PERGAMON

Pattern Recognition 34 (2001) 1469–1482

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Fuzzy convex set-based pattern classification for analysis of mammographic microcalcifications

Wojciech M. Grohman, Atam P. Dhawan*

New Jersey Institute of Technology, Chair, Elect. & Comp. Engineering, University Heights, Newark, NJ 01702, USA

Received 4 October 1999; accepted 2 May 2000

Abstract

There are many different criteria for the comparative analysis of pattern classifiers. They include generalization ability, computational complexity and understanding of the feature space. In some applications such as the medical diagnostic systems it is crucial to use reliable tools, whose behavior is always predictable, so that the risk of misdiagnosis is minimized. In such applications the use of the popular feedforward backpropagation (BP) neural network algorithm can be seen as questionable. This is because it is not inherent for the backpropagation method to analyze the problem's feature space during training, which can sometimes result in inadequate decision surfaces. A novel convex-set-based neuro-fuzzy algorithm for classification of difficult-to-diagnose instances of breast cancer is described in this paper. With its structural approach to feature space the new method offers rational advantages over the backpropagation algorithm. The classification performance, computational and structural efficiencies are analyzed and compared with that of the BP network. A 20-dimensional set of "difficult-to-diagnose" mammographic microcalcifications was used to evaluate the neuro-fuzzy pattern classifier (NFPC) and the BP methods. In order to evaluate the learning ability of both methods, the relative size of training sets was varied from 40 to 90%. The comparative results obtained using receiver operating characteristic (ROC) analysis show that the ability of the convex-set-based method to infer knowledge was better than that of backpropagation in all of the tests performed, making it more suitable for use in real diagnostic systems.

Keywords: Neural networks; Pattern classification; Convex sets; Breast cancer; Mammography

1. Introduction

The pattern recognition systems such as BP, radial basis function (RBF) networks or of *k*-nearest-neighbor (KNN), which use crisp decision surfaces often suffer from low immunity to noise in the training patterns. To address this issue and effectively improve their system's performance some researchers have introduced a combination of pattern recognition neural networks and various concepts from the fuzzy set theory. The example ideas include intended improvement of fuzzy decision systems [1–8], incorporation of existing knowledge into neural network architectures [6,9,10], improvement of the system's performance on noisy input data [11], allowing fuzzy input to the neural network [1–6,12], and others [8,13,14]. In this work we introduce a novel pattern recognition method NFPC, that incorporates the fuzziness into the decision surfaces to further improve the classification performance. The new method's performance was compared with that of a leading implementation of the backpropagation algorithm. The results of this comparison are reported in this paper.

The main motivation behind the use of the new algorithm is its structural approach to pattern recognition. Unlike the popular and widely used [1–5] backpropagation neural network which uses no information about pattern points position in the feature space in any explicit way, the NFPC training method first identifies clusters within the training data, and then constructs the actual

* Corresponding author.
E-mail address: dhawan@adm.njit.edu (A.P. Dhawan).

0031-3203/01/$20.00
PII: S0031-3203(00)00085-6

network architecture. This feature significantly increases the robustness of the new approach, as the danger of falling into local minima of the error function is minimized, and pattern classification is based on the real data clusters. With the only assumption being that the identified data clusters are convex, the presented method retains the greatest advantage of backpropagation and other feedforward neural networks, i.e. their flexibility. The assumption made about the convexity of identified clusters is not a strict requirement, since even concave clusters can be represented as a union of a finite number of convex ones. The input to the classifier is crisp and the output is based on the values of fuzzy membership functions that partition the input space. This allows for higher immunity to noise in the training patterns and is designed to improve the classifier's performance. An additional method's benefit is its self-determined network structure, which eliminates the need for heuristical guessing of the number of neurons and layers common to all backpropagation networks.

At the present time mammography associated with clinical breast examination is the only reliable mass-screening method for breast cancer detection [15–17]. In most cases the breast cancer diagnosis based on mammographic images is a routine task for an experienced physician, however there are instances where the prognosis cannot be easily made. Since there are over 50 million women over the age of 40 at risk of breast cancer, and approximately 144,000 new breast cancers per year are expected to be diagnosed in the United States alone [16], the absolute number of difficult-to-diagnose cases is quite significant.

It has been empirically recognized that certain kinds of microcalcifications are associated with a high probability of cancer [15]. However, the analysis of the mammographic images is usually difficult and the results are ambiguous. The difficulty in the interpretation of microcalcifications leads to a significant increase in the number of biopsy examinations. At present, only one out of five biopsies recommended on the basis of the microcalcification sign yields positive results. A reduction in the false-positive call rate will not only reduce health care costs by avoiding nonproductive biopsies, it will also provide women better patient care. Reduction in the false-positive rate must be achieved while maintaining sensitivity [18]. The recent introduction of pattern recognition methods has made the computerized image analysis possible. Such analysis, designed to help decision making for biopsy recommendation and diagnosis of breast cancer, shall be of significant value to improve the true-positive rate of breast cancer detection. This can lead to the decrease of the false-positive rate, thus effectively reducing the health care costs.

Section 2 describes the architecture and construction process of the NFPC. Section 3 presents the experimental results on the breast cancer data [18] and it is followed by discussion and conclusion in Sections 4 and 5, respectively. The basic concepts from convex set theory are presented in the appendix.

2. Method description

2.1. Theoretical background and derivation of NFPC construction method

The main idea behind the described method comes from the basic properties of feedforward artificial neural networks. Any layer of a feedforward network performs partitioning of its d-dimensional input feature space into a specific number of subspaces that are always convex and which number can be estimated [19]. This is regardless of the training algorithm or the neural function $f(\varphi)$ used. The only requirement is that the connection weights w_i are linear, i.e., that the relationship between the layer's inputs x_i and the post-synaptic signal φ processed by the neural function is of a form

$$\varphi = \sum_{i=1}^{d} x_i w_i + w_0. \tag{1}$$

Most popular feedforward networks, including radial basis function and backpropagation, satisfy this requirement. The corresponding, general neuron architecture is shown in Fig. 1.

For $\varphi = 0$ (or any other constant), the synaptic equation (1) represents a $(d-1)$-dimensional hyperplane H in the d-dimensional input space separating two regions defined by the connection weights w_i [21]:

$$(H : \varphi = 0) \Rightarrow \left(H: \sum_{i=1}^{d} x_i w_i + w_0 = 0 \right). \tag{2}$$

Each network layer comprises many such hyperplanes, which by intersecting with one another create a finite number of the aforementioned convex subspaces. Therefore, there is a direct relationship between the connection weight values and the obtained d-dimensional convex subspaces. The process of network training could be seen

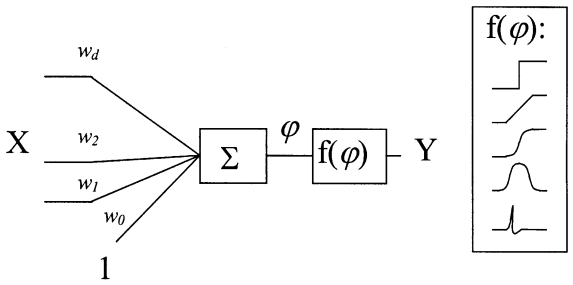

Fig. 1. Neuron model with linear synapses.

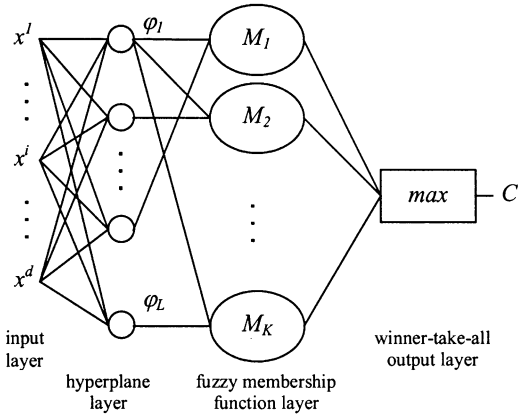

Fig. 2. Architecture of the NFPC.

as the attempt at finding an optimal dichotomy of the input space into these convex regions. Moreover, the relationship goes both ways, i.e. proceeding in the reverse order, one might say that finding the optimal dichotomy of the input space into convex subspaces is equivalent to network training.

As a result of the above reasoning, the following procedure is proposed: in the feature space one separates the given number of training points from all categories. It is achieved by dividing them into homogeneous (containing only points from one category), non-overlapping, closed convex subsets, and then placing separating hyperplanes between neighboring subsets from different categories. This completes the design of a network hyperplane layer. At that point, there are a number of possible procedures of utilizing the created layer [20]. In any case, since the hyperplane separation of the obtained subsets results in creation of the homogenous convex regions, the consecutive network layer is to determine to which region a given input pattern belongs. In our approach a fuzzy membership M_f function is devised for each created convex subset ($f = 1, 2, \ldots, K$). The classification decision is made by the output layer based on the "winner-take-all" principle. The resulting category C is the convex set category with the highest value of membership function for the input pattern. The structure of the proposed neuro-fuzzy pattern classifier is shown in Fig. 2.

Summarizing, our neuro-fuzzy pattern classifier design method includes three stages: convex set creation, hyperplane placement (hyperplane layer creation), and construction of the fuzzy membership function for each convex set (generation of the fuzzy membership function layer).

(1) *Convex set creation*: There are two requirements for the convex sets: they have to be homogeneous and non overlapping. To satisfy the first condition, one needs to devise a method of finding one-category points within another category's hull. Thus, two problems can be defined: 1 — how to find whether the point P lies inside of a convex hull CH of points; 2 — how to find out if two convex hulls of points are overlapping. The second problem is more difficult to examine because hulls can be overlapping over a common (empty) space that contains no points from either category. This problem can be defined as a generalization of the first one [20], and the first condition can be seen as a special case of the second requirement, when one of the convex sets is a single point. An interesting discussion on the first problem and its complexity can be found in Refs. [22,23]. The complexity of the second problem is far greater. For more detailed analysis of the problem, see Ref. [20].

In real-world situations, when training samples are not completely noise-free it would not be advisable to insist on high accuracy of solutions to problems 1 and 2. In such a case a compromise between computational efficiency and accuracy should be reached. With this in mind a new algorithm for solving problem 1 is proposed below. It is based on another property of convex sets, described by the separation theorem [24], which states that for two closed non-overlapping convex sets S_1 and S_2 there always exists a hyperplane that separates the two sets — separating hyperplane.

Algorithm A1. Checking if the point P lies inside of a convex hull CH

1. Put P in Origin.
2. Normalize points of CH (the vectors $V = (v_1, v_2, \ldots, v_n)$ from the origin).
3. Find min and max vector coordinates in each dimension.
4. Find set E of all vectors V that have at least one extreme coordinate.
5. Take their mean and use it as projection vector ϕ:

$$\phi = (\bar{v}_i | \forall v_i \in E).$$

6. Set a maximum number of allowed iterations (usually $= 2n$).
7. Find a set $U = (u_1, u_2, \ldots, u_m)$ (where $m \leqslant n$) of all points in CH that have negative projection on ϕ.
8. If U is empty (P is outside of CH) exit, else proceed to 9.
9. Compute coefficient ψ:

$$\psi = \phi^{\mathrm{T}} \bar{U},$$

$$\bar{U} = \frac{1}{m} \sum_{i=1}^{m} ui.$$

10. Calculate correction vector $\delta\phi$ by computing all of its k-dimensional components $\delta\phi^k$:

$$\begin{pmatrix} \bar{U}^k \neq 0 \Rightarrow \delta\phi^k = \dfrac{\psi}{\bar{U}^k} d \\ \\ \bar{U}^k = 0 \Rightarrow \delta\phi^k = \dfrac{\psi}{d} \end{pmatrix}, \quad k = 1, 2, \ldots, d$$

where d is the data's dimension.
11. Update ϕ: $\phi = \phi - \eta \cdot \delta\phi$, where $\eta > 1$ is a training parameter.
12. if iteration limit exceeded exit (assume P inside of CH), otherwise go to 7.

The value of the training parameter η should be close to 1, so even the points lying outside but close to the hull can be found. Heuristically, it has been found that the values of α should fall in the range $1.0001 < \eta < 1.01$. They are, however, dependent on the precision of the training data and should be adjusted accordingly.

The principal idea behind the algorithm is to find the hyperplane (defined by its orthogonal vector ϕ) separating P and CH. If such a hyperplane is found within a certain amount of iterations, the point is definitely outside of CH. If the hyperplane has not been found, it is assumed that P is inside.

Now, having found the solution to problem 1, we can propose a method for constructing convex subsets:

Algorithm A2. Convex subset creation

1. Select one category. Consider the set of all its training points. This is a positive set of samples. The training points from all the remaining categories constitute a negative set. Both sets are in d-dimensional linear space L. Mark all positive points as "not yet taken" and order them in a specific way. For example, choose an arbitrary starting point in the input space and order all positive points according to their Euclidean distance from that point. Use an index array Λ to store the order.
2. Build the convex subsets.
   Initialize current subset S by assigning to it the first point in Λ. Loop over ordered positive category points (in Λ) until there are no more points remaining. Consider only points that have not yet been "taken":
   (a) Add the current point P to the subset S.
   (b) Loop over points from negative category. Consider only negative points that are closer than P to the middle of the current subset. Using Algorithm A, look for at least one negative point inside of conv S. If there is one, disregard the latest addition to S. Otherwise mark the current point P as "taken".
   (c) Update Λ. Reorder the "not yet taken" positive category points according to their distance from the mean of points in the current subset.
3. If all points in the category have been assigned to a subset proceed to step 4, otherwise go back to step 2 and create the next convex subset. The starting point is the first "not yet taken" point in the list.
4. Check if all categories have been divided into convex subsets. If not, go back to step 1 and create subsets of the next category.

In the step 2b it is not always necessary to use Algorithm A1 for checking the presence of every single negative point within the current convex subset. Once a separating hyperplane is found for one negative point it should be used to eliminate all other negative points that lie on the opposite side of the hyperplane than the convex subset, from the checklist. Thus, both the presented algorithms should be used together in order to save computations. Using procedures A1 and A2 does not guarantee that the constructed convex subsets are not overlapping, since problem 2 is essentially not addressed. It is of no significance when the subsets are from the same category. However, when they are not, this could result in linear non-separability of the neighboring subsets. This might seem as a drawback of the proposed solution since the overall accuracy seems to have been compromised for the benefit of computational efficiency. However, the results of performed test show that this compromise is acceptable, since the performance of the NFPC was equal, or better than that of the backpropagation network classifier — see Section 3. Not directly addressing problem 2 does not mean that the constructed subsets are always overlapping. Contrarily, the more representative the training set (i.e., greater number of training samples), the smaller probability of the overlap, as the likelihood of finding a common empty space decreases. In reality, as the obtained results for the tests performed show, see Section 3, this approximation yields acceptable results that are comparable to and often better than that of other methods.

In Ref. [23] the authors proposed a different method for solving problem 1 — separating hyperplane detection (SHD) algorithm. As opposed to the approximate procedure A1, SHD always provides a definite answer. However, as the results in Section 3 show, its computational complexity is always higher. This is because the separating hyperplane is not found, only detected, so no negative points can be eliminated from the checklist in step 2b of Algorithm A2.

(2) *Initial subset point selection*: The presented algorithm requires initialization in the form of starting points for convex subsets from each category (step 1 of Algorithm A2). There are many possible ways of finding these starting points. In the simplest case they may be chosen randomly or by taking the mean of all category points' coordinates. In the conducted experiments these starting points for each category were obtained by resolution coarsing, which was performed to place the starting point

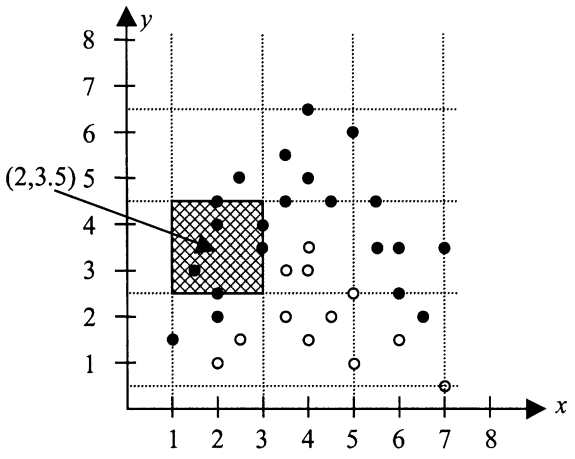

Fig. 3. Resolution coarsing.

in an area with the greatest concentration of that particular category's points. The concept is explained by the example in Fig. 3, where the starting point for the black category is determined.

Every dimension was divided into a specific number of bins, or intervals. These intervals are created first by taking the distance, along that particular dimension, from the minimum coordinate value of all points to the maximum coordinate value, and then dividing this distance by a predefined number, which is 3 in our example. All point coordinates that fall within one of the bins are changed to the coordinate value of the middle point for that particular bin. Thus, in our example in Fig. 3, all points with x coordinates from 1 to 3 are assigned x coordinate value of 2, all points from 3 to 5 are assigned 4, and all points from 5 to 7 are assigned 6. Correspondingly, points with y coordinates from 0.5 to 2.5 are assigned $1.5y$ coordinate value, etc. Next, a number of occurrences for each modified point is counted. The point that is repeated most is chosen as a starting point for the category. In our example it would be (2, 3.5).

(3) *Placing hyperplanes — hyperplane layer creation*: Once the convex subsets have been found, it is assumed that they are not overlapping, so that only one hyperplane is needed to separate two neighboring subsets. The program loops over subsets from all categories and places a hyperplane between two sets from different categories that have not yet been separated by existing hyperplanes. Thus, a number of hyperplanes can vary depending on the training set. Several algorithms can be used to place a separating hyperplane, however it has been proven [25] that backpropagation with batch training performs better than other methods when the two classes are linearly separable. Since we are primarily dealing with linearly separable convex subsets, backpropagation with batch training was used in our implementation. A hyperplane was represented by a single neuron trained to output a positive value (+ 1) for one category and a negative value (− 1) for the other. The NPFC hyperplane layer comprises a set of all hyperplanes needed to fully separate all convex subsets from different categories.

(4) *Fuzzy membership function construction*: The placed hyperplanes define the convex regions trained from the presented samples. These created regions are the bases for constructing fuzzy membership functions, which represent the point's relative membership in a given convex subset, rather than in a category. It means that for a single point the sum of its membership values for different convex clusters is bound from below — it can never be negative — and from above by a total number of convex subsets for all categories. The utilized fuzzy membership function M_f has to be flexible to reflect the true shape of the convex subset with the greatest precision possible. In our case it was defined for each subset $f(f = 1, 2, \ldots, K)$ as follows:

$$M_f(\mathbf{x}) = L_f\sqrt{\prod_{i=1}^{L_f}\theta_i}, \quad \theta_i = \frac{1}{(1 + e^{\lambda_{if}\varphi_i\mathbf{x}})}, \tag{3}$$

where L_f is the number of separating hyperplanes for the subset f, φ_i the ith separating hyperplane function for the subset, in the vector form, x the network's input vector in the augmented form and λ_{if} the steepness (scaling) coefficient for the ith hyperplane in the subset f.

The value of λ_{if} depends on the depth of convex subset f, as projected onto the separating hyperplane H_i (defined by φ_i):

$$\lambda_{if} = \frac{-\log((1-\chi)/\chi)}{\mu_{if}}, \mu_{if} = \frac{1}{n}\sum_{j=1}^{n}\varphi_i\mathbf{x}_j, \tag{4}$$

where n is the number of training points in the covex subset f, φ_i the ith hyperplane equation in the vector form, μ_{if} the depth of the convex subset f, as projected onto ith hyperplane, $\mathbf{x}_j$ the augmented coordinate vector of the jth point in the subset, and χ the center value of the membership function.

Since the sigmoidal function in Eq. (3) is continuous and only reaches the value of 1 at infinity, the resulting maximum value of M_f is less than 1. In practice, the maximum possible value is controlled by the center value χ, which is the goal membership value for a point with the mean projection value onto H_i for the entire subset. In the performed tests χ was set to 0.99.

Other versions of fuzzy membership functions are possible. An alternative approach is represented by two examples shown in Eqs. (5) and (6) below:

$$M_f^*(\mathbf{x}) = L_f\sqrt{\prod_{i=1}^{L_f}\theta_i}, \theta_i = \frac{1}{(1 + e^{\lambda_{if}\varphi_i\mathbf{x}})(1 + e^{-\lambda_{if}(\varphi_i\mathbf{x} + \delta_{if})})}, \tag{5}$$

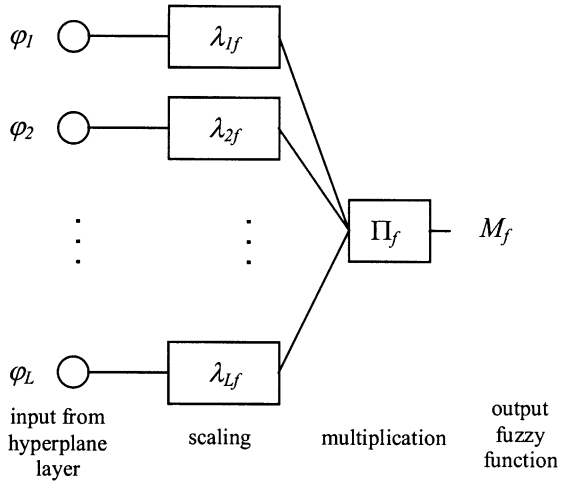

Fig. 4. Fuzzy membership function neuron.

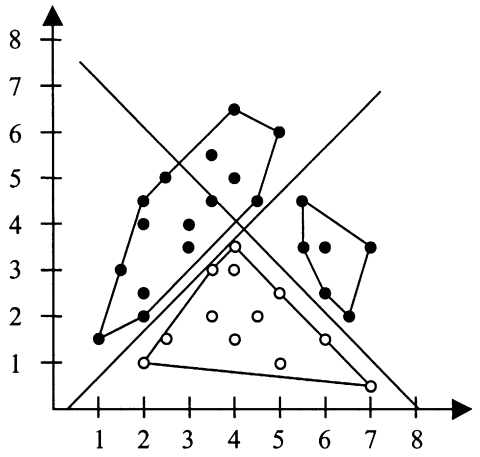

Fig. 5. Convex set-based separation of two categories.

where L_f is the number of separating hyperplanes for the subset f, φ_i the ith separating hyperplane function for the subset, in the vector form, $\mathbf{x}$ the networks input vector in the augmented form, λ_{if} the steepness (scaling) coefficient for the ith hyperplane in the subset f, defined by Eq. (4) and δ_{if} the width of the subset f as projected on the ith hyperplane.

$$M_f^{\wedge}(\mathbf{x}) = L_f\sqrt{\prod_{i=1}^{L_f}\theta_i},$$

$$\theta_i = \frac{1}{\sqrt{2\pi\sigma_{if}}}\exp\left(\frac{-(\varphi_i\mathbf{x}-(\delta_{if}/2))^2}{2\sigma_{if}^2}\right), \tag{6}$$

where L_f is the number of separating hyperplanes for the subset f, φ_i the ith separating hyperplane function for the subset, in the vector form, $\mathbf{x}$ the network's input vector in the augmented form, σ_{if} the fuzziness coefficient for the ith hyperplane in the subset f, and δ_{if} the width of the subset f as projected on the ith hyperplane.

Both of these membership functions were designed to limit the depth of each subset as seen from the separating hyperplane. This was intended to minimize the influence of the resulting open convex regions over the space where no training points were located. The fuzzy membership function with the normal distribution version in Eq. (6) does not preserve the shape of the original convex regions as well as does the sigmoid-based one in Eq. (5). Therefore, only the later one was chosen for evaluation in the experiments. However, as the results show, this implementation did not perform as well as the one from Eq. (4).

The structure of the designed fuzzy membership function neuron is shown in Fig. 4. Scaling and multiplication stages are represented by Eqs. (4) and (3), respectively. The input to the neuron is the hyperplane layer, created as described in the previous section. The neuron's output is the fuzzy membership function M_f for convex subset f. The neuron structure for fuzzy membership functions from Eqs. (5) and (6) is analogous.

(5) *Winner-take-all output*: The output *Out* of the classifier is the category C of the convex set fuzzy membership function M_i that attains the highest value for the specified input pattern x, i.e.:

$$(Out = C|\forall\ 1 \leqslant f \leqslant K, M_f(x) < M_i(x), M_i \in C, \quad f \neq i),$$

where *Out* is the output of the classifier, x the input pattern, K the number of convex sets obtained during training (number of fuzzy function neurons in the fuzzy membership function layer), M_i the highest fuzzy membership function value for the input x, and C the category of convex subset used to construct membership function M_i.

In other words, the output is based on the winner-take-all principle, with the convex set category corresponding to M_i. determining the output.

A decision surface for each category can be determined by the fuzzy union of all of the fuzzy membership functions for the convex subsets belonging to this category. Thus, if the decision surface for a particular category can be defined as:

$$(M_{category}(x) = \max(M_i(x))|\forall i, M_i \in category),$$

where $M_{category}(x)$ is the decision surface for the *category*, and M_i the fuzzy membership functions for convex cluster i.

To illustrate the design process consider a hyperplane placement shown in Fig. 5. The hyperplanes were placed to separate two convex subsets of the black category from the convex subset of the white category.

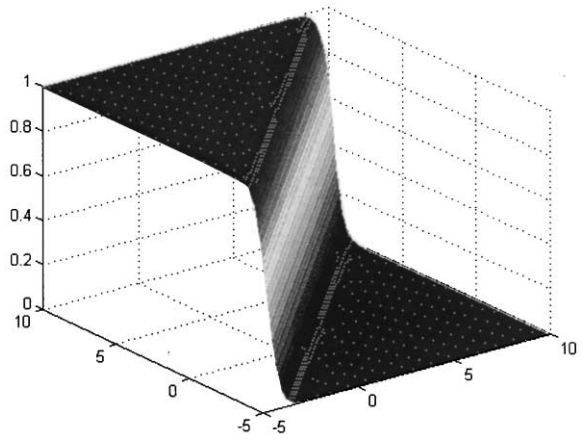

Fig. 6. Fuzzy membership function $M_1(x)$ for the subset $\#1$ of the black category.

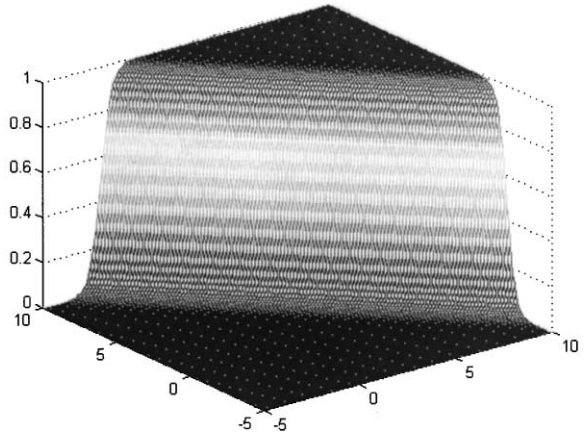

Fig. 7. Fuzzy membership function $M_2(x)$ for the subset $\#2$ of the black category.

Figs. 6 and 7 show the constructed fuzzy membership functions $M_1(x)$ and $M_2(x)$ for black category subsets. Fig. 8 illustrates the membership function $M_3(x)$ for the white category. The resulting decision surface $M_{black}(x)$ for the black category is shown in Fig. 9. The decision surface $M_{white}(x)$ for the white category is identical with $M_3(x)$, since there is only one white points cluster.

Figs. 10 and 11 show the constructed fuzzy membership functions $M_1^*(x)$ and $M_2^*(x)$ for black category subsets. Fig. 12 illustrates the membership function $M_3^*(x)$ for the white category. The resulting decision surface $M_{black}^*(x)$ for the black category is shown in Fig. 13. The decision surface $M_{white}^*(x)$ for the white category is identical with $M_3(x)$, since there is only one white points cluster.

Note that even though constructed convex subsets of training points are always closed, the resulting convex

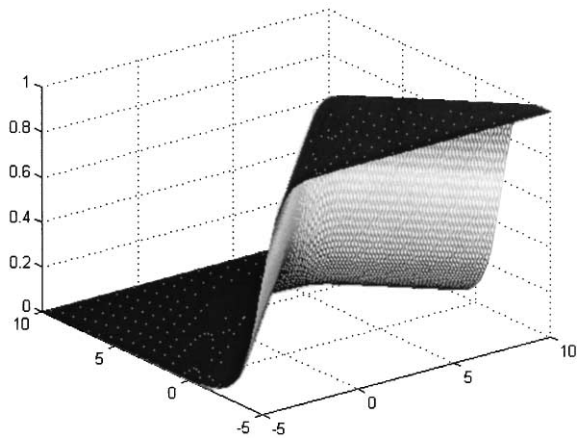

Fig. 8. Fuzzy membership function $M_3(x)$ (decision surface) for the white category membership.

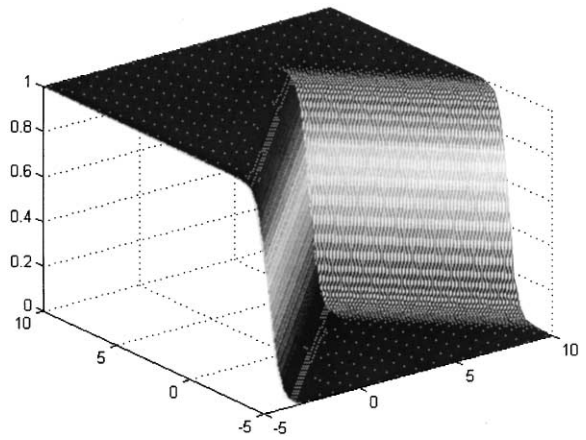

Fig. 9. Resulting decision surface $M_{black}(x)$ for the black category membership function.

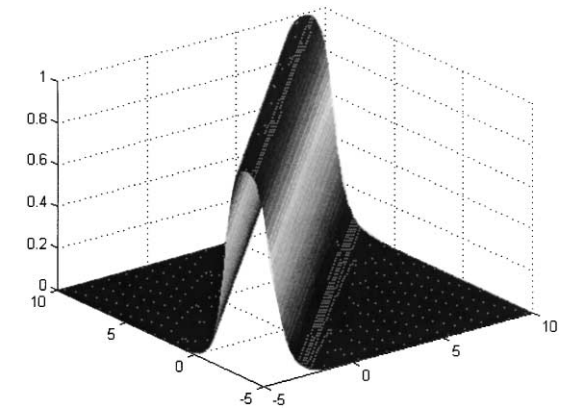

Fig. 10. Fuzzy membership function $M_1^*(x)$ for the subset $\#1$ of the black category.

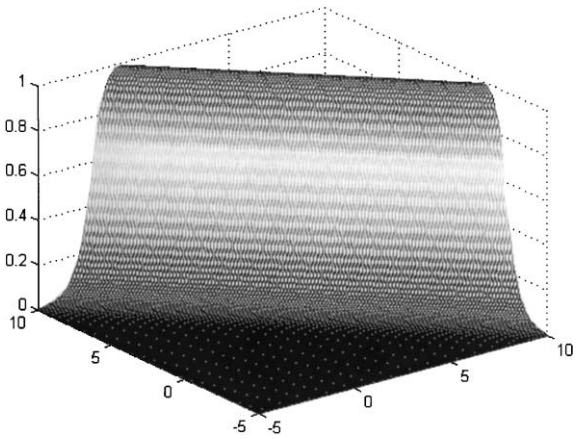

Fig. 11. Fuzzy membership function $M_2^*(x)$ for the subset $\#2$ of the black category.

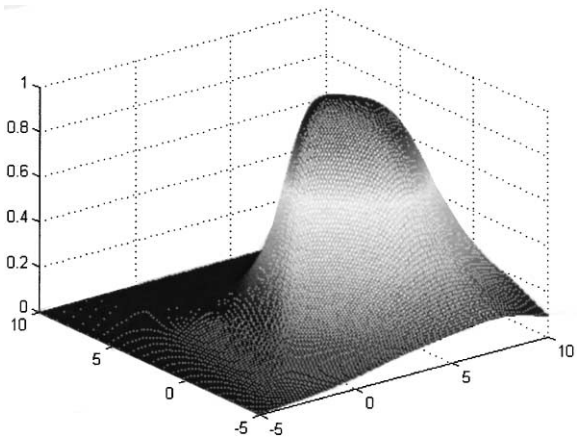

Fig. 12. Fuzzy membership function $M_3^*(x)$ (decision surface) for the white category membership.

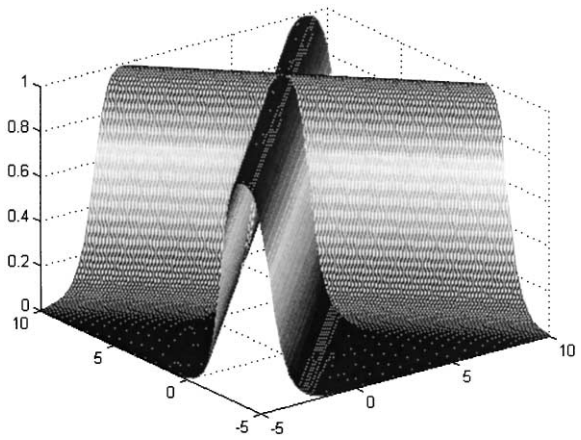

Fig. 13. Resulting decision surface $M_{black}^*(x)$ for the black category membership function.

regions that are the bases for fuzzy membership functions may not be always closed.

3. Results

There are a number of non parametric classification methods available that could be used in the diagnostic system. In their study Dhawan et al. [18] compared the performance of k nearest-neighbor and backpropagation network classifiers. Additionally, an extensive analysis of radial basis function (RBF) classifier was performed in Ref. [26]. The backpropagation network achieved significantly higher classification rates than the KNN and RBF classifiers and thus it was used as a reference method for comparative performance evaluation of the NFPC. Several backpropagation training algorithms available in the Matlab neural network toolbox were analyzed. They included Powell–Beale conjugate gradient BP, Fletcher–Powell conjugate gradient BP, Polak–Ribiere conjugate gradient BP, gradient descent BP, gradient descent with momentum BP, gradient descent with adaptive learning rate BP, Levenberg–Marquardt BP, and others. In all tests the backpropagation network with momentum and the adaptive learning rate, combined with batch training [27] performed best and was consequently used as a standard reference method.

The data set used for evaluation of the methods represents a set of 191 difficult-to-diagnose cases of mammographic microcalcifications selected from the database of more than 18,000 mammograms. There are 128 benign and 63 malignant instances. The selected images were digitized and the gray-level subimages containing the microcalcification areas were extracted and then stretched to the gray-level range of 0–255. These normalized gray-level subimages were used for feature extraction using the second-order histogram statistics (10 features). In addition, wavelet packets were computed in the regions containing the microcalcifications. The energy and entropy features were computed for the decomposed Daubechies D_6 and D_{20} wavelet packets for Levels 0 and 1 (20 features). The segmentation of microcalcification regions was performed to obtain cluster feature representation (10 features). From the set of all 40 features 20 were selected using a genetic algorithm (GA) based method. These features were used for classification. The list of used features is shown in Table 1.

The new neuro-fuzzy pattern classifier was compared with backpropagation using exactly the same training and test sets. For backpropagation training a maximum of 20,000 epochs were allowed. Two different approaches were used to constructing convex subsets — one (NFPC1) using SHD, the other (NFPC2) using Algorithm A1 to check for point inclusion in the convex set. Two versions of fuzzy membership functions M_f were

Table 1
The extracted features selected for classification of breast cancer images

Feature no.	Description
1	Entropy computed from the second order gray-level histogram
2	The contrast feature of the second order gray-level histogram
3	The mean of the second order gray-level histogram
4	The mean of the difference second order gray-level histogram
5	Energy for the D_6 wavelet packet at Level 0
6	Energy for the D_6 high–high wavelet packet at Level 1
7	Energy for the D_6 high–low wavelet packet at Level 1
8	Energy for the D_6 low–low wavelet packet at Level 1
9	Entropy for the D_6 wavelet packet at Level 0
10	Entropy for the D_6 high–high wavelet packet at Level 1
11	Entropy for the D_6 high–low wavelet packet at Level 1
12	Entropy for the D_6 low–low wavelet packet at Level 1
13	Energy for the D_{20} wavelet packet at Level 0
14	Energy for the D_{20} low–low wavelet packet at Level 1
15	Entropy for the D_{20} high-high wavelet packet at Level 1
16	Entropy for the D_{20} low–high wavelet packet at Level 1
17	Number of microcalcifications
18	Standard deviation of number of gray levels per pixel
19	Standard deviation of gray levels
20	Average distance between calcification and center of mass

implemented resulting in four neuro-fuzzy classifiers. The ones denoted with the asterik (*) used Eq. (5) and the ones without it used Eq. (3). In tables pertaining to computational complexity and number of convex subsets no distinction between the two different membership function implementations is made. This is because the difference in required computations was negligible and the number of convex subsets was not dependent on the utilized convex set fuzzy membership function. All five networks were trained with two outputs for benign and malignant category. The algorithms' ability to infer the knowledge was examined by increasing the ratio (from 40 to 90%, with a step of 10%) of the number of training to total samples, randomly chosen from the population of 191. Each test was run 500 times, giving a total of 3000 runs for an algorithm. For each run the ROC curves were computed from the true and false-positive rates of classification. The ROC curve provides estimates of probabilities of decision outcomes of true-positive and false-positive decisions for any and all of the decision criteria a system might have. In this case the decision criterion used for all classifiers was the threshold at the malignant output node of the classifier above which a test case was classified as malignant. The threshold was varied from -1 to 1, with a step of 0.1, for the backpropagation and from 0 to 1, with a step of 0.05, for the NFPC. In total there were 21 points for each fitting of the ROC curve. The best 50 of the ROC curves that gave chi-square goodness-of-fit parameters that were not significant at the 0.05-probability level, with standard deviation of the fit below 0.08 were used to compute classification rates. In accordance with the results in Ref. [18] the best results for backpropagation network were obtained with 30 hidden layer neurons. The obtained results are shown in Tables 2 and 3, representing mean and maximum classification rates. Table 4 shows a number of ROC fits that met required criteria. Table 5 shows the mean ROC area when best 50% instead of 50 curves were used.

As the relative size of the training set increases the network's classification rate should also increase. This however can happen only when the training set provides more information and the additional samples are not redundant, i.e., they come from other, previously not identified convex clusters. To illustrate the change in information content for the training set, Table 6 shows average number of convex clusters for both categories.

The computational complexity of the new NFPC and the used implementation of the backpropagation algorithm was evaluated and the results, as reported by Matlab, are shown in Tables 7 and 8.

4. Discussion

The new method proposed in this paper was compared with the leading implementation of the backpropagation algorithm. The large number of performed tests for each classifier ensured the accuracy of the comparison between different classifiers. The experiments showed that NFPC performed better than BP neural network classifiers in almost every test, as illustrated by Tables 2–5. The NFPC training method's ability of converging to a valid result is illustrated by Table 4, where the number of valid classification results obtained from each classifier is shown. Only runs with area under the ROC curve greater than 50% and with the standard deviation of fit less than 0.08 were considered valid. The average number of achieved acceptable results for the NFPC was over 60% higher than for backpropagation network. Since identical training sets were used to train all classifiers, it can be inferred that the learning capability of the NFPC

Table 2
Mean ROC area for the five classifiers

	40%	50%	60%	70%	80%	90%
NFPC1*	69.50%	71.63%	73.08%	75.10%	81.40%	83.37%
NFPC2*	67.74%	67.93%	69.59%	69.55%	77.06%	84.79%
NFPC1	70.59%	73.41%	74.97%	76.54%	80.69%	88.37%
NFPC2	69.56%	74.20%	75.90%	74.05%	78.19%	87.17%
BPNN	63.81%	67.89%	70.43%	72.67%	75.45%	81.78%

Table 3
Maximum ROC area for the five classifiers

	40%	50%	60%	70%	80%	90%
NFPC1*	85.62%	86.96%	90.64%	97.83%	100%	100%
NFPC2*	86.68%	85.89%	83.77%	100%	92.31%	100%
NFPC1	85.30%	87.92%	86.64%	92.05%	100%	100%
NFPC2	84.90%	90.66%	94.21%	100%	90.49%	100%
BPNN	74.62%	77.46%	78.44%	84.68%	90.42%	100%

Table 4
Number of successfully computed ROC curves

	40%	50%	60%	70%	80%	90%
NFPC1*	209	217	175	127	155	114
NFPC2*	210	200	189	106	140	101
NFPC1	217	237	217	145	196	155
NFPC2	223	223	193	130	201	149
BPNN	106	123	123	132	120	71

Table 6
Average number of convex subsets for NFPC

	40%	50%	60%	70%	80%	90%
Benign	1.88	2.09	2.31	2.52	2.75	3.04
Malignant	1.98	2.12	2.29	2.45	2.70	2.82
Total NFPC1	3.86	4.21	4.60	4.98	5.45	5.87
Benign	2.43	2.79	3.10	3.41	3.65	3.86
Malignant	2.35	2.68	2.97	3.25	3.51	3.82
Total NFPC2	4.78	5.46	6.07	6.66	7.17	7.68

training algorithm was greater than that of the best backpropagation method available.

The NPFC construction method utilizes an algorithm that checks for point inclusion inside of a convex set. Two versions of the algorithm were used — Algorithm A1 and SHD. An approximate Algorithm A1 requires fewer computations but produces larger number of convex subsets (see Table 6), effectively increasing the size of the classifier. However, the classification rate of the NFPC designed using Algorithm A1 or SHD does not change significantly. This is a proof to the method's overall robustness, since even when the learned convex subsets are not accurate, classifier's performance is not affected.

The introduced algorithm is a constructive method that continues to expand the classifier's structure until all training samples are separated in the feature space.

Table 5
Mean ROC area when best 50% curves were used

	40%	50%	60%	70%	80%	90%
NFPC1*	64.55%	66.08%	68.03%	72.81%	77.05%	81.95%
NFPC2*	63.35%	63.70%	65.37%	69.07%	74.55%	84.54%
NFPC1	65.67%	67.75%	69.73%	74.02%	76.27%	83.52%
NFPC2	64.78%	68.51%	71.40%	72.17%	74.01%	82.64%
BPNN	63.55%	66.67%	69.07%	70.68%	73.99%	86.41%

Table 7
Mean number of floating point operations ($\times 10^9$)

	40%	50%	60%	70%	80%	90%
Set creation	0.0425	0.1098	0.2312	0.4814	0.9107	1.6919
Separation	0.0576	0.0760	0.0954	0.1168	0.1439	0.1763
Total NFPC1	0.1001	0.1858	0.3266	0.5981	1.0546	1.8682
Set creation	0.0402	0.0916	0.1748	0.3017	0.4477	0.6683
Separation	0.0623	0.0877	0.1166	0.1511	0.1865	0.2268
Total NFPC2	0.1025	0.1794	0.2914	0.4528	0.6343	0.8951
BPNN	1.8735	2.4607	3.3889	4.2499	5.4922	6.6037

Table 8
Max number of floating point operations ($\times 10^9$)

	40%	50%	60%	70%	80%	90%
Set creation	0.0794	0.2095	0.4474	0.8968	1.5555	2.7161
Separation	0.0875	0.1144	0.1354	0.1709	0.2120	0.2419
Total NFPC1	0.1485	0.3203	0.5484	1.0231	1.7086	2.8737
Set creation	0.0937	0.3245	0.4608	0.6674	1.4373	2.1125
Separation	0.0947	0.1298	0.1786	0.2273	0.2867	0.3267
Total NFPC2	0.1692	0.4349	0.5554	0.8404	1.6220	2.3125
BPNN	4.7792	5.9357	7.0338	8.2522	9.4091	10.6096

Table 9
Expected vs. real complexity — breast cancer data

	40%	50%	60%	70%	80%	90%
Mean set creation	0.0402	0.0916	0.1748	0.3017	0.4477	0.6683
Estimate $O(n^4d)$	0.6672	1.6987	3.4980	6.4484	10.9596	17.5043
Ratio estimate/mean	16.60	18.54	20.01	21.37	24.48	26.19

Therefore it always converges to a solution. The inherent structural approach to (convex) clustering data improves the algorithm's chances of obtaining valid solutions, as it was demonstrated by the experimental results.

The computational complexity of the entire method depends predominantly on the complexity of Algorithms A1 and A2. In Algorithm A1, step 7 has the most decisive impact on the total algorithm's performance, as its complexity is $O(nd)$. It is repeated $2n$ times, resulting in the entire algorithm's complexity $O(n^2d)$. In Algorithm A2, it is step 2 that determines overall complexity of the algorithm. In the worst case, that step is repeated $\sim O(N^2)$ times, where N is a number of training cases from all categories. Thus, the maximum computations required for the algorithm to complete is $O(N^4d)$. In reality, as the results of the performed tests show, this complexity is significantly lower. If instead of Algorithm A1 SHD is used, the method's expected complexity remains of the same order.

Results shown in Tables 7 and 8 prove that the proposed method's computational complexity is lower than that for backpropagation training in every test performed. Also, the number of hyperplanes in the hyperplane layer of the NFPC was always smaller than that of used backpropagation network. The algorithm's computational complexity is strongly dependent on the number of training samples, and only linearly dependent on the data's dimensionality. The method's nature makes a precise computation estimate difficult, and as it is expected, the cluster structure significantly influences the complexity. Table 9 illustrates the difference between expected and measured computational complexity. It shows the comparison of mean set creation time versus its estimate. The mean set creation and estimate units in

the corresponding tables are 10^9 flops and 10^8 flops. In both tables second version of the NFPC construction algorithm (using Algorithm A1) was used for comparison. The method performed 16.6–26 times faster than the estimates. The increase in the estimate/mean ratio with increasing size of the training set suggest that the real computational complexity of the algorithm is smaller than $O(N^4 d)$.

5. Conclusion

The introduced method constructs a neuro-fuzzy pattern classifier by identifying convex subsets of pattern points in the feature space. It provides a clear theoretical basis for understanding the significance of the feature space and its contribution towards classification. The input to the system is a set of crisp feature vectors. The training result is a pattern classifier comprising a set of fuzzy functions that reflect the input pattern's degree of membership in a number of convex subsets of the feature space, identified during the NFPC training stage. The proposed training procedure is completely automated — function parameters are automatically computed from statistical distributions of the data. Two different approaches to construction of fuzzy membership functions were tested: sigmoidal decision surfaces — (backpropagation-like approach) and bell-shaped functions — cluster-specific approach. For the tests performed the backpropagation-like approach achieved considerably better results than the cluster specific approach. In the process of constructing convex sets two algorithms were used: A1 and SHD. Both performed equally well proving robustness of the undertaken approach to clustering. In the conducted tests the proposed training method performed better than the leading implementation of the backpropagation training method in terms of rate of convergence and computational complexity. In most of the tests the resulting neuro-fuzzy classifier achieved higher classification rates than the backpropagation network. Additionally, the convergence rate of the NFPC training was shown to be higher than that of the leading implementation of backpropagation algorithm.

Appendix

Definition A.1 (*Line segment*). If L is a linear space, $X \in L$, $Y \in L$, the line segment XY joining X and Y is the set of all points of the form $\alpha X + \beta Y$ where $\alpha \geqslant 0, \beta \geqslant 0, \alpha + \beta = 1$.

Definition A.2 (*Convex set*). A set $S \subset L$ is convex if for each pair of points $X \in S$, $Y \in S$ it is true that $XY \subset S$, where XY is the line segment joining X and Y.

Definition A.3 (*Extreme point*). If S is a convex set in L, then a point $X \in S$ is an extreme point of S if no degenerate segment in S exists which contains X in its relative interior. In other words, X lies on the end of every line segment contained in S, that it belongs to. A set of extreme points is a subset of the set of all boundary points of S.

Definition A.4 (*Closed set*). A set S is said to be closed if every boundary point of S belongs to S.

Definition A.5 (*Convex set hull*). The convex hull of a set $S \subset L$ is the intersection of all convex sets in L containing S, and it is denoted as conv S. The closed convex hull of S is the intersection of all closed convex sets containing S. Clearly S is convex if and only if conv $S = S$.

Definition A.6 (*Hyperplane*). The equations

$$H: x_d w_d + x_{d-1} w_{d-1} + \cdots + x_1 w_1 + w_0 = \sum_{i=1}^{d} x_i w_i + w_0 = 0 \tag{A.1}$$

and

$$H: \varphi(x) = 0 \quad \text{where } \varphi(x) = \sum_{i=1}^{d} x_i w_i + w_0$$

represent a $(d-1)$-dimensional hyperplane H in the d-dimensional space. This hyperplane divides the space into two separate regions, in one the value of $\varphi(x)$ is positive — positive side of the hyperplane, and the other in which the value of $\varphi(x)$ is negative — negative side of the hyperplane.

Lemma A.1. *As a consequence of the above definitions, a convex hull of S is identical with the convex hull of extreme points of S.*

Lemma A.2. *The coordinates of every point P laying inside of the convex hull with vertices $v_1, v_2, \ldots, v_n$ can be expressed by the linear combination of the coordinates of the hull's vertices, i.e.*

$$P = \sum_{i=1}^{n} \alpha_i v_i \quad \text{and} \quad \sum_{i=1}^{n} \alpha_i = 1.$$

The proof of the above comes from the definition, which states that for every point P in a convex set S there exist two points X_1 and $X_2 \in S$, such that [28]

$$P = \alpha X_1 + (1-\alpha) X_2 \quad \text{where } \alpha \in \langle 0, 1 \rangle.$$

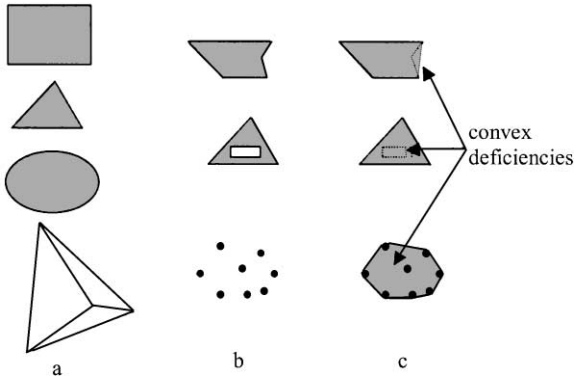

Fig. 14. Convex, concave sets and convex hulls.

The figure is convex when any two of its points can be connected by a straight-line segment that lies entirely within this figure. Examples of convex and concave sets are shown in Fig. 14a and b, respectively. Thus any circle, rectangle, trapezoid or triangle is a 2-D convex set and any sphere, cube or tetrahedron is a 3-D convex set. On the other hand, any figure with empty space engulfed in it, is not convex in any dimension. Obviously, a single point constitutes a convex set, but any set of more than one points is never convex. However, any concave set can always be turned into a convex set (its convex hull can be created) by adding its convex deficiencies. This concept is illustrated in Fig. 14c, where the convex hulls of concave sets from Fig. 14b are shown.

In d-dimensional space s*implex* is a d-dimensional convex hull constructed on exactly $d+1$ points. It has exactly $d+1$ *facets* and *vertices* (*extreme points*). When $d=2$ the simplex is a triangle, when $d=3$ it is a tetrahedron, etc.

References

[1] S.K. Pal, S. Mitra, Multilayer perceptron, fuzzy sets and classification, IEEE Trans. Neural Networks 3 (5) (1992) 683–697.

[2] S. Mitra, S.K. Pal, Fuzzy multi-layer perceptron, inferencing and rule generation, IEEE Trans. Neural Networks 6 (1) (1995) 51–63.

[3] J.N.K. Liu, K.Y. Sin, Fuzzy neural networks for machine maintenance in mass transit railway system, IEEE Trans. Neural Networks 8 (4) (1997) 932–941.

[4] P. Gader, M. Mohamed, J.-H. Chiang, Comparison of crisp and fuzzy character neural networks in handwritten word recognition, IEEE Trans. Fuzzy Systems 3 (3) (1995) 357–364.

[5] J. Chang, G. Han, J.M. Valverde, N.C. Griswold, J.F. Duque-Carrillo, E. Sánchez-Sinencio, Cork quality classification system using a unified image processing and fuzzy-neural network methodology, IEEE Trans. Neural Networks 8 (4) (1997) 964–974.

[6] Y.-Q. Zhang, A. Kandel, Compensatory neurofuzzy systems with fast learning algorithms, IEEE Trans. Neural Networks 9 (1) (1998) 83–105.

[7] P.K. Simpson, Fuzzy min-max neural networks — Part 1: classification, IEEE Trans. Neural Networks 3 (5) (1992) 776–787.

[8] W. Pedrycz, Computational Intelligence: An Introduction, CRC Press, New York, 1998.

[9] J. Zhang, A.J. Morris, Recurrent neuro-fuzzy networks for nonlinear process modeling, IEEE Trans. Neural Networks 10 (2) (1999) 313–326.

[10] I.H. Suh, T.W. Kim, Fuzzy membership function based neural networks with applications to the visual servoing of robot manipulators, IEEE Trans. Fuzzy Systems 2 (3) (1994) 203–220.

[11] H.K. Kwan, Y. Cai, A fuzzy neural network and its application to pattern recognition, IEEE Trans. Fuzzy Systems 2 (3) (1994) 185–193.

[12] V. Petridis, V.G. Kaburlasos, Fuzzy Lattice Neural Network (FLNN): a hybrid model for learning, IEEE Trans. Neural Networks 9 (5) (1998) 877–890.

[13] J.-H. Chiang, P.D. Gader, Hybrid fuzzy-neural systems in handwritten word recognition, IEEE Trans. Fuzzy Systems 5 (4) (1997) 497–510.

[14] G. Purushothaman, N.B. Karayiannis, Quantum Neural Networks (QNN'S): inherently fuzzy feedforward neural networks, IEEE Trans. Neural Networks 8 (3) (1997) 679–693.

[15] M. Lanyi, Diagnosis and Differential Diagnosis of Breast Calcifications, Springer, New York, 1986.

[16] E. Marshall, Search for a killer: focus shifts from fat to hormones, Science 259 (1993) 618–621.

[17] E.A. Sickles, D.B. Kopans, Mammographic screening for women aged 40 to 49 years: the primary practitioners dilemma, Ann. Int. Med. 122 (7) (1995) 534–538.

[18] A.P. Dhawan, Y. Chitre, C. Kaiser-Bonasso, M. Moskowitz, Analysis of mammographic microcalcifications using gray-level image structure features, IEEE Trans. Med. Imaging 15 (3) (1996) 246–259.

[19] G. Mirchandani, W. Cao, On hidden nodes for neural nets, IEEE Trans. Circuits Systems 36 (5) (1989) 661–664.

[20] W.M. Grohman, Neuro-fuzzy pattern classifier with convex sets, Ph.D. Dissertation, Department of Bioengineering, University of Toledo, 1999.

[21] N.J. Nilsson, The Mathematical Foundations of learning Machines, Morgan Kaufmann, San Mateo, CA, 1990.

[22] D.G. Luenberger, Optimization by Vector Space Methods, Wiley, New York, 1969.

[23] I.H. Suh, J.-H. Kim, F.Ch-H. Rhee, Convex-set-based fuzzy clustering, IEEE Trans. Fuzzy Systems 7 (3) (1999) 271–285.

[24] J.-B. Hiriart-Urruty, C. Lemaréchal, Convex analysis and minimization algorithms, Springer, Berlin, 1993.

[25] M. Gori, A. Tesi, On the problem of local minima in backpropagation, IEEE Trans. Pattern Anal. Mach. Intell. 14 (1) (1992) 76–86.

[26] Ch. Kaiser-Bonasso, Genetic algorithm input feature selection and radial basis function classification of mammographic microcalcifications, M.Sc. Thesis, Department of Electrical and Computer Engineering, University of Cincinnati, 1995.

[27] T.P. Vogl, J.K. Mangis, A.K. Rigler, W.T. Zink, D.L. Alkon, Accelerating the convergence of the backpropagation method, Biol. Cybernet. 59 (1988) 257–263.

[28] F. Valentine, Convex Sets, McGraw-Hill, New York, 1964.

About the Author—WOJCIECH M. GROHMAN received his M.S. degree in Electronics from Silesian Technical University in Gliwice, Poland, in 1995. He was a research assistant at the University of Toledo, where he obtained his Ph.D. in Bioengineering with specialization in artificial intelligence in 1999. Currently he is working for EG&G Astrophysics Research Corp. His research interests include pattern recognition, neural networks and other artificial intelligence methods.

About the Author—ATAM P. DHAWAN, Ph.D. obtained his B.Eng. and M. Eng. degrees in Electrical Engineering from the University of Roorkee, Roorkee, India. He was a Canadian Commonwealth Fellow at the University of Manitoba where he completed his Ph.D. in Electrical Engineering with specialization in medical imaging and image analysis in 1985. In 1984, he won the first prize and the Martin Epstein Award in the Symposium of Computer Application in Medical Care Paper Competition at the Eighth SCAMC Annual Congress in Washington, DC, for his work on developing a three-dimensional (3D) imaging technique to detect early skin-cancer called melanoma. From 1985 to 1988, he was an Assistant Professor in the Department of Electrical Engineering at the University of Houston. Later, in 1988, he joined the University of Cincinnati as an Assistant Professor where he became Professor of Electrical and Computer Engineering and Computer Science, and Radiology (joint appointment). From 1990 to 1996, he was the Director of Center for Intelligent Vision and Information System. From 1996 to 1998, he was Professor of Electrical Engineering at the University of Texas at Arlington, and Adjunct Professor of Radiology at the University of Texas Southwestern Medical Center at Dallas. He is currently Professor in the Department of Electrical and Computer Engineering at the New Jersey Institute of Technology. Dr. Dhawan has published more than 50 research articles in refereed journals, and edited books, and 85 research papers in refereed conference proceedings. Dr. Dhawan is a recipient of Martin Epstein Award (1984), National Institutes of Health FIRST Award (1988), Sigma-Xi Young Investigator Award (1992), University of Cincinnati Faculty Achievement Award (1994) and the prestigious IEEE Engineering in Medicine and Biology Early Career Achievement Award (1995). He is an Associate Editor of IEEE Transactions on Biomedical Engineering, Associate Editor of IEEE Transactions on Rehabilitation Engineering, and Editor of International Journal of Computing Information and Technology. He has served on many IEEE EMBS professional committees and has delivered Workshops on Intelligent Biomedical Image Analysis in IEEE EMBS International Conferences (1996, 1997). He is the Chair of the "New Frontiers in Biomedical Engineering" Symposium at the World Congress 2000 on Medical Physics and Biomedical Engineering.

His current research interests are medical imaging, multimodality brain mapping, intelligent image analysis, multigrid image reconstruction, wavelets, genetic algorithms, neural networks, adaptive learning and pattern recognition.