ELSEVIER

Fuzzy Sets and Systems 70 (1995) 1–13

FUZZY sets and systems

Improving classification performance using fuzzy MLP and two-level selective partitioning of the feature space

Sushmita Mitra[a,*], Ludmila I. Kuncheva[b]

[a] *Machine Intelligence Unit, Indian Statistical Institute, 203, B.T. Road, Calcutta 700035, India*

[b] *Central Laboratory of Biomedical Engineering, Bulgarian Academy of Sciences, Acad. G. Bonchev Street, Block 105, 1113 Sofia, Bulgaria*

Received March 1994; revised May 1994

Abstract

A fuzzy MLP model, developed by one of the authors, is used for obtaining selective two-level partitioning of the feature space in order to improve its classification performance. The model can handle uncertainty and/or impreciseness in the input as well as the output. The input to the network is modelled in terms of linguistic pi-sets whose centres and radii along the feature axes in each partition are generated automatically from the distribution of the training data. The performance of the model at the end of the first stage is used as a criterion for guiding the selection of the appropriate partition to be subdivided at the second stage, in order to improve the effectiveness of the model. A comparative study of the performance of the two-level technique with other methods, viz., the conventional MLP, linear discriminant analysis and the k-nearest neighbours algorithms, is also provided to demonstrate its superiority.

Keywords: Fuzzy neural networks; Multilayer perceptron; Pattern classification; Partitioning; Fuzzy subspace

1. Introduction

Artificial neural networks [11, 17] are massively parallel interconnections of simple neurons that function as a collective system. They have been found to be proficient in solving various pattern recognition problems. Fuzzy sets [7, 20, 21], on the other hand, are capable of modelling uncertain or ambiguous data so often encountered in real life. Therefore, fuzzy neural networks [1, 8, 15] are designed to utilize a synthesis of the computational power of the neural networks along with the uncertainty handling capabilities of fuzzy logic. The multilayer perceptron (MLP) [17] is a feed-forward neural network model consisting of multiple layers of simple, sigmoid processing elements or neurons. A fuzzy version of the MLP (developed by one of the authors [14]) is used in this work.

A recent and potentially fruitful idea in pattern recognition, that has been directly announced or implied in several papers, is the partitioning of the initial feature space into regions and the application of different

* Corresponding author.

SSDI 0165-0114(94)00244-4

classification rules to them [3, 4, 5, 9, 12, 16]. The problem that remains is how to perform the partitioning in order to preserve at least the same classification accuracy, and hopefully achieve a better one. This classification strategy bears an analogy with the diagnostic process in medicine. If the physician does not feel competent to resolve a special case, he summons a consultation team of professionals in that particular field.

The main question that arises here concerns the way of partitioning the feature space. In fact, every rule-based classifier performs a partitioning through antecedent clauses and assigns a classification rule to each region through the implication. A partition may be based on the geometric properties of the classes detected by a preliminary clustering [3] or by a sequential groping about for the class boundaries [12]. In the fuzzy classification rule described in [4, 5] the partitioning is uniform, i.e., the regions continue to be split until a sufficiently high certainty of the rule, generated by each region, is achieved. In this work we employ a selective two-level partitioning scheme in conjunction with a fuzzy MLP network to establish the effectiveness of this notion of improving the performance in a pattern classification problem.

The fuzzy MLP model [13, 14] has already been used with data consisting of fuzzy as well as linearly nonseparable, nonconvex and disjoint pattern classes. Here we demonstrate the enhanced classification performance of the network (as compared to the status before the onset of the selective partitioning, measured by the recognition score on the training and test sets) by incorporating a two-level selective partitioning of the input space. In the first phase, the input vector (which can be in quantitative/linguistic/set forms) is represented in terms of the linguistic properties low, medium and high while the output decision is in terms of class membership values. The centres and radii of the pi-functions along each feature axis are determined automatically from the distribution of the training patterns. In the second stage, the feature space is further partitioned selectively, in order to improve the performance of the classifier. The performance index of the classifier in the first stage is used to guide the selection of the partition that has to be further subdivided for this purpose. The generation of the input description of the patterns in terms of overlapping pi-functions, corresponding to each second-level partition along the different feature axes, is also automated using the training data.

The potential ability of the model to achieve higher classification accuracy is demonstrated on two sets of synthetic data and one set of medical data on hepatobiliary disorders. A comparative study is made with the classificatory performance of the fuzzy neural network [13, 14] at the end of the first stage (i.e., before the onset of the second-level partitioning) and the more conventional approaches, viz., the standard MLP, linear discriminant analysis and the k-nearest neighbours algorithms.

2. The fuzzy MLP model

In this section we describe the fuzzy MLP model [13, 14]. Consider the layered network given in Fig. 1. The output of a neuron in any layer other than the input layer is given as

$$y_j^{h+1} = \frac{1}{1 + \exp(-\sum_i y_i^h w_{ji}^h)}, \tag{1}$$

where y_i^h is the state of the ith neuron in the preceding hth layer and w_{ji}^h is the weight of the connection from the ith neuron in layer h to the jth neuron in layer $h+1$. For nodes in the input layer, y_j^0 corresponds to the jth component of the input vector. The mean square error in output vectors is minimized by the backpropagation algorithm using a gradient descent with a gradual decrease of the gain factor.

2.1. Input vector

An n-dimensional pattern $\boldsymbol{F}_i = [F_{i1}, F_{i2}, \ldots, F_{in}]$ is represented as a $3n$-dimensional vector

$$\boldsymbol{F}_i = [\mu_{low(F_{i1})}(\boldsymbol{F}_i), \mu_{medium(F_{i1})}(\boldsymbol{F}_i), \mu_{high(F_{i1})}(\boldsymbol{F}_i), \ldots, \mu_{high(F_{in})}(\boldsymbol{F}_i)] = [y_1^0, y_2^0, \ldots, y_{3n}^0] \tag{2}$$

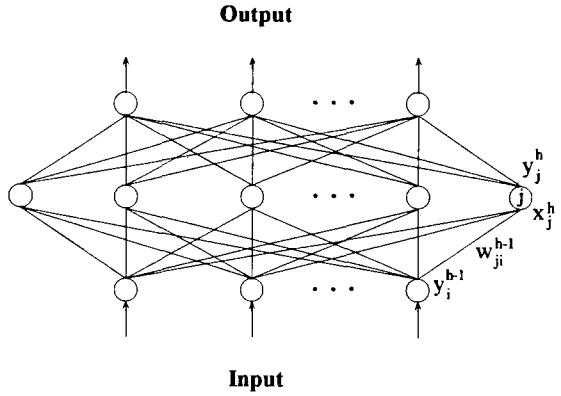

Fig. 1. The three-layered MLP model.

where the μ values indicate the membership functions of the corresponding linguistic pi-sets along each feature axis.

When the input feature is numerical, we use the π-fuzzy sets (in the one-dimensional form), with range $[0,1]$, represented as

$$\pi(F_j;c,\lambda)=\begin{cases}2\left(1-\dfrac{\|F_j-c\|}{\lambda}\right)^2 & \text{for } \dfrac{\lambda}{2}\leqslant\|F_j-c\|\leqslant\lambda,\\[2ex] 1-2\left(\dfrac{\|F_j-c\|}{\lambda}\right)^2 & \text{for } 0\leqslant\|F_j-c\|\leqslant\dfrac{\lambda}{2},\\[2ex] 0 & \text{otherwise},\end{cases} \tag{3}$$

where $\lambda>0$ is the radius of the π-function with c as the central point.

When the input feature is linguistic, its membership values for the π-sets low, medium and high are quantified as

$$low\equiv\left\{\frac{0.95}{L},\ \frac{\pi\left(F_j\left(\frac{0.95}{L}\right);c_m,\lambda_m\right)}{M},\ \frac{\pi\left(F_j\left(\frac{0.95}{L}\right);c_h,\lambda_h\right)}{H}\right\},$$

$$medium\equiv\left\{\frac{\pi\left(F_j\left(\frac{0.95}{M}\right);c_l,\lambda_l\right)}{L},\ \frac{0.95}{M},\ \frac{\pi\left(F_j\left(\frac{0.95}{M}\right);c_h,\lambda_h\right)}{H}\right\}, \tag{4}$$

$$high\equiv\left\{\frac{\pi\left(F_j\left(\frac{0.95}{H}\right);c_l,\lambda_l\right)}{L},\ \frac{\pi\left(F_j\left(\frac{0.95}{H}\right);c_m,\lambda_m\right)}{M},\ \frac{0.95}{H}\right\},$$

where $c_l, \lambda_l, c_m, \lambda_m, c_h, \lambda_h$ refer to the centres and radii of the three linguistic properties and $F_j(0.95/L)$, $F_j(0.95/M)$, $F_j(0.95/H)$ refer to the corresponding feature values F_j at which the three linguistic properties attain membership values of 0.95.

Let F_{jmax} and F_{jmin} denote the upper and lower bounds of the dynamic range of feature F_j in all L pattern points considering numerical values only. Let m_j be the mean of the pattern points along the jth axis. Then m_{jl} and m_{jh} are defined as the mean (along the jth axis) of the pattern points having co-ordinate values in the ange $[F_{jmin}, m_j)$ and $(m_j, F_{jmax}]$, respectively. For the three linguistic property sets we define the centres as

$$\begin{aligned} c_{medium(F_j)} &= m_j, \\ c_{low(F_j)} &= m_{jl}, \\ c_{high(F_j)} &= m_{jh}, \end{aligned} \tag{5}$$

and the corresponding radii as

$$\begin{aligned} \lambda_{low(F_j)} &= 2(c_{medium(F_j)} - c_{low(F_j)}), \\ \lambda_{high(F_j)} &= 2(c_{high(F_j)} - c_{medium(F_j)}), \\ \lambda_{medium(F_j)} &= fnos \frac{\lambda_{low(F_j)}(F_{jmax} - c_{medium(F_j)}) + \lambda_{high(F_j)}(c_{medium(F_j)} - F_{jmin})}{F_{jmax} - F_{jmin}}, \end{aligned} \tag{6}$$

where *fnos* is a multiplicative parameter controlling the extent of the overlapping. Here we take into account the distribution of the pattern points along each feature axis while choosing the corresponding centres and radii of the linguistic properties. This has been found to be more efficient in modelling skewed data distributions [13]. Besides, the amount of overlap between the three linguistic properties can be different along the different axes, depending on the pattern set. We are also able to ensure that any feature value along the jth axis for pattern $\boldsymbol{F}_i$ is assigned membership value combinations in the corresponding 3-dimensional linguistic space of (2) in such a way that at least one of $\mu_{low(F_{ij})}(\boldsymbol{F}_i)$, $\mu_{medium(F_{ij})}(\boldsymbol{F}_i)$ or $\mu_{high(F_{ij})}(\boldsymbol{F}_i)$ is greater than 0.5 in the interval $[c_{low} - \lambda_{low}/2, c_{high} + \lambda_{high}/2]$. Note that this range corresponds to that region of the feature axis which contains the majority of the pattern points and thereby represents the relevant region of the feature space *sans* outliers. This is because the centres and radii of the three pi-functions, used to represent the input to the neural network, are chosen automatically from the distribution of the training patterns. It also enables us to minimize the effect of those regions of the feature space that are empty. This allows most pattern vectors $\boldsymbol{F}_i$ to have strong membership to at least one of the properties low, medium and high.

2.2. Output representation

Consider an l-class problem domain such that we have l nodes in the output layer. Let the n-dimensional vectors $\boldsymbol{O}_k$ and $\boldsymbol{V}_k$ denote the mean and standard deviation, respectively, of the numerical training data for the kth class. The weighted distance of the training pattern $\boldsymbol{F}_i$ from the kth class is defined as

$$z_{ik} = \sqrt{\sum_{j=1}^{n} \left[\frac{F_{ij} - o_{kj}}{v_{kj}} \right]^2} \quad \text{for } k = 1, \ldots, l, \tag{7}$$

where F_{ij} is the value of the jth component of the ith pattern point.

The membership of the ith pattern in class k, lying in the range $[0, 1]$, is defined as

$$\mu_k(\boldsymbol{F}_i) = \frac{1}{1 + (z_{ik}/f_d)^{f_e}}, \tag{8}$$

where z_{ik} is the weighted distance from (7) and the positive constants f_d and f_e are the denominational and exponential fuzzy generators controlling the amount of fuzziness in this class-membership set.

Then, for the ith input pattern, the desired output of the jth output node is defined as

$$d_j = \mu_j(\boldsymbol{F}_i). \tag{9}$$

According to this definition a pattern can simultaneously belong to more than one class, and this is determined basically from the training set used during the learning phase. However, it may be noted that in the crisp case a pattern can either belong or not belong to a class. Then we have $z_{ik} \in \{0, \infty\}$ in (7), such that the output membership value of (8) reduces to $\mu_k(\boldsymbol{F}_i) \in \{1, 0\}$.

The learning rate of the algorithm is gradually decreased in discrete steps, taking values from the chosen set $\{2, 1, 0.5, 0.3, 0.1, 0.05, 0.01, 0.005, 0.001\}$, depending on the value of the mean square error, while the momentum factor is also decreased (generally from 0.9 to 0.5) [14]. The algorithm terminates when the learning rate of value 0.001 is reached.

3. Selective second-level partitioning of the feature space

An efficient partitioning of the feature space refers to generating neither too many nor too few partitions along the different feature axes. The distribution of the patterns in the input space is likely to play an important role in this selection. Besides, some regions of the input space may require finer partitioning than others. We use the fuzzy MLP model to determine an effective two-level partitioning, using linguistic pi-functions.

An n-dimensional pattern space is initially divided into 3^n overlapping partitions of different sizes, depending upon the centres and radii of the linguistic pi-functions determined automatically from the training set distribution using (2)–(6). The upper and lower bounds for partition x, corresponding to linguistic property p, along axis j are defined as $c_{xp_j} + \lambda_{xp_j}/2$ and $c_{xp_j} - \lambda_{xp_j}/2$ respectively. Here c_{xp_j} and λ_{xp_j} refer to the centre and radius of the π-function defining the linguistic property p for partition x along the jth axis. Next, the classification performance of the fuzzy MLP corresponding to the recognition score, with respect to each of these partitions, is evaluated. The fuzzy subspace providing the largest number of misclassifications is selected for further subdivision into 3^n overlapping regions defined by the pi-functions of (5)–(6). This is designated as the *doubtful* region, while the remaining part of the feature space is termed the *more certain* region. In this manner we can generate a total of $s \cdot 3^n - s + 1$ subspaces at the end of the sth stage. Note that we stop the process at the second level ($s = 2$), as each stage of partitioning corresponds to a related increase in the number of neurons at the input layer of the fuzzy MLP. This also helps in avoiding problems of overlearning and resultant poor generalization ability of the network. Note that we have to compromise between the classification performance of the neural network and the associated overhead due to the increase in number of input neurons with consecutive stages of partitioning of the feature space. Nevertheless, it is observed from the results of Section 4, that this selective second-level partitioning scheme serves to enhance the performance of the model to a considerable and satisfactory extent.

Let the xth subspace be selected for further division at the end of the first stage. Note that the algorithm terminates at the end of the first stage according to the criterion of decreasing learning rate, as described earlier. We determine the appropriate linguistic properties p_j corresponding to this subspace x along each feature axis j. Let m_{xp_j} be the mean of the pattern points in this subspace along the jth axis. For subdivision into three partitions along this axis, we have from (5)

$$\begin{aligned} c_{xp_{medium(F_j)}} &= m_{xp_j}, \\ c_{xp_{low(F_j)}} &= m_{xp_{j_1}}, \\ c_{xp_{high(F_j)}} &= m_{xp_{j_h}}, \end{aligned} \tag{10}$$

where $m_{xp_{j_l}}$ and $m_{xp_{j_h}}$ are the mean of all the pattern points that lie in the range $[c_{xp_j} - \lambda_{xp_j}/2, m_{xp_j})$ and $(m_{xp_j}, c_{xp_j} + \lambda_{xp_j}/2]$, respectively, of partition x (for linguistic property p). The corresponding radii along the three new linguistic property sets, along this axis, are defined analogous to (6). Note that once again the distribution of pattern points in the subspace is considered during the automatic evaluation of the required centres and radii of the three new pi-sets along each feature axis.

The new enhanced set of input features is now submitted at the input layer of the fuzzy neural model and the network trained on the pattern set under consideration. The classification performance of the model is evaluated with respect to both the training and test sets.

The idea to perform separate classification rules or strategies in different regions of the feature space have also been investigated from a different perspective by Kuncheva [9] and Rastrigin and Erenstein [16]. Ishibuchi et al. [4, 6] used an idea of sequential partitioning of the feature space into fuzzy subspaces, until a pre-determined stopping criterion was satisfied, for solving pattern classification problems. We employ a related but different idea in our fuzzy neural net-based model. Here the distribution of the training patterns is used to automate the generation of the centres and radii of the linguistic pi-functions determining the nature of the overlapping subspaces. As a result, the corresponding membership functions can be automatically tuned by the input data. Only a two-level scheme is used to compensate for the overhead involved by the considerable increase in the number of input neurons with every stage of consecutive partitioning. This also helps avoid the problem of overlearning by the neural network.

Takagi et al. [18] reported the development of a neural network architecture based on the structure of the fuzzy inference rules involved. The identification error can be analysed to improve the performance of the structured network. For this, the appropriate region of the feature space is further clustered and the corresponding *Then* parts accordingly added. We, on the other hand, use the second-level partitioning of the input feature space of the fuzzy MLP. Note that this corresponds to augmentation of the *If* parts of the relevant rules for the required pattern classification problem.

4. Implementation and results

We used two measures of percent correct classification performance for the training set. The output, after a number of updating steps, was considered a perfect match p if the value of each output neuron y_j^H was within a margin of 0.1 from the desired membership value d_j. This was a stricter criterion than the best match b_1 where we tested whether the jth neuron output y_j^H had the maximum activation when the jth component d_j of the desired output vector also had the highest value. The factor b_2 corresponded to the performance of the model when one also considered the second best choice (i.e., the output neuron with second highest activation corresponded to the correct pattern class). Note that p, b_1, b_2 refer to the training set while t_1 (best choice), t_2 (with second best choice) are indicative of the test set. The individual classwise performance (with best choice) are also provided for the test set patterns for the output classes. The second best choices b_2 and t_2 are depicted for the data on *Hepatobiliary Disorders* only (as the synthetic data consist of two classes).

Initially two artificial data sets have been used, each containing 200 cases, as the training sets. Two equiprobable classes were considered. Both data sets used the same discrimination boundary but the first one (*Random*) uses a pseudo-uniform distribution, while the second (*Cluster*) contained three pseudo-Gaussian clusters. These are illustrated in Figs. 2 and 3, respectively, with dots and triangles indicating the two classes. Two numerical features F_1 and F_2 are involved, so that the pattern points lie in the region $[0, 1] \times [0, 1]$ and may be easily visualized. The decision boundary is given as

$$f = -0.25 \sin(7\pi x_1^3) + x_2 - 0.5 .$$

Two separate data sets, each consisting of 1000 pattern points, were used as the test sets for the above two cases. In each case the corresponding training and test sets were taken from the respective distributions. The

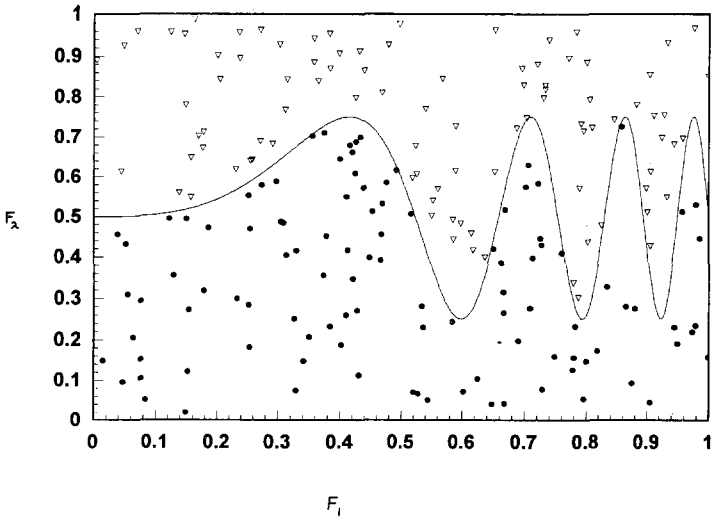

Fig. 2. The *Random* data set.

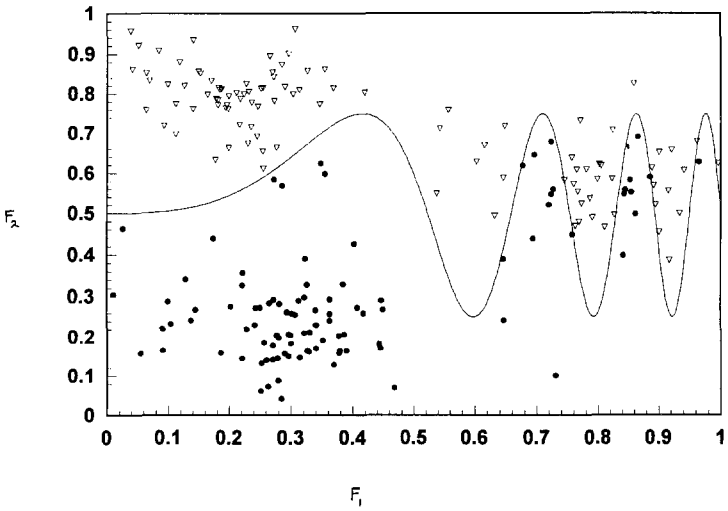

Fig. 3. The *Cluster* data set.

separate test sets were selected in order to avoid an eventual optimistic bias in assessing the classification accuracy. For a comparison, linear discriminant analysis and k-nearest neighbours algorithm were applied on the training sets and the classification accuracy assessed both for the training as well as test sets. Note that the output class membership values by (8) were crisp in this case, belonging to the set $\{1,0\}$.

The generated membership functions along the two feature axes F_1 and F_2, both at the end of the first and second stages (i.e., before and after splitting), are depicted in Figs. 4 and 5 for the *Random* and the *Cluster* data sets, respectively. Note that we use the linguistic π-sets *low*, *medium* and *high* defined by (3)–(6) and (10) in the process. The subspace *high*, *medium*, corresponding to (F_1, F_2), is partitioned in both cases.

Table 1 compares the recognition scores, with the two synthetic data sets *Random* and *Cluster*, using the fuzzy MLP both at the end of the first and second stages; those of the crisp and fuzzy k-nearest neighbours

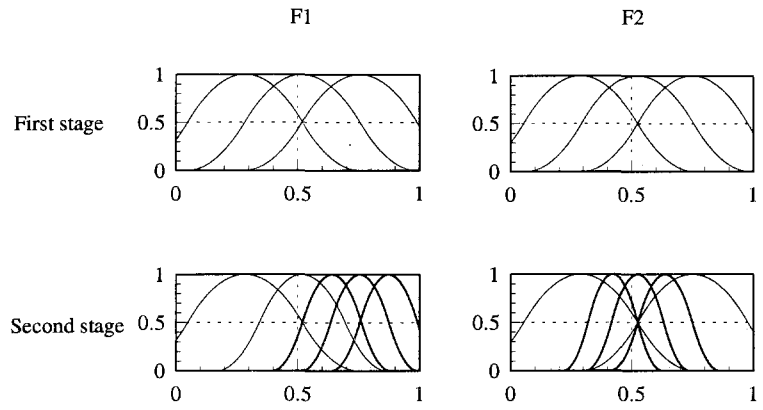

Fig. 4. Generated membership functions for *Random* data.

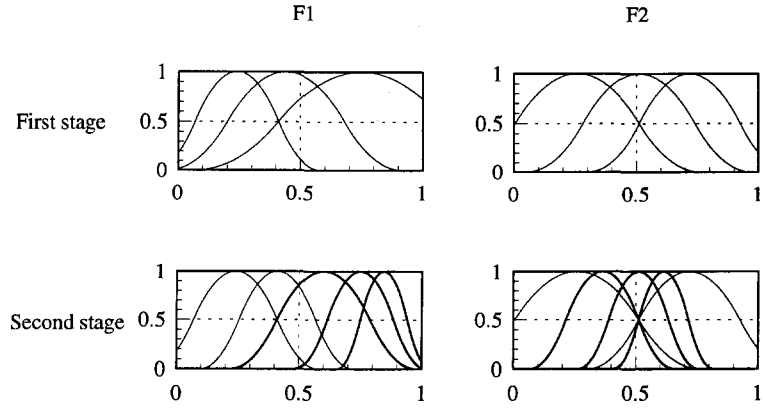

Fig. 5. Generated membership functions for *Cluster* data.

Table 1
Comparative study of recognition score on synthetic data

Data set	Model	*k*-nearest neighbours				Linear discrm. analys.	Conv. MLP	Fuzzy MLP	
		Crisp		Fuzzy				Stage 1	Stage 2
		$k=1$	$k=3$	$k=1$	$k=3$				
Random	*Train* b_1	93.5	90.5	93.5	92.5	83.0	90.0	99.5	98.5
	Test t_1	92.8	93.0	92.8	92.9	85.4	88.2	92.8	93.2
Cluster	*Train* b_1	92.0	92.0	92.0	93.0	89.0	91.0	92.5	100.0
	Test t_1	91.7	91.2	91.7	91.7	87.2	86.0	86.7	94.5

algorithms; the conventional MLP; and the linear discriminant analysis technique. We employed one hidden layer for the *Random* data and two hidden layers for the *Cluster* data, with 13 nodes in each such layer. For both data sets, the algorithm selected the subspace *high, medium* (corresponding to the first and second feature axes) for the second-level partitioning. This corresponds to the regions [0.518, 0.99] and [0.294, 0.756] along the two feature axes respectively. It can be verified from both Figs. 2 and 3 that this area corresponds to the most complicated decision region in the feature space with respect to the two pattern classes. It is observed that the proposed selective second-level partitioning provides appreciably better results on the *Cluster* data with respect to the other algorithms. This is perhaps because this pattern set has some inherent structure embedded in it, as compared to the randomness involved in the *Random* data set. However, the generalization capability of the fuzzy MLP on the separate test set is enhanced after the second-level partitioning in both cases. On the whole, the fuzzy MLP performed better than the k-nearest neighbours, linear discriminant analysis and the conventional MLP in case of both the synthetic data sets. It is to be noted that the two-level scheme helps avoiding too many partitions, associated with a related increase in the number of input neurons, which would also have resulted in overlearning and therefore poor generalization ability.

The model was next used on a set of 536 patient cases of various *Hepatobiliary Disorders* [2, 19]. There were nine input features corresponding to the results of different biochemical tests, viz., glutamic oxalacetic transaminate (GOT, Karmen unit), glutamic pyruvic transaminase (GPT, Karmen Unit), lactate dehydrase (LDH, iu/l), gamma glutamyl transpeptidase (GGT, mu/ml), blood urea nitrogen (BUN, mg/dl), mean corpuscular volume of red blood cell (MCV, fl), mean corpuscular haemoglobin (MCH, pg), total bilirubin (TBil, mg/dl) and creatinine (CRTNN, mg/dl). The 10th feature corresponded to the sex of the patient and was represented in binary mode as (1, 0) or (0, 1). The hepatobiliary disorders used for the four output classes were alcoholic liver damage (ALD), primary hepatoma (PH), liver cirrhosis (LC) and cholelithiasis (C). The network was trained by randomly selecting *perc*% samples from each representative pattern class of the data set. The remaining 100 − *perc*% samples constituted the test set. We selected $f_d = 5$ and $f_e = 1$ in (8) and *fnos* = 1 in (6), depending on the performance of the model, after several experiments.

Comparisons are provided in Table 2 with the results obtained by the fuzzy MLP model at the end of the first stage, i.e., before the use of the second-level selective partitioning of the feature space, as well as with the performances of the more conventional linear discriminant analysis method, the crisp and the fuzzy versions of k-nearest neighbours algorithm – using the *Hepatobiliary Disorders* data set. We used *perc* = 70% of the samples for training the network consisting of three hidden layers with 20 nodes in each such layer. The

Table 2
Comparative study of recognition score on *Hepatobiliary Disorders* data

Model		k-nearest neighbours				Linear discrim. analysis	Fuzzy MLP	
		Crisp		Fuzzy			Stage 1	Stage 2
		$k=1$	$k=3$	$k=1$	$k=3$			
best b_1		71.7	70.9	71.7	74.1	67.0	97.8	100.0
T	ALD	83.9	61.3	83.9	67.7	57.6	48.5	60.0
e	PH	77.2	68.4	77.2	71.9	64.7	70.3	75.9
s	LC	68.4	55.3	68.4	68.4	65.7	68.4	73.7
t	C	82.9	85.7	82.9	88.6	63.6	80.5	94.4
Net t_1		77.6	67.7	77.6	73.9	63.2	67.5	76.1

proposed algorithm is found to provide a good performance on both the training and test sets. Note that the conventional MLP provided very poor results, viz., 56.2% overall recognition score for the training set in this case, and was therefore omitted from the table.

What we emphasize in this paper is the selection strategy. The first stage partition provided by the fuzzy MLP can be further processed by other classification techniques. Some encouraging results have been obtained by applying the k-nearest neighbours rules separately to the *certain* and *doubtful* regions after detection in the first stage by the fuzzy MLP [10]. It seems natural that a properly trained scheme based on this strategy would outperform a single k-nearest neighbour rule applied on the whole sample.

In Tables 3–5 we study the effect on the recognition score (%) with the *Random*, *Cluster* and *Hepatobiliary Disorders* data, respectively, using different numbers of hidden layers and nodes. The number of hidden nodes in each case corresponds to the network configuration (found experimentally) providing good results with the given combination of number of layers and training set.

Table 4 indicates better results for the *Cluster* data set in all cases, after the second stage of partitioning, as compared to the case of the *Random* data set of Table 3. This is perhaps because of the absence of any

Table 3
Performance of fuzzy MLP on *Random* data before and after partitioning

Layers	3						4					
Nodes	11		12		13		10		11		12	
Stage	1	2	1	2	1	2	1	2	1	2	1	2
best b_1	99.0	99.5	93.5	99.5	99.5	98.5	99.0	98.0	93.0	99.5	93.0	95.0
perf p	74.0	69.0	56.5	81.5	58.5	63.0	95.5	94.5	86.5	90.0	86.5	53.0
Sweeps	940	320	310	290	980	420	860	350	420	480	510	150
Class 1	93.7	93.3	92.3	93.2	92.8	94.4	94.0	94.7	95.1	91.8	92.2	94.2
Class 2	89.9	90.6	89.6	91.1	92.8	91.5	89.1	90.6	88.4	93.5	90.6	89.4
Net t_1	92.1	92.2	91.2	92.3	92.8	93.2	92.0	93.0	92.3	92.5	91.5	92.2

Table 4
Performance of fuzzy MLP on *Cluster* data before and after partitioning

Layers	3						4					
Nodes	16		17		18		11		12		13	
Stage	1	2	1	2	1	2	1	2	1	2	1	2
best b_1	91.5	99.5	93.0	100.0	91.5	99.5	51.0	96.0	91.5	99.5	92.5	100.0
perf p	65.0	89.0	70.0	85.0	43.5	89.0	33.0	91.5	72.0	93.5	76.0	93.0
Sweeps	350	630	460	920	370	480	270	520	350	920	570	760
Class 1	76.0	89.0	76.2	89.8	76.2	93.1	0.0	81.7	74.2	88.8	78.1	94.5
Class 2	97.0	93.9	96.1	94.3	96.7	94.7	100.0	94.3	97.4	93.7	95.5	94.5
Net t_1	86.3	91.4	86.0	92.0	86.3	93.9	49.2	87.9	85.6	91.2	86.7	94.5

Table 5
Performance of fuzzy MLP on *Hepatobiliary Disorders* data before and after partitioning

Layers	3				4						5					
Nodes	20				20		15				10		20			
perc	10		50		10		50		70		10		50		70	
Stage	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
b_1	98.0	100.0	88.4	94.8	100.0	100.0	96.6	97.4	84.1	95.2	100.0	100.0	100.0	98.9	97.8	100.0
b_2	98.0	100.0	94.0	97.0	100.0	100.0	98.5	99.6	95.4	99.5	100.0	100.0	100.0	99.6	99.4	100.0
p	56.9	98.0	27.3	56.4	92.2	100.0	26.9	74.6	10.0	58.7	94.2	98.0	41.8	84.7	68.9	86.1
cycle	350	200	410	270	330	78	370	270	390	320	250	280	360	340	480	250
ALD	31.4	44.8	37.9	58.6	29.5	61.0	56.9	53.4	45.7	51.4	35.2	53.3	56.9	60.3	48.5	60.0
PH	44.7	55.3	59.5	73.0	45.3	60.9	77.5	77.5	77.7	81.5	98.2	66.5	74.1	79.8	70.3	75.9
LC	56.2	51.8	59.6	69.4	42.8	54.5	56.4	72.6	57.9	65.8	13.4	32.1	48.4	61.3	68.4	73.7
C	80.3	74.8	71.2	55.9	74.7	62.6	89.8	76.3	75.0	88.9	75.7	68.2	74.5	79.7	80.5	94.4
t_1	52.3	56.5	57.4	65.3	47.8	59.8	70.9	70.9	65.6	73.0	50.1	56.7	64.5	71.3	67.5	76.1
t_2	71.3	79.2	79.1	85.1	72.9	80.2	83.2	85.1	85.2	86.5	72.5	75.9	80.2	84.7	79.7	85.9

Table 6
Effect of f_d, f_e, *fnos* on the recognition score for *Hepatobiliary Disorders* data

f_d	3.0	4.0	5.0								6.0	7.0
f_e	1.0	1.0	0.25	0.5					1.0	2.0	1.0	1.0
fnos	1.0	1.0	1.0	0.8	0.9	1.0	1.1	1.2	1.0	1.0	1.0	1.0
b_1	98.6	98.6	100.0	99.2	97.8	99.4	97.8	99.2	97.3	85.5	95.1	95.9
b_2	99.7	99.2	100.0	99.4	98.6	99.4	98.9	99.7	97.3	94.6	98.1	97.5
p	88.5	79.4	96.0	90.4	77.5	89.3	82.9	87.2	56.3	8.6	64.9	61.7
t_1	73.0	71.1	72.4	72.4	73.0	74.2	73.0	74.2	72.4	71.1	75.4	73.6
t_2	85.2	89.5	90.1	89.5	87.7	86.5	90.1	90.1	89.5	86.5	85.8	85.8

inherent class structure in the synthetically generated pattern space of the *Random* data of Fig. 2, relative to the *Cluster* data of Fig. 3.

It is observed from Table 5 that generally better results are obtained for the data on *Hepatobiliary Disorders*, with less training cycles, after incorporating the proposed partitioning technique. Note that here the training set size *perc* was varied. Usually cases representing large training set size coupled with large network configuration (in terms of hidden layers and nodes) provided better results. Small training set sizes resulted in poor generalization capabilities on the test set.

Some results on the effect of f_d and f_e (controlling the amount of fuzziness in the output membership) and *fnos* (controlling the extent of overlap among the linguistic π-sets at the input) on the classification performance of a four-layered fuzzy MLP with 25 nodes in each hidden layer, using 70% of the data on *Hepatobiliary Disorders* for training, are provided in Table 6.

5. Conclusions and discussion

A neuro-fuzzy classifier, using two-level selective partitioning of the input feature space, has been described. The model could handle uncertainty both at the input and the output. The input to the network was modelled in terms of the primary linguistic properties low, medium and high, using pi-functions. The centres and radii of these π-sets were automatically determined from the distribution of the training patterns. The performance of the model at the end of the first stage was used as a criterion for guiding the selection of the appropriate partition to be further subdivided at the second stage, in order to improve the effectiveness of the model. The two-level scheme helped avoid the problem of a large increase in the number of input neurons, thereby preventing cases of overlearning. A comparative study with the performance of the model at the end of the first phase as well as with those obtained by the more conventional linear discriminant analysis and the k-nearest neighbours techniques indicated the superiority of the algorithm described.

Medical information such as results of biochemical tests and/or the diagnosed disorder(s) are often ambiguous and/or fuzzy [2]. Hence incorporation of fuzziness at input and output levels was found to be more effective in modelling such problems. The skewness of the data set under consideration could be appropriately handled by the chosen input description that automatically determined the centres and radii of the linguistic pi-sets.

Although the experiments with the generated data have been carried out using a completely separable set of classes, the proposed two-level partitioning scheme can be even more effective in cases of complex classification structures. Due to the strategy adopted here, the regions of overlapping would be notified as *doubtful* in the first stage and thereby lead to a separate consideration of *certain* regions applying a simple fuzzy MLP configuration. This simple structure, and hence less amount of trainable parameters, can be viewed as the background for a higher generalization ability over the *certain* regions. On the other hand, the overlapping regions are paid special attention, developing a proper network configuration and then training that in more detail on this data. Therefore, applying a different classification strategy on the *less certain* (*doubtful*) regions detected at the end of the first stage, one can achieve better performance for the overall scheme.

Acknowledgements

This work was carried out when both the authors were in the European Laboratory for Intelligent Techniques Engineering, Aachen, Germany. The authors are grateful to Dr. Y. Hayashi of Ibaraki University, Japan, for the data. We also thank Prof. Dr. Dr. h.c. H.-J. Zimmermann of Aachen Institute of Technology, for his interest in this study and his kind help in obtaining the data. The hospitality shown by the staff of ELITE, during the authors' stay there, is gratefully acknowledged. The suggestions provided by the referees helped us in improving the quality of the paper.

References

[1] J.C. Bezdek and S.K. Pal, Eds., *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data* (IEEE Press, New York, 1992).

[2] Y. Hayashi, Neural expert system using fuzzy teaching input and its application to medical diagnosis, *Proc. 2nd Internat. Conf. on Fuzzy Logic and Neural Networks*, Iizuka, Japan (1992) 989–993.

[3] K. Hirota and W. Pedrycz, Geometric-logical pattern classification, *Proc. 2nd Internat. Conf. on Fuzzy Logic and Neural Networks*, Iizuka, Japan (1992) 675–678.

[4] H. Ishibuchi, K. Nozaki and H. Tanaka, Distributed representation of fuzzy rules and its application to pattern classification, *Fuzzy Sets and Systems* **52** (1992) 21–32.

[5] H. Ishibuchi, K. Nozaki and H. Tanaka, Efficient fuzzy partition of pattern space for classification problems, *Proc. 2nd Internat. Conf. on Fuzzy Logic and Neural Networks*, Iizuka, Japan (1992) 671–674.
[6] H. Ishibuchi, K. Nozaki and H. Tanaka, Efficient fuzzy partition of pattern space for classification problems, *Fuzzy Sets and Systems* **59** (1993) 295–304.
[7] G.J. Klir and T. Folger, *Fuzzy Sets, Uncertainty and Information* (Addison-Wesley, Reading, MA, 1989).
[8] B. Kosko, *Neural Networks and Fuzzy Systems* (Prentice Hall, New Jersey, 1991).
[9] L.I. Kuncheva, 'Change-glasses' approach in pattern recognition, *Pattern Recognition Lett.* **14** (1993) 619–623.
[10] L.I. Kuncheva and S. Mitra, A two-level classification scheme trained by a fuzzy neural network, *Proc. 12th Internat. Conf. Pattern Recognition*, Jerusalem, Israel (October, 1994).
[11] R.P. Lippmann, An introduction to computing with neural nets, *IEEE Acoust. Speech Signal Process. Magazine* **4** (1987) 4–22.
[12] D.P. Mandal, C.A. Murthy and S.K. Pal, Formulation of a multi-valued recognition system, *IEEE Trans. Systems Man Cybernet.* **22** (1992) 607–620.
[13] S. Mitra, Fuzzy MLP based expert system for medical diagnosis, *Fuzzy Sets and Systems* **64** (1994).
[14] S.K. Pal and S. Mitra, Multi-layer perceptron, fuzzy sets and classification, *IEEE Trans. on Neural Networks* **3** (1992) 683–697.
[15] Y.H. Pao, *Adaptive Pattern Recognition and Neural Networks* (Addison-Wesley, Reading, MA, 1989).
[16] L.A. Rastrigin and R.H. Erenstein, *Method of Collective Recognition* (Energoizdat, Moscow (in Russian), 1981).
[17] D.E. Rumelhart and J.L. McClelland, Eds., *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Vol. 1 (MIT Press, Cambridge, MA, 1986).
[18] H. Takagi, N. Suzuki, T. Koda and Y. Kojima, Neural networks designed on approximate reasoning architecture and their applications, *IEEE Trans. on Neural Networks* **3** (1992) 752–760.
[19] K. Yoshida, Y. Hayashi, A. Imura and N. Shimada, Fuzzy neural expert system for diagnosing hepatobiliary disorders, *Proc. 1990 Internat. Conf. Fuzzy Logic and Neural Networks*, Iizuka, Japan, (1990) 539–543.
[20] L.A. Zadeh, Making computers think like people, *IEEE Spectrum* (August 1984) 26–32.
[21] H.J. Zimmermann, *Fuzzy Set Theory – and its Applications* (Kluwer, Boston, 1991).