtained for incremental learning presented in [5] the agglomerative algorithm resulted in a very similar recognition performance and generally with fewer hyperboxes required to encode the data. It is due to the above mentioned feature of creating clusters (hyperboxes) in the densest areas first and ability to discard clusters representing small number of input data.

Another interesting feature observed, especially for classifiers obtained using (11) and (12), was that although the recognition rates (number of misclassified patterns) for both were similar they were generally making errors for different testing input patterns. This is especially valuable characteristic known as a classifiers diversity when trying to construct a multiple classifier system [10].

A definite drawback of the agglomerative method is that for large data sets it can be very slow due to the size of similarity matrix containing similarity values for all pairs of hyperboxes. This, however, can be overcome to a certain extent by using the incremental learning in the initial stages of GFMM training with a relatively small value of parameter Θ. The conducted experiments showed that a significant training time reduction could be obtained with increasing Θ, however, the vulnerability of including noise and outliers in the clusters formed at this stage was also increased.

V. SUMMARY AND CONCLUSIONS

An agglomerative learning algorithm utilising hyperbox fuzzy sets as cluster representatives has been presented. New similarity measures defined for hyperbox fuzzy sets have been introduced and used in the agglomerative scheme preserving the hybrid clustering/classification character of GFMM. A robust behaviour in presence of noise and outliers and insensitivity to training data presentation have been identified as main and the most valuable complimentary features to the previously proposed incremental learning.

Using different similarity measures in a classifiers training process showed a potential for producing divers classifiers, a feature that is currently under investigation in an attempt to produce an effective multiple classifier system.

REFERENCES

[1]J.Boberg and T.Salakoski, "General Formulation and Evaluation of Agglomerative Clustering Methods with Metric and Non-metric Distances", *Pattern Recognition*, vol. 26, no. 9, pp. 1395-1406, 1993

[2]H.Frigui and R.Krishnapuram, "Clustering by Competitive Agglomeration", *Pattern Recognition,* vol. 30, no. 7, pp. 1109-1119, 1997

[3]H.Frigui and R.Krishnapuram, "A Robust Competitive Clustering Algorithm with Applications in Computer Vision", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 21, no. 5, pp. 450-465, May 1999

[4]B.Gabrys and A.Bargiela, "Neural Networks Based Decision Support in Presence of Uncertainties", *J. of Water Resources Planning and Management*, vol. 125, no. 5, pp.272-280, September/October 1999

[5]B.Gabrys and A.Bargiela, "General Fuzzy Min-Max Neural Network for Clustering and Classification", *to appear in IEEE Trans. on Neural Networks (available from http://www.paisley.ac.uk/staff/gabr-ci0/publications.html),* 2000

[6]M.H.Hassoun, *Fundamentals of artificial neural networks*, The MIT Press, 1995

[7]S.Mitra and K.Pal, "Self-Organizing Neural Network As a Fuzzy Classifier", *IEEE Trans. on Systems, Man and Cybernetics,* vol. 24, no. 3, March 1994

[8]K.Ozawa, "Classic: A Hierarchical Clustering Algorithm Based on Asymetric Similarities", *Pattern Recognition,* vol. 16, no.2, pp. 201-211, 1983

[9]W.Pedrycz, "Fuzzy Neural Networks with Reference Neurons as Pattern Classifiers", *IEEE Trans. on Neural Networks,* vol.3, no.5, September 1992

[10]D.Ruta and B.Gabrys, "An Overview of Classifier Fusion Methods", *Computing and Information Systems*, (Ed. Prof. M.Crowe), University of Paisley, ISSN 1352-9404, vol. 7, no.1, pp. 1-10, February 2000

[11]P.K.Simpson, "Fuzzy Min-Max Neural Networks - Part 1: Classification", *IEEE Trans on Neural Networks*, vol. 3, no. 5, pp.776-86, September 1992

[12]P.K.Simpson, "Fuzzy Min-Max Neural Networks - Part 2: Clustering", *IEEE Trans on Fuzzy Systems*, vol. 1, no. 1, pp.32-45, February 1993

[13]S.Theodoridis and K.Koutroumbas, *Pattern Recognition,* Academic Press, 1999

[14]R.R.Yager and L.A.Zadeh (editors), *Fuzzy sets, neural networks, and soft computing*, Van Nostrand Reinhold, 1994