A Fuzzy Neural Network Approach Based on Dirichlet Tesselations for Nearest Neighbor Classification of Patterns

K. Blekas, A. Likas and A. Stafylopatis
Department of Electrical and Computer Engineering
National Technical University of Athens
157 73 Zographou, Athens, Greece
Tel.: +301 7485056, Fax: +301 7784578
e-mail: andreas@theseas.ntua.gr

Abstract

A neural network classifier using fuzzy set representation of pattern classes is presented. Network construction and learning is performed incrementally in a single pass by building an aggregate of space-filling regions that constitutes a simplified variant of the construction known as Dirichlet tesselation (or Voronoi diagram). Each region is delimited by a set of hyperplanes and is endowed by a fuzzy membership function that forms the basis of learning and recall. Experimental results concerning difficult recognition problems show that the proposed approach is very successful in applying fuzzy sets to pattern classification.

1 Introduction

Several models have been developed during the last years in an attempt to combine fuzzy systems and neural networks. Some of them focus on applying this synergistic combination to building efficient pattern classifiers [5, 7, 9], as the application of fuzzy sets to pattern classification has been considered for many years.

The fuzzy neural network presented here is an example of neural network classifier that builds decision boundaries by creating subsets of the pattern space. The creation of fuzzy subsets is based on the partition of the n-dimensional space in a way that constitutes a direct adaptation of the notion of Dirichlet tesselations, also known as Voronoi diagrams or Thiessen polygons [1].

A Dirichlet tesselation of a set S of points (called *sites*) is a partition of the n-dimensional space into convex polytopes. Each polytope which is also called 'cell' or 'tile' belongs to one site of the set S and contains all points of the space for which this site is the closest, or the one with the dominant influence. Each cell is defined with respect to an arrangement of halfspaces as the intersection of a finite number of hypeplanes, which are the perpendicular bisectors of the segments joining pairs of sites. ¿From a given set of n-dimensional points classical Dirichlet tesselation can be constructed by obtaining the convex hull of these points [1] or by incremental insertion of the regions of these sites [3, 4]. Dirichlet tesselations express the proximity information of a set of given points in a very explicit and computationally useful manner that makes it applicable in many diverse areas among which are biology, visual perception, crystallography and archeology.

The application of Dirichlet tesselations to the design of neural networks has been considered recently. In [8] two neural network construction algorithms for pattern classification are proposed that rely directly or indirectly on the Dirichlet tesselation of the space based on the given training patterns. An efficient adaptation of the above algorithms is presented in [6], whereas a systematic procedure for designing neural networks following the same principle is formulated in [2]. In this paper we develop an analogous construction approach which incorporates the idea of fuzzy set classes by defining fuzzy decision boundaries for the regions of the tesselation. The proposed scheme allows for efficient on-line supervised learning using appropriately defined fuzzy membership functions during both learning and recall.

A description of the proposed fuzzy classification network is provided in the next section, while the network construction algorithm is presented in Section 3. Section 4 concerns experimental results from the application of the approach to difficult classification problems. Section 5 briefly describes the extension of the model to the case of both continuous and discrete attributes, and finally Section 6 summarizes the main conclusions.

2 Fuzzy Set Classes and Network Topology

Consider a classification problem with n continuous attributes, such that the n-dimensional patterns belong to p distinct classes. By means of the proposed construction scheme, we shall define a set of *regions* filling the feature space such that each region is associated with exactly one from the pattern classes. A properly computed fuzzy membership function (taking values in $[0, 1]$) indicates the degree to which a pattern is contained within each of the regions. During operation, the region with the maximum membership value is selected and the class associated with the winning region is considered as the desicion of the network.

Learning in the fuzzy classification network consists of creating and adjusting regions and associating a class label to each of them. Each region is characterized by a point, which will be called the site of the region, and can be expressed as the intersection of a finite number of closed half-spaces defined by hyperplanes that separate regions of different classes. Regions corresponding to the same class can be overlapping. In general, not all training patterns constitute sites of regions. Following the principle of Dirichlet tesselations, the points of a region are closer to the site of the region than to all other sites belonging to different classes. This feature constitutes a relaxation with respect to the strict definition

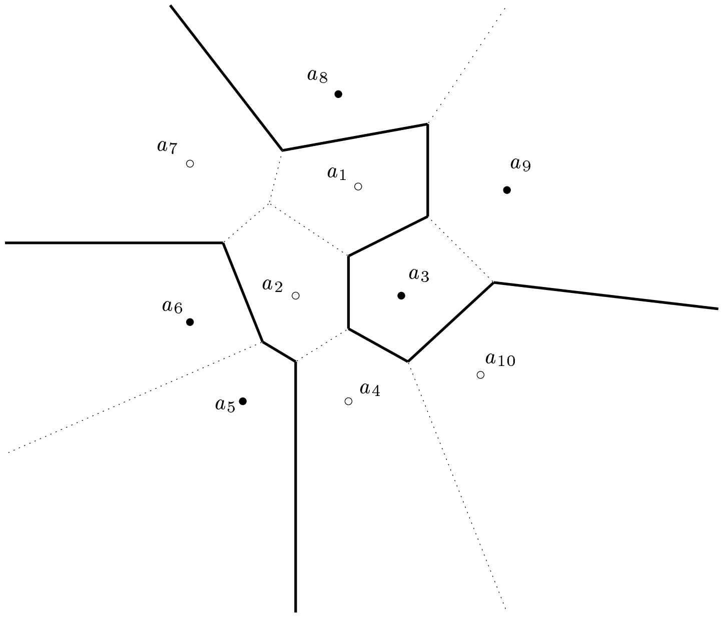

Figure 1: A partition of the plane

of Dirichlet tesselations and implies a construction scheme that prescribes no separating hyperplane between regions of the same class. Figure 1 represents such a convex construction on the 2-dimensional space, based on the Dirichlet tesselation principle, for a set of 10 input points with three seperated classes (dotted lines would be present in a classical Voronoi diagram).

When an input pattern $a = (a_1, \ldots, a_n)$ is presented to the network during operation, the corresponding membership function for each region is computed. The membership function $\mu_i(a)$ for the ith 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 is situated the pattern from all the hyperplanes which define the region. When the pattern a is in the interior of the region and far from the hyperplanes then $b_i(a)$ approaches 1, the value 1 meaning that the point is very close to the site of the region. When the pattern falls outside the region then the membership value approaches zero, the value 0 meaning that this point is close to some other site. A function following the above guidelines is the average value of the normalized vertical distances x_h of the pattern from all hyperplanes h supporting the region. Each distance x_h is normalized with respect to the distance l_h of the site of the region from hyperplane h.

Consider the function $\text{sign}_h(a)$ which describes on which side of the hyperplane h lies the pattern a. If it lies in the positive half space h^+ we have $\text{sign}_h(a) = 1$, else if it lies in the negative half space h^- then $\text{sign}_h(a) = -1$. Also consider the quantities v_{ih} which take the values 1 or -1 depending on whether the site i is situated in the positive or negative half-space defined by the hyperplane h, respectively.

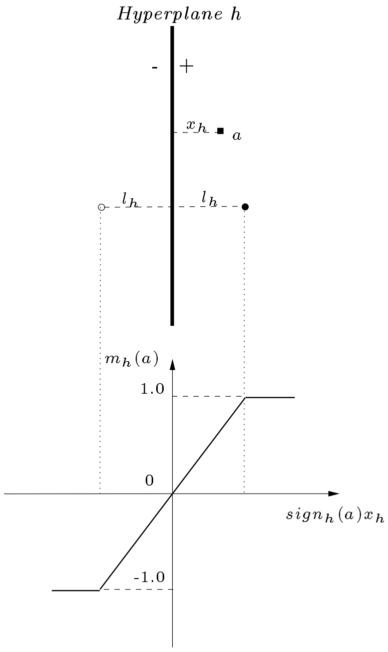

Figure 2: Fuzzy decision boundaries

The membership function taking values in $[0, 1]$ can be computed as follows:

$$\mu_i(a) = \frac{1}{2|H_i|} \sum_{h \in H_i} v_{ih} m_h(a) + \frac{1}{2} \tag{1}$$

where H_i is the set of hyperplanes defining the region i (having cardinality $|H_i|$) and m_h has the following form (Figure 2):

$$m_h(a) = \begin{cases} 1 & \text{if } x_h > l_h \text{ and } \mathrm{sign}_h(a) = 1 \\ -1 & \text{if } x_h > l_h \text{ and } \mathrm{sign}_h(a) = -1 \\ \mathrm{sign}_h(a) x_h / l_h & \text{otherwise} \end{cases} \tag{2}$$

Other choices can be made for the computation of the membership functions, e.g. the form adopted in [5].

The fuzzy classifier can be implemented as a neural network that exploits the fuzzy set structure and allows for efficient implementation. Figure 3 illustrates the neural network that implements this approach. It consists of three layers such that connections exist between successive layers. The number of nodes in the first layer is equal to the number q of hyperplanes that define regions. Each first layer node computes the value of the function m_h for every input pattern using equation (2). The second layer contains as many nodes as the number r of regions. The output of each node of this layer represents the membership value of the pattern for the corresponding region as computed in equation (1). The connections between nodes of the first and second layer associate regions with their supporting hyperplanes and assume the values v_{ih} defined above. The last layer embodies

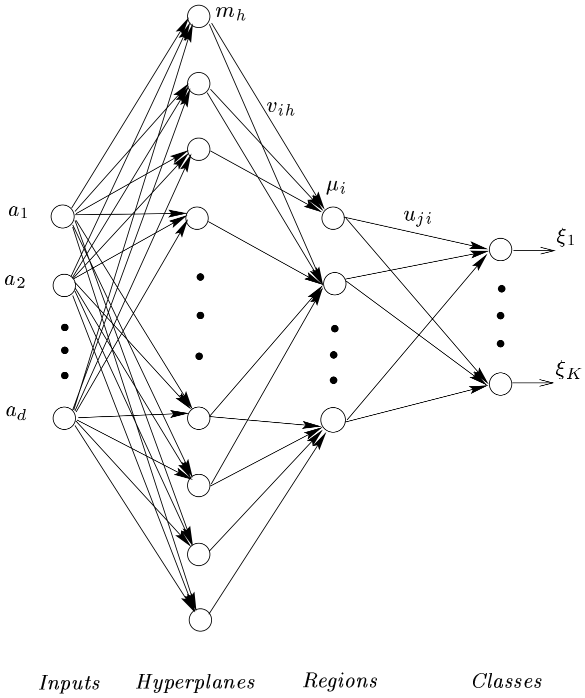

Figure 3: The Fuzzy Neural Network Classifier

nodes which correspond to the set of p classes. The connections u_{ji} between the second and third layer take binary values, such that $u_{ji} = 1$ if i is a region of class j and $u_{ji} = 0$ otherwise. Each node of the third layer computes the degree to which the input pattern fits within class j. The function that performs this computation is the fuzzy union of the appropriate region fuzzy set values. This operation is defined for each of the p classes as

$$\xi_j = \max_{i=1}^{r}[u_{ji} b_i(a)] \tag{3}$$

3 Learning and Construction

Consider a set A of training patterns. The learning algorithm creates a division of the feature space by appropriately constructing regions. Each region is defined by hyperplanes that are successively created to separate neighboring regions of different classes. Implementation of the below scheme requires the definition of the appropriate data structures for holding all the information necessary during the construction.

At an initialization step, the first two training patterns considered (which should be of different classes) become the sites of the first regions which are originally separated by a hyperplane (the perpendicular bisector of the segment joining the two sites). These regions will be restricted in the sequel as new sites are created.

During learning, each training pattern a_k is presented once and the following general step is performed.

- First we compute the values of the membership functions $\mu_i(a_k)$, as defined previously, for all existing regions i. Then we find the regions whose membership values exceed a given threshold value θ ($0 \leq \theta \leq 1$), which is generally taken high (typically, greater than 0.7).

- If all the regions meeting the above criterion belong to the same class as the presented pattern a_k, no further action is taken.

- If one or more of the selected regions belong to classes different than that of pattern a_k, then the latter becomes a new site and its region is constructed by drawing bisecting hyperplanes between this site and its neighboring sites of different classes. No hyperplane is created between the new site and sites belonging to the same class, thus allowing for overlapping. The neighboring regions of the new region are successively determined by applying a simple adaptation of standard techniques used in the creation of Dirichlet tesselations by incremental insertion of sites [3, 4].

- The new site acquires its region by winning territory from the regions of its neighbors (belonging to different classes). As some of the already presented (non-site) patterns may be contained in the affected regions, it should be checked whether such patterns are now included in the newly created region. Thus, these patterns are successively examined and if they are contained in the new region they create their own new regions by winning territory from the latter, following the procedure applied in the previous step for a_k. Obviously, this construction of new regions need not take place for all such points, since several of them may be covered by each newly created region of the correct class.

4 Experimental Results

We have studied the proposed fuzzy neural network classifier on a variety of difficult classification problems. We have tried to select databases whose instances are defined on a high-dimensional space so that the applicability of the Dirichlet tesselation approach on such problems could be evaluated. In addition, some of the data sets were noisy containing hard examples so as to illustrate the operation and performance of the fuzzy neural network classifier. To evaluate the effectiveness of our model we have mainly compared it with the fuzzy min-max classifier [9].

The first data set is the Johns Hopkins University ionosphere database which is a collection of radar data. The ionosphere data set consisted of 351 feature vectors described by 34 continuous valued attributes with two decision classes (either show evidence of some type of structure in the ionosphere or not). The data set was divided into a training set of 200 examples that were used to adjust the network hypeplanes and convex polytopes, while the remaining 151 examples were applied to the constructed network structure to estimate the performance of the proposed fuzzy neural classifier. In all of our experiments we trained the network for certain θ values and then computed the percentage of correct classification over the test set. Best results were found for $\theta = 0.75$. For this parameter value the network consisted of 127 cells and the success rate was 97%. On the other hand, using the same data set to train a fuzzy min-max neural network classifier several experiments were conducted for different values of θ. The best classification rate obtained was 95.5%

The second data set we used to train and test our fuzzy neural classifier was the Fisher's Iris data. Iris data is a collection of 150 four-dimensional featute 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. After a series of experiments using different values of the parameter θ we found the best classification rate 97.3% for $\theta = 0.75$ in which we obtained 22 polygon cells. For the fuzzy min-max classifier the best classification rate for the same data set was exactly the same [9].

We have also used the James Cook University Thyroid gland database in our model. Thyroid database is a collection of 215 feature vectors consisting of 5 continuous attributes, such that the vectors belong to three classes. Any of these three decision classes defines a prediction of a patient's thyroid to the class of euthyroidism, hypothyroidism or hyperthyroidism. The database is divided into 150 instances of first class, 35 instances of second and 30 of last class. We used a training set of size 100 while the remaining data set (size 115) was used as testing set. Best performance was obtained for $\theta = 0.8$ (34 polygon cells) with classification rate 94%. Training and testing the fuzzy min-max classifier network with the same data sets we were able to achieve a success rate of 90.5% using parameter value $\theta = 0.082$ (60 cells).

It must be noted that in all the experiments the choice of the value of the parameter θ was not very critical with respect to the success rate as was the case with the fuzzy min-max neural network. There were intervals of θ values where the rate remained the same and only the number of the hyperplanes and the convex polygons being created were different. Besides, while the value of θ was increasing the network structure (hypeplanes and cells) was reduced, and so we were choosing the maximum θ value of such intervals so as to achieve the least network architecture with the best overall success rate.

5 Treating Discrete Attributes

The model of fuzzy neural network based of Dirichlet Tesselations considers as basic assumption that all attributes take continuous values. Thus, we are able to map the pattern space corresponding to each class to a number of regions (convex polygons) by creating perpedincular bisectors (hyperplanes) between sites of different classes. Nevertheless, when the data set consists of both continuous and discrete attributes we cannot treat the discrete features in the same way, and so it is necessary to find another mode of operation.

Suppose that $\mathcal{D}$, $n_D = |\mathcal{D}|$ and $\mathcal{C}$, $n_C = |\mathcal{C}|$ denote the set and the number of the discrete and the continuous attributes respectively. Let also D^j be the domain of each discrete attribute $j \in \mathcal{D}$. A n-dimensional pattern $a = (a_1, a_2, \ldots, a_n)$ having both types of attributes, consists of continuous features a_j for $j \in \mathcal{C}$ and discrete $a_j \in D^j$ for $j \in \mathcal{D}$. Each polygon i is described by providing the proper hyperplanes with respect to the continuous attributes and moreover a set of attribute values $D_{ij} \subseteq D^j$ for discrete attributes $j \in \mathcal{D}$. It is obvious that the sets D_{ij} must be crisp, i.e., an element either belongs to a set (membership value is 1) or not (membership value is 0). Including the above analysis to the computation of the membership function of a pattern a to a polygon i, equation (1) takes the following form:

$$\mu_i(a) = \frac{1}{4|H_i|} \sum_{h \in H_i} v_{ih} m_h(a^*) + \frac{1}{4} + \frac{1}{2n_D} \sum_{j \in \mathcal{D}} m_{D_{ij}}(a_j) \tag{4}$$

where a^* denotes the subvector of a containing only continuous attributes and $m_S(x)$ is the membership function corresponding to the crisp set S. It must be noted that if a new input pattern a_k is contained in a cell i of the same class, i.e., no creation of new cell takes place, the crisp sets D_{ij} are adjusted as follows: $D_{ij}^{new} = D_{ij}^{old} \cup a_{kj}$.

6 Conclusions

We have introduced a new model of fuzzy neural network classifier by representing fuzzy sets through a suitable partition of the solution space into a number of convex regions following the principle of Dirichlet tesselations. This type of network has the advantage of fast one-shot training and is very efficient for hard pattern classification problems as indicated by the experiments. Further research is focused on the introduction of a learning component for adaptively determining good parameter values.

References

[1] F. Aurenhammer, "Voronoi Diagrams — A Survey of a Fundamental Geometric Data Structure," ACM Computing Surveys, Vol. 23, No. 3, pp. 345–405, Sept. 1991.

[2] N.K. Bose and A.K. Garga, "Neural Network Design Using Voronoi Diagrams," IEEE Trans. on Neural Networks, Vol. 4, No. 5, pp. 778–787, Sept. 1993.

[3] A. Bowyer, "Computing Dirichlet Tesselations," The Computer Journal, Vol. 24, No. 2, pp. 168–173, 1978.

[4] P.J. Green and R. Sibson, "Computing Dirichlet Tesselations in the Plane," The Computer Journal, Vol. 21, No. 2, pp. 168–173, 1978.

[5] J. Keller and D. Hunt, "Incorporating Fuzzy Membership Functions into the Perceptron Algorithm," IEEE Trans. on Patt. Anal. and Mach. Intell., Vol. 7, pp. 693–699, 1985.

[6] K. Koutroumbas and N. Kalouptsidis, "Nearest Neighbor Pattern Classification Neural Networks," Proc. World Congress on Computational Intelligence, pp. 2911–2915, Orlando, Florida, July 1994.

[7] A. Likas, K. Blekas and A. Stafylopatis, "Application of the Fuzzy Min-max Neural Network Classifier to Problems with Continuous and Discrete Attributes," Proc. NNSP 94, pp. 163–170, Ermioni, Greece, Sept. 1994.

[8] O.J. Murphy, "Nearest Neighbor Pattern Classification Perceptrons," Proc. IEEE, Vol. 78, No. 10, pp. 1595–1598, Oct. 1990.

[9] P. K. Simpson, "Fuzzy Min-Max Neural Networks-Part 1: Classification," IEEE Trans. on Neural Networks, Vol. 3, No. 5, pp. 776–786, Sept. 1992.