Proc. 9th IEEE Int. Conference on Tools with Artificial Intelligence (ICTAI'97), Nov. 1997, Newport Beach, California, USA

A Fuzzy Neural Network Approach to Classification Based on Proximity Characteristics of Patterns

K. Blekas* A. Likas‡ A. Stafylopatis*

*National Technical University of Athens
Department of Electrical & Computer Engineering
157 73 Zographou, Athens, Greece
E-mail: kblekas/andreas@cs.ntua.gr

‡University of Ioannina
Department of Computer Science
451 10 Ioannina, Greece
E-mail: arly@cs.uoi.gr

Abstract

A neural network classifier is presented, which is based on geometrical fuzzy sets. Starting from the construction of the Voronoi diagram of the training patterns, an aggregation of Voronoi regions is performed leading to the identification of larger regions belonging exclusively to one of the pattern classes. The resulting scheme is a constructive algorithm that defines fuzzy clusters of patterns. Based on observations concerning the grade of membership of the training patterns to the created regions, decision probabilities are computed through which the final classification is performed. Experimental results concerning several classification problems indicate that the proposed method achieves high classification rates and compares favorably with other well-known approaches.

1. Introduction

Computational intelligence embraces many approaches to pattern recognition that result from the combination of techniques belonging to different fields. In this sense, several models combining fuzzy systems and neural networks have been developed that build efficient pattern classifiers exploiting the particular advantages offered by each technique in a synergistic manner [3, 7, 16, 17]. Most of these methods use the training set to produce geometrical hyperboxes and then compute suitable membership functions in order to specify the desicion boundaries of pattern classes.

A popular approach to the partitioning of the input space given a set of points is based on the construction of *Voronoi diagrams*. A Voronoi diagram, also known as Dirichlet tesselation or Thiessen polygon, is a partition of the pattern space into convex regions. Each of these regions contains points with minimum distance from a specific point (the region generator) compared to their distance from the other points used for the construction of the diagram. Voronoi diagrams have been largely used in pattern recognition problems because they provide a topological division of the pattern space based on the *nearest neighbor* property. This property has been widely exploited in pattern recognition approaches.

The proposed fuzzy neural approach creates fuzzy sets from the Voronoi diagram of the training patterns and builds class boundaries in a statistical manner. Given a set of points in the feature space, the resulting Voronoi diagram can be viewed as a puzzle and the Voronoi regions as the pieces of this puzzle. We can assemble neighboring pieces according to their position and class, in order to appropriately specify the boundaries between classes. This formulation leads to a *reduced Voronoi* diagram where the new broader regions contain more than one adjoining Voronoi regions having the same class label. The resulting aggregate regions are no longer convex and may be considered as fuzzy sets by defining membership functions indicating the degree of belongingness of points of the input space to each region. Each fuzzy set is characterized by a set of hyperplanes (separating the corresponding region from other regions) and a class label.

After constructing the fuzzy sets, decision probabilities are computed based on the density of membership values for each region and the respective performance in the selection of the correct region. Through discretization of the membership axis, a probabilistic function is created that establishes a correspondence between membership values in a specific region and the probability of correct classification. Mapping the above procedure into a neural architecture we are able to obtain an algorithm for the design of fuzzy neural networks for pattern classification.

Voronoi diagrams, with reference to their use in constructing neural network architectures, are briefly discussed in Section 2. Section 3 deals with the construction of the reduced Voronoi diagram, while Section 4 concerns the

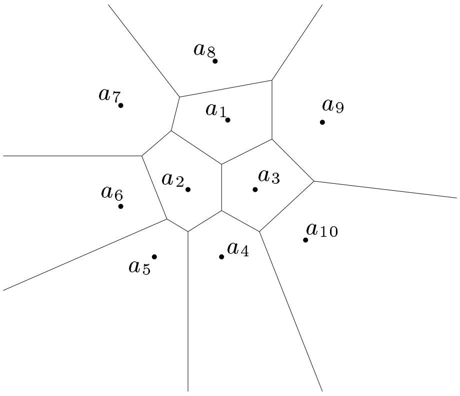

Figure 1. Classical Voronoi diagram

computation of decision probabilities. The architecture of the proposed fuzzy neural classifier is described in Section 5. Experimental results of the application of the proposed scheme and comparisons with other established classification methods are provided in Section 6. Finally, Section 7 contains conclusions and directions for further research.

2. Voronoi Diagrams and Neural Networks

The Voronoi diagram or Dirichlet tesselation is a fundamental concept in computational geometry with many applications. Voronoi diagrams reveal proximity information about a set of given points in a very explicit and computationally useful manner. This makes them applicable to many diverse areas, among which are biology, visual perception and crystallography [1].

Let $D(a, b)$ denote the Euclidean distance between two points a and b in $\Re^d$. Given a finite set $\mathbf{A} = \{a_1, \ldots, a_N\}$ of points in $\Re^d$, a *Voronoi region* V_n $(n = 1, \ldots, N)$ is defined as the set of points:

$$V_n = \{x \in \Re^d \mid D(x, a_n) \leq D(x, a_k) \;\; \forall k \neq n\} \qquad (1)$$

Each Voronoi region V_n contains those points of $\Re^d$ for which the point a_n is the closest. The partition of $\Re^d$ implied by the Voronoi regions $V_1, \ldots, V_N$ is the Voronoi diagram for the set $\mathbf{A}$. Each element of $\mathbf{A}$ is called a *generator* of the Voronoi diagram. A common *boundary* of two Voronoi regions is the perpendicular bisector (hyperplane) of the segment joining the pair of respective generators. Thus, each Voronoi region is defined as the interesection of a finite number of halfspaces determined by the hyperplanes. A point at which boundaries of three or more Voronoi regions meet is called a Voronoi point. An example of a Voronoi diagram in 2-dimensional space is given in Fig. 1.

Several algorithms for the construction of Voronoi diagrams have been proposed. Classical Voronoi diagrams can be constructed by obtaining the convex hull of the given set of points [14, 1, 2] or by incremental insertion of the Voronoi regions [9, 6, 10].

The application of Voronoi diagrams to the design of neural networks has been considered recently. In [13] two neural network construction algorithms for pattern classification are proposed that rely directly or indirectly on the Voronoi tesselation of the input space produced by the given training patterns. A systematic procedure for designing neural networks following the same principle is proposed in [5, 8]. These methods specify the architecture of the neural model based on the construction of corresponding Voronoi diagrams for the training data. They also describe ways to specify the values of the connection weights and thresholds of each node at all layers of the neural architecture. The neural network design is robust and adaptable in order to accomodate new training patterns. It must be noted that these approaches essentially suggest ways to implement a Voronoi diagram using a neural architecture and the classification behaviour of the resulting networks is equivalent to that of nearest neighbor techniques.

In [4] an analogous construction approach was presented which incorporated the idea of fuzzy classification by defining fuzzy decision boundaries for the regions of the tesselation. This scheme is based on an approximate incremental construction of Voronoi diagrams and allows on-line supervised learning using appropriately defined fuzzy membership functions.

3. Reducing the Voronoi regions

Consider a classification problem with d continuous attributes, such that N d-dimensional patterns belong to K distinct classes. By constructing the Voronoi diagram of these generators the d-dimensional feature space is divided into N regions reflecting the *proximity* property.

In pattern recognition we are mostly interested in dividing the input space into a number of regions (clusters) characterized by the same class label. In general, the number of clusters is much smaller than the number of patterns. The Voronoi diagram divides the space into a number of compartments as many as the input patterns, which is not very convenient. Nevertheless, if we managed to join Voronoi regions that are of the same class, we would only consider clusters of patterns. This can be achieved by removing all those hyperplanes (boundaries) of the Voronoi regions that bisect pairs of patterns (generators) belonging to the same class. In this way, the feature space is divided into regions larger than Voronoi regions, where each of these regions is associated with exactly one class label. We shall refer to these major regions as *class regions*, as they come from the union of neighboring Voronoi regions whose generators correspond to the same class. Fig. 2 illustrates such a construction on the 2-dimensional space for a set of 10 input

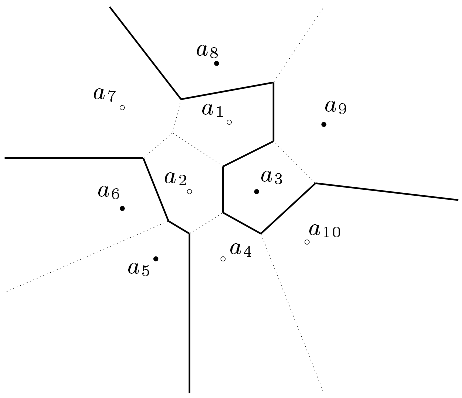

Figure 2. Class regions

points belonging to 2 classes (dotted lines are present in a classical Voronoi diagram).

More specifically, we can define a *class region* as the union of a set of Voronoi regions, such that each region is adjoining (has common boundaries) to at least one other region of the set and their generators belong to the same class. Each class region is characterized by a set of equations that describe the hyperplanes defining class borders. The number of new regions may be equal to or greater than the number K of classes. We only keep large class regions, since small regions containing few generators may be considered as outliers or black holes inside large clusters and therefore they can be ignored.

In fact, the Voronoi diagram can be treated as an undirected graph, where vertices represent generators (or equivalently regions) and edges join vertices corresponding to adjoining Voronoi regions. This graph configuration is equivalent to the construction known as *Delaunay triangulation*. In order to form class regions we start with an arbitrary Voronoi region and mark it as belonging to the first class region. Then we perform a search of the graph structure. The information available from the original construction of the Voronoi diagram is sufficient for performing the search. (Depth-first or breadth-first search could be used.) All regions connected to the starting region and bearing the *same class label* are marked as belonging to the same class region. Neighboring regions bearing a different class label are left unmarked to be included in some subsequent aggregation. A similar procedure is followed from any region marked during the search. When the search is exhausted, i.e. no more Voronoi regions can be included in the current class region, a new unmarked region is selected and the procedure is repeated to construct a new class region, until there are no more unmarked Voronoi regions.

In order to encourage the formation of *nearly-convex* class regions we impose the following restriction during the above aggregation phase. The search can only proceed from the current Voronoi region if the number of neighboring regions bearing the same class label as the current one exceeds a given threshold value λ. If this criterion is not satisfied, the remaining neighboring regions of the current one are left unmarked (independently of their class label) and the search is continued from some other region. The choice of the value of λ is related to the average number of boundaries of a Voronoi region which in turn depends upon the dimension of the feature space. In our implementation, appropriate values of the parameter λ were determined experimentally for the different classification problems considered.

After construction of the reduced Voronoi diagram, it is possible to exploit proximity information for the purposes of classification. In order to formulate the problem as a fuzzy classification problem, we must give an estimation of how much a new pattern belongs to each class region, thus considering class regions as fuzzy sets. When a new input pattern $a = (a_1, \ldots, a_d)$ is presented, an appropriate fuzzy membership value (in $[0, 1]$) is computed. The membership function $\mu_i(a)$ for the ith class region must measure the degree to which the given pattern falls inside or outside the region. This can be considered as a measurement of how far the pattern is situated from all the hyperplanes which define the boundaries of the class region. When the pattern a is in the interior of the region and far from the hyperplanes then the value of $\mu_i(a)$ is large, which means that the point is close to some generator having participated to the formation of that class region. When the pattern falls outside the region then the membership value approaches zero, which means that the point is close to some generator belonging to a different class region.

A function respecting the above guidelines is the average value of the exponential differences between the vertical distances $x_h(a)$ of the input pattern from all hyperplanes h supporting the class region i and the distances l_h of the respective generators from each hyperplane h. (Clearly, within each class region, there is a generator associated with each supporting hyperplane of the region.) It must be noted that the information concerning the distances l_h is already known from the original Voronoi diagram, and is stored for each hyperplane h. The vertical distance $x_h(a)$ of a pattern $a = (a_1, \ldots, a_d)$ from a hyperplane h (described by the equation $\pi_{h1}x_1 + \ldots + \pi_{hd}x_d + \pi_{h,d+1} = 0$) is computed as follows:

$$x_h(a) = \frac{|\sum_{j=1}^{d} \pi_{hj} a_j + \pi_{h,d+1}|}{\sqrt{\sum_{j=1}^{d} \pi_{hj}^2}} \quad (2)$$

Each hyperplane h divides the pattern space into two halfspaces. Consider the quantities $u_{ih}(a)$ which take the values 1 or -1 depending on whether or not the generator corresponding to hyperplane h and belonging to region i is situated in the same halfspace (defined by h) as the pattern a.

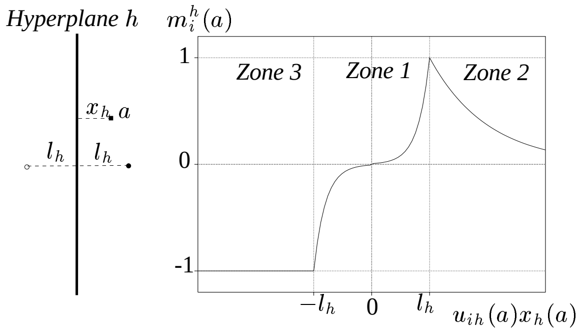

Figure 3. Fuzzy decision boundaries

The membership function of class region i can be computed as follows:

$$\mu_i(a) = \frac{1}{2|H_i|} \sum_{h \in H_i} m_i^h(a) + \frac{1}{2} \qquad (3)$$

where H_i is the set of hyperplanes defining class region i (having cardinality $|H_i|$) and m_i^h has the following form :

$$m_i^h(a) = \begin{cases} u_{ih}(a)\exp(\frac{-|x_h(a)-l_h|}{\sigma_1}) & \text{if } x_h(a) \leq l_h \\ u_{ih}(a)\exp(\frac{-|x_h(a)-l_h|}{\sigma_2}) & \text{if } x_h(a) > l_h \\ & \text{and } u_{ih}(a) = 1 \\ -1 & \text{otherwise} \end{cases} \qquad (4)$$

A graphical representation of Eq. 4 is shown in Fig. 3, which represents m_i^h as a function of the quantity $u_{ih}(a)x_h(a)$. Based on this quantity, we divide the space into three *zones* with respect to the hyperplane h, each zone being characterized by different properties. In the *first zone* the pattern is close to the hyperplane ($x_h(a) \leq l_h$) independently of the side of the region border on which it is situated. Starting from the value -1 (when $x_h(a) = l_h$ and $u_{ih}(a) = -1$), the value of m_i^h increases with a steep slope until it reaches the highest value 1 (when $x_h(a) = l_h$ and $u_{ih}(a) = 1$). The *second zone* is related to the case where the pattern and the generator are on the same halfspace but the pattern is far from the hyperplane ($x_h(a) > l_h$ and $u_{ih}(a) = 1$). Although the pattern lies on the good side of the hyperplane with respect to the class region, we must be cautious to avoid circumstances where the pattern is at long distance from the hyperplane and does not belong to the class region. For this reason the value of m_i^h decreases to zero (but with a smoother slope) as the vertical distance $x_h(a)$ from the hyperplane grows. From the above specification it is clear that the value of σ_1 in the first zone must be higher than the value of σ_2 in the second zone to achieve the desired slope. The *third zone* represents the case where the pattern is situated far outside the region of interest and far from the hyperplane h ($x_h(a) > l_h$ and $u_{ih}(a) = -1$). In this case, m_i^h takes its lowest value -1.

It must be noted here that $m_i^h(a)$ does not include precise information about whether or not the pattern is situated inside the class region. Even when the pattern and the generator are in the same halfspace with respect to the hyperplane, we are not aware of what happens with other hyperplanes as we individually examine each hyperplane and not the whole region. After computing all the $m_i^h(a)$ values, we obtain a global estimation of the degree of belongingness of pattern a to the region i through the value of the membership $\mu_i(a)$.

The above membership function takes into account useful proximity information provided by the characteristics of the original Voronoi construction, such as the equations describing hyperplanes, the position of generators relative to hyperplanes and the vertical distances l_h. In addition, it exploits information related to class regions, such as the hyperplanes defining the boundaries of each class region.

4. Computing Decision Probabilities

In the previous sections we have described the first phase of the proposed approach to pattern classification that incorporates the construction of the Voronoi diagram corresponding to the training patterns, the integration of Voronoi regions to a number of class regions and the definition of appropriate membership functions. With this kind of preprocessing the problem of selecting the correct class is transformed to the problem of selecting the correct region. In this section we describe how the membership values of the training patterns are used to construct models providing decision probabilities that a pattern with a given membership value can be assigned to a specific region.

After having computed the membership values corresponding to an input pattern we could perform the classification procedure simply by selecting the class of the region with the maximum membership value. This is what happens in most fuzzy approaches to pattern classification (e.g in the fuzzy min-max network). Unfortunately, this decision scheme does not seem to provide good classification results, since some regions tend to constantly exhibit higher membership values compared to other regions. It seems more effective to evaluate the membership of a given pattern to a region by taking into account the distribution of membership values of the training patterns to this region. This leads to the construction of a probability model for each class region, which provides useful information for the selection of the appropriate region during classification.

The construction of the probability models is based on the search for ranges of membership values that have more chance to lead to the successful selection of a region. More specifically, considering class region i, the interval $[0, 1]$ of membership values is divided into a number L_i of equal-size cells. To each cell v ($v = 1, \ldots, L_i$) we assign a probability value p_i^v computed as the *percentage of the training patterns*

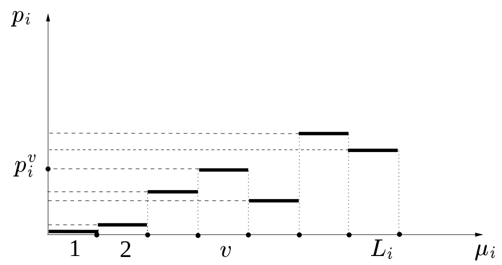

Figure 4. Membership histogram

belonging to region i that have their membership value in the cell v. An example of the above histogram construction is displayed in Fig. 4.

It should be observed that, after the membership values of training patterns to each region have been computed, it is possible to view the classification problem as mapping from the space of membership vectors to the set of classes, $c : (\mu_1, \ldots, \mu_R) \rightarrow \{1, \ldots, K\}$ (where R is the number of class regions and K is the number of classes). Such a mapping can be easily constructed using for example a multilayer perceptron trained by the backpropagation algorithm or any of its variants. Although this approach is intuitively more appealing and exhibited excellent performance on training sets, its performance on the test set was inferior (in all examined datasets) compared with the approach based on the probability model.

5. Neural Network Implementation

In the previous sections we have shown how one can construct fuzzy sets (corresponding to class regions) starting from a Voronoi diagram, as well as how a model of probabilities can be built for each fuzzy set using the distribution of membership values. The proposed construction algorithm can be summarized into the following steps:

1. Construct the classical Voronoi diagram of a set of N patterns $\mathbf{A} = \{a_1, \ldots, a_N\}$, in a d-dimensional space.

2. Reduce the Voronoi diagram into a number of class regions.

3. Discard small class regions and let R be the number of the remaining large class regions.

4. For each pattern a_j, $j = 1, \ldots, N$, compute the membership values $\mu_i(a_j)$, $i = 1, \ldots, R$.

5. For each region i, $i = 1, \ldots, R$, categorize the membership values in a histogram using a number L_i of equal-size cells in $[0, 1]$.

6. Compute selection probabilities p_i^v, $i = 1, \ldots, R$, $v = 1, \ldots, L_i$.

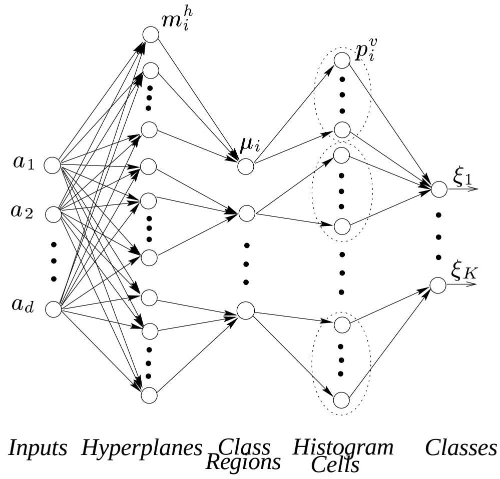

Figure 5. Neural network architecture

In order to use the method for the classification of a new pattern, first the membership values of the pattern to each region i are computed. Then the corresponding probabilities p_i^v are determined (where v represents the cell containing the membership value of the pattern) and the region i with maximum p_i^v is selected. The class of this region is considered as the final classification desicion.

The above desicion approach can be implemented by means of a neural network architecture as illustrated in Fig. 5. The architecture consists of five layers and connections exist only between successive layers. The first layer is the input layer having as many nodes as the dimension of patterns. The nodes in the second layer represent hyperplanes that define class regions. For each class region i there are $|H_i|$ nodes, one for each hyperplane supporting that class. As a hyperplane separates two regions, there will generally be two nodes referring to the same hyperplane (except for hyperplanes supporting small regions that were discarded during construction). Each second layer node computes the value of the function m_i^h for an input pattern using Eq. 4. The third layer contains as many nodes as the number R of class regions. The output of each node i of this layer provides the membership value μ_i of the pattern to the corresponding region as computed in Eq. 3. The connections between nodes of the second and third layer assume binary values 1 or 0 to associate regions with their supporting hyperplanes.

The fourth layer implements the membership histogram. Each region i of the third layer is connected to L_i nodes of the fourth layer corresponding to the cells of the histogram. Each such node v ($v = 1, \ldots, L_i$) fires only in the case where the μ_i value falls inside the corresponding cell and provides the respective probability p_i^v, otherwise the output of the node is zero. This representation allows an efficient implementation of the histogram by means of simple nodes yielding a fixed output on an on-off basis.

The fifth layer embodies one node for each of the K classes. If region i has class label k then the set of L_i nodes of the fourth layer (representing the histogram of region i) is connected to node k of the fourth layer. In other words, the connections between nodes of the fourth and fifth layer take binary values 1 or 0 to associate class regions (histogram cells) with class labels. The output ξ_k of each node k of the last layer is taken equal to the maximum of the outputs (probabilities p_i^v) of the cell nodes connected to that node. Finally, the class k with the maximum ξ_k is the decision of the fuzzy neural classification network.

When a new pattern a is applied to the network, the membership values $\mu_i(a)$ of the pattern to each class region i are initially computed. The computation proceeds by determining the probabilities p_i^v corresponding to the respective histogram cells. The decision of the network is the class of the region with the maximum probability p_i^v, expressed by the quantity ξ_k. As is the case with other neural network designs based on Voronoi diagram information [5, 8, 4], the method has the ability to completely define the neural structure, namely the number of hidden layers and nodes of each layer along with the connection values between nodes of successive layers. The proposed neural architecture incorporates an additional layer with respect to other approaches accounting for the computation of decision probabilities.

6. Experimental results

We have studied the proposed fuzzy neural network classifier on a variety of classification problems, by performing experiments with known datasets, some of which are noisy containing hard examples. In addition we have conducted comparative experiments with many other well-studied classification techniques.

The first dataset we used to train and test our fuzzy neural classifier was the Fisher's Iris dataset. Iris data is a collection of 150 four-dimensional feature vectors in three separate classes, 50 for each class. We considered a training set and a test set of size 75, each of them containing 25 examples of each of the three iris classes. We have also tried the James Cook University Thyroid gland database. Thyroid database contains a set of 215 feature vectors (belonging to one of three classes) that contain 5 continuous attributes. The three classes represent the classification of a patient's thyroid to euthyroidism, hypothyroidism and hyperthyroidism respectively. The database is divided into 150 instances of the first class, 35 instances of the second and 30 of the last class. We used a training set of size 100 (in a manner proportional to the distribution of patterns to classes), while the remaining data set (of size 115) was used for testing.

Another dataset used in our experiments was the synthetic two-class problem taken from Ripley [15]. It is a realistic problem with 1250 2-dimensional patterns that belong to 2 classes, where 250 and 1000 patterns were used for training and testing respectively. Finally, we have also considered the clouds database. Clouds is an artificial collection of 5000 two-dimensional patterns divided into two equal size sets that correspond to two overlapping decision classes. The first 2000 clouds data were handled as the training set while the remaining 3000 data applied to the constructed fuzzy neural classifier for testing. Figs. 6 and 7 provide a graphical representation of the last two datasets.

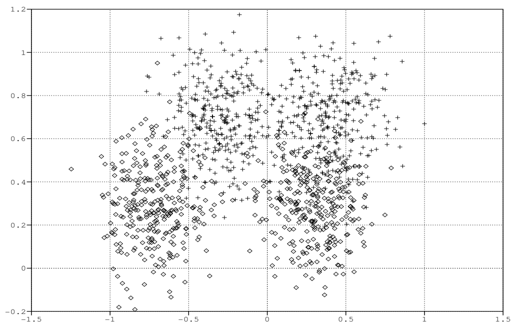

Figure 6. Graphical representation of the synthetic two-class dataset

The parameters in our approach are the number of cells L_i for each class region i and the construction parameter λ, as well as the σ_1, σ_2 values for the membership function (Eq. 4). Although we could select a different value L_i for each region (depending on characteristics of patterns belonging to the region), we have opted for the use of a global value L for all class regions, depending on the difficulty of the considered problem. In the first two problems (iris and thyroid), that are relatively easy, the specification of L was not very critical. For almost all experiments with different values of L ($L \geq 4$) the network achieved the same classification rate on training and testing phases. Moreover, the number of class regions obtained in the case of these two datasets was exactly the same as the number of classes (i.e. $R = 3$), suggesting that the classes in these problems are well separated. Specifically, in the case of the iris dataset a total of 306 hyperplanes were needed for the separation of class regions, where 142, 262 and 208 of these were used to define each one of the three regions respectively. In the thyroid problem the required number of hyperplanes was equal to 676 and the three classes were defined using 527, 386 and 439 hyperplanes repsectively.

The other two datasets (synthetic and clouds) are noisy. As illustrated in Figs. 6, 7 the class boundaries in both problems are not very clear, therefore classification is difficult. The Voronoi diagram in the case of the synthetic problem was reduced to 10 class regions, where only two of

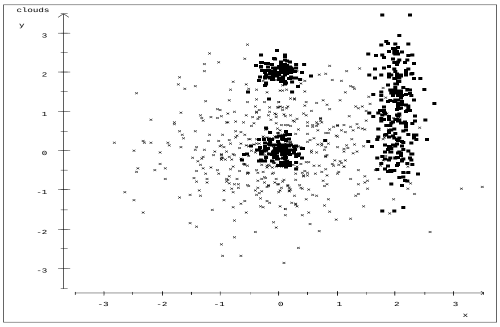

Figure 7. Graphical representation of the clouds dataset

them were big enough including a large number of patterns ($R = 2$). The other eight regions were very small containing only one or two patterns and were rejected as outliers. The two selected class regions were separated with 106 and 112 hyperplanes from a set of 140 hyperplanes of the reduced Voronoi scheme (the difference is due to discarded regions). In the case of the clouds dataset, the number of constructed class regions was equal to 68 out of which we eventually chose only four ($R = 4$). The number of required hyperplanes was 258, 464, 450 and 117 respectively. In this kind of problems, the selection of the parameter L was critical. We observed that in the range of $L = 10, \ldots, 20$ the fuzzy neural network classifier had the same classification behaviour achieving the best classification rate.

In what concerns the effect of the parameter λ on the classification performance the following observations can be made. As stated in Section 3, greater values of λ, lead to more convex and uniform class regions, since the regions are derived from the union of Voronoi regions having at least λ neighboring regions of the same class. The selection of the best value for this parameter depends on the dimension of the pattern space. In the case of the first two examined problems there was a large interval of λ values leading to the construction of exactly the same reduced Voronoi diagrams ($\lambda = 1, \ldots, 9$ for the iris problem and $\lambda = 1, \ldots, 15$ for the thyroid problem) and thus yielding the same best classification rate. As the value of λ was growing, the class regions contained less generators and the number of such regions increased, leading to degradation in the performance of the method. In the last two classification problems the effect of the parameter λ was identical and exhibited the following characteristics. For the first two values ($\lambda = 1, 2$) we obtained similar rates and similar constructed regions. As the value of λ was increased, the network achieved better classification rates attaining the highest values for $\lambda = 3$ and $\lambda = 4$ for the synthetic and clouds problems respectively. Although the number of constructed class regions in these cases was the same as for the previous values of λ, they contained less generators and were characterized by increased generalization capabilities. It must be noted that in all the examined classification problems, when the value of the threshold λ was big enough, the reduced Voronoi scheme was the same as the original Voronoi diagram. Finally, as far as the σ_1, σ_2 parameters are concerened, the appropriate values were determined experimentally and were generally less than 3.

In order to assess the potential of the proposed classifier, we have also applied some other known neural network methods to the same datasets. For that reason we have used the fuzzy min-max classification neural network (FMM), a multilayer perceptron (MLP) trained with the backpropagation algorithm, as well as the learning vector quantization (LVQ) algorithm. We have applied also a simple nearest-neighbor approach (1-NN), which consisted of assigning to a given pattern the class label of its closest training pattern.

The fuzzy min-max classifier [16, 12] is an on-line supervised learning classifier whose operation and training are based on the concept of hyperbox fuzzy sets. The MLP neural network was a single hidden-layer network with a learning rate of 0.09 and a momentum rate of 0.9. The specific LVQ scheme used was the LVQ1 algorithm [11] having a learning rate equal to 0.03. The best results were obtained by considering 6, 6, 6 and 8 centers for the four databases respectively. For these approaches several experiments were conducted using different seed values and taking the best classification rate as the performance measure.

Table 6 summarizes the performance of the proposed classification scheme, denoted RVoD (Reduced Voronoi Diagram), in comparison with the FMM, MLP, LVQ and 1-NN approaches in terms of the classification rate on the test set for the four selected databases. The superiority of the proposed fuzzy neural classifier is apparent. The results indicate that the simple nearest-neighbor scheme (and consequently the approaches based on pure Voronoi diagrams) have been considerably improved so as to outperform the other well-known algorithms.

If we compare our approach to the fuzzy min-max algorithm, we can observe that, in the case of the fuzzy min-max algorithm, the created hyperboxes have only the information of their min and max values along each dimension. The hyperboxes are of finite volume (therefore the technique can be considered as 'local') and their geometrical structure is not so flexible to be able to adapt to strange class boundaries. In the proposed approach, hyperplanes are used to separate classes (therefore the technique can be considered as 'global'), the class regions have in most cases a convex shape and the nearest neighbor criterion is more explicitly taken into account. Apart from the geometrical differences, the FMM algorithm is based on the selection of the hyperbox with the maximum membership value and accept as final

Table 1. Comparative results (classification rates)

Problem	*RVoD*	*FMM*	*MLP*	*LVQ*	*1-NN*
Iris	97.33%	97.33%	94.67%	97.33%	94.67%
Thyroid	98.26%	97.39%	94.78%	95.65%	95.65%
Synthetic	91.14%	90.60%	88.00%	89.50%	85.50%
Clouds	88.27%	84.83%	79.77%	80.13%	84.27%

decision the class of the selected hyperbox. On the other hand the proposed fuzzy neural classifier is able to decide in a more informative manner, since the decision is based on the distribution of the membership values of the training patterns to the constructed regions. Moreover, the proposed method requires no normalization of the feature values of the patterns as is the case with many other approaches.

7. Conclusions

We have proposed a neural network classifier that is based on geometrical fuzzy sets. The approach is based on the construction of Voronoi diagrams in the pattern space and the creation of region aggregates inside the Voronoi puzzle. For the constructed class regions, decision probabilities are computed in terms of the distribution of membership values to these regions. The whole technique can be implemented by means of a five-layer feedforward neural network architecture.

Experimental results concerning several classification problems indicate that the proposed method is effective in terms of the rate of correct classification. Comparisons with other established approaches have shown that the proposed scheme can overcome the difficulties arising from the problem of overlapping classes. Moreover, it has the characteristic of maintaining the powerful geometrical features of the Voronoi structure as well as of creating efficient decision boundaries through the statistical processing of membership values.

This work allows us to further experiment with the use of proximity based approaches to the construction of fuzzy neural classifiers and to discover more efficient techniques in the area of soft decision making. Since the complexity of the construction of Voronoi diagrams becomes high as the dimension of the feature space grows, we are interested in applying effective geometrical algorithms that can suggest neighbors of a given pattern (in the sense of the Voronoi diagram) for large dimensional problems.

References

[1] F. Aurenhammer. Voronoi Diagrams - A Survey of a Funtamental Geometric Data Structure. *ACM Computing Surveys*, 3:345–405, 1991.

[2] C. Barber, D. Dobkin, and H. Huhdanpaa. The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software*, 22:469–483, 1996.

[3] J. Bezdek and S. Pal. *Fuzzy Models for Pattern Recognition*. IEEE Press, 1992.

[4] K. Blekas, A. Likas, and A. Stafylopatis. A Fuzzy Neural Network Approach Based on Dirichlet Tesselations for Nearest Neighbor Classification of Patterns. In *Proc. of IEEE Workshop on Neural Networks for Signal Processing (NNSP '95)*, pages 153–161, 1995.

[5] N. Bose and A. Garga. Neural Network Design Using Voronoi Diagrams. *IEEE Trans. on Neural Networks*, 4:778–787, 1993.

[6] A. Bowyer. Computing Dirichlet Tesselations. *The Computer Journal*, 24:162–166, 1981.

[7] G. Carpenter, S. Grossberg, N. Markuzon, J. Reynolds, and D. Rosen. Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps. *IEEE Trans. on Neural Networks*, 3(5):698–713, 1992.

[8] A. Garga and N. Bose. A Neural Network Approach to the Construction of Delaunay Tesselation of Poinds in R^d. *IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications*, 41:611–613, 1994.

[9] P. Green and R. Sibson. Computing Dirichlet tesselations in the plane. *The Computer Journal*, 4:778–787, 1978.

[10] L. Guibas, D. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, 9:534–560, 1992.

[11] T. Kohonen. The self-organization map. *Proceedings of the IEEE*, 78:1464–1480, 1990.

[12] A. Likas, K. Blekas, and A. Stafylopatis. Application of the Fuzzy Min-Max Neural Network Classifier to Problems with Continuous and Discrete Attributes. In *Proc. of IEEE Workshop on Neural Networks for Signal Processing (NNSP '94)*, pages 163–170, 1994.

[13] D. Murphy. Nearest Neighbor Pattern Classification Perceptrons. *Proceedings of the IEEE*, 78:1595–1598, 1990.

[14] F. Preparata and M. Shamos. *Computational Geometry - An Introduction*. New York: Springer-Verlag, 1985.

[15] B. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[16] P. Simpson. Fuzzy Min-Max Neural Networks-Part 1: Classification. *IEEE Trans. on Neural Networks*, 3(5):776–786, 1992.

[17] L. Tsoukalas and R. Uhrig. *Fuzzy and Neural Approaches in Engineering*. John Wiley & Sons, New York, NY, 1996.