Lecture Notes in Computer Science, vol. 5164, 754-762, 2008.

Projection Pursuit Constructive Neural Networks Based on Quality of Projected Clusters

Marek Grochowski and Włodzisław Duch

Department of Informatics, Nicolaus Copernicus University, Toruń, Poland,
grochu@is.umk.pl; Google: W Duch

Abstract. Linear projection pursuit index measuring quality of projected clusters (QPC) is used to discover non-local clusters in high-dimensional multiclass data, reduction of dimensionality, feature selection, visualization of data and classification. Constructive neural networks that optimize the QPC index are able to discover simplest models of complex data, solving problems that standard networks based on error minimization are not able to handle. Tests on problems with complex Boolean logic and a few real world datasets show high efficiency of this approach.

1 Introduction

Theoretical analysis of non-separable classification problems, introduced in [1], shows that complexity of data classification is proportional to the minimum number of intervals that are needed to separate pure clusters of data in a single linear projection. Problems that require at least k such intervals are called k-separable. For example, n-bit parity problems are $n+1$-separable [2], because linear projection of binary strings exists that forms clusters with fixed number of 1 bits, from 0 to n. Neural networks based on basis set expansions, such as the Radial Basis Function (RBF) networks, Multi-Layer Perceptrons (MLPs), or other standard neural models [3] that use error minimization techniques, are not able to discover such simple models of data for high index k. The same is true for Support Vector Machines (SVMs) and other classifiers. Yet many problems in bioinformatics or text analysis may have inherent complex logic that needs to be discovered.

Transformations on the input space may allow to find interesting low dimensional representations, revealing structures impossible to anticipate looking at the original dataset. The simplest transformations with easy interpretations are linear projections. When a given data is linearly separable then a single projection is sufficient to solve the problem. For more complicated data structures projection on a single direction may show multimodal data distributions, creating clusters that reflect interesting information about the data. Even if k-separable solution exist clusters projected on a line may be small and separation between them may be narrow. More reliable predictions are possible if all data vectors are projected on large pure clusters. Additional projections may provide useful large clusters. For example, in parity problem projection on a $[1, 1, 1..1]$ direction and $[1, -1, 1, -1...]$ direction creates pure clusters of different size allowing for high confidence predictions for all data vectors.

A lot of methods that search for optimal and most informative linear transformations have been developed. A general projection pursuit (PP) framework to find interesting data transformations by maximizing some ”index of interest” has been proposed by Friedman [4, 5]. PP index assigns numerical values to data projections (or more general transformations). For example, the Fisher Discriminant Analysis (FDA) algorithm [6] defines an index that measures the ratio of between-class scatter to within-class scatter. Local FDA (LFDA) is an interesting extension of Fisher's method, proposed by Sugiyama [7]. This method tries to preserve local structure of data and can deal with multimodal class distributions where FDA fails. The approach presented below also belongs to the general projection pursuit framework, and may be implemented as a constructive neural network. In the next section a new index based on the Quality of Projected Clusters (QPC) is proposed. It allows to find compact, pure clusters of vectors separated from other clusters. In contrast to most similarity-based methods [8, 9] that optimize metric functions to capture local clusters, projection pursuit may discover interesting non-local projections. Maximization of the QPC index by gradient descent optimization are quite interesting. A few practical issues related to the application of the QPC index are presented in the next section. Searching for projections into two or more-dimensional spaces allows for visualization of data, as shown in section three. The use of QPC index as a basis of a constructive neural network is described in section four. The final section contains discussion and future perspectives.

2 The QPC Projection Index

The celebrated MLP backpropagation of errors training algorithm does not define specific target for hidden layers, trying instead to adapt input weights in a way that contributes to the overall reduction of some error function. It has been quite successful for problems that have relatively low complexity (measured by the separability index), but fails already for Boolean functions that are 4 or 5-separable. The PP index should help to discover interesting linear projections of multiclass data, and localize groups of vectors that belong to the same class in compact clusters separated from other clusters. Consider a dataset $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathcal{R}^d$, where each vector $\boldsymbol{x}_i$ belongs to one of k different classes. For a given vector $\boldsymbol{x} \in \mathcal{X}$ with a label $\mathcal{C}$ QPC index is defined as:

$$Q(\boldsymbol{x};\boldsymbol{w}) = A^{+} \sum_{\boldsymbol{x}_k \in \mathcal{C}} G\left(\boldsymbol{w}^T(\boldsymbol{x}-\boldsymbol{x}_k)\right) - A^{-} \sum_{\boldsymbol{x}_k \notin \mathcal{C}} G\left(\boldsymbol{w}^T(\boldsymbol{x}-\boldsymbol{x}_k)\right) \tag{1}$$

where the $G(x)$ localized function achieves maximum for $x = 0$ and should have compact ϵ-support for all $x \in \mathcal{R}$. The first term in Eq. (1) is large if projection on a line, defined by $\boldsymbol{w}$ weight vector, groups many vectors from class $\mathcal{C}$ close to $\boldsymbol{x}$. The second term estimates separation between a given vector $\boldsymbol{x}$ and vectors from classes other than $\mathcal{C}$. It introduces penalty for placing projected $\boldsymbol{x}$ too close to projected vectors from other classes. The average of the $Q(\boldsymbol{x};\boldsymbol{w})$ indices for all vectors:

$$QPC(\boldsymbol{w}) = \frac{1}{n} \sum_{\boldsymbol{x} \in \mathcal{X}} Q(\boldsymbol{x};\boldsymbol{w}) \,, \tag{2}$$

forms an overall measure of how interesting a projection on direction $\boldsymbol{w}$ is, providing a leave-one-out type estimation. The value of this index is large if projection on $\boldsymbol{w}$ gives clusters that are pure, compact and well separated from clusters of vectors with other labels. For linearly separable problems function $QPC(\boldsymbol{w})$ achieves maximum if projection $\boldsymbol{wx}$ creates two well-separated pure clusters. If dataset is k-separable then maximization of this index should find a projection with k separated clusters, that may then be easily classified defining simple intervals or using a special neural architecture [2].

Parameters A^+, A^- control influence of each term in Eq. (1), and may simply be fixed to balance and normalize the value of projection index, for example at $A^+ = 1/p(\mathcal{C})$ and $A^- = 1/(1 - p(\mathcal{C}))$ (where $p(\mathcal{C})$ is the *a priori* class probability). If large A^- values are used a stronger separation between classes is enforced, while larger A^+ will have an impact mostly on compactness and purity of created clusters. Influence of each vector on projection index is determined by properties of function $G(x)$. This function should be localized, achieving a maximum value for $x = 0$. If $G(x)$ is continuous then gradient-based methods may by used to find maximum of the QPC index. All bell-shaped functions are suitable for $G(x)$, including Gaussian and an inverse quartic function:

$$G(x) = \frac{1}{1 + (bx)^4} \tag{3}$$

where parameter b controls the width of $G(x)$, and thus determines influence of the neighboring vectors on the index value. Another useful function may be constructed from a combination of two sigmoidal functions (a bicentral function [10, 11]):

$$G(x) = \sigma(x + b) - \sigma(x - b) \tag{4}$$

Constructive MLP networks with special architecture could be used to implement approximations to the $QPC(\boldsymbol{w})$ index using several hidden layers. Neurobiological justification for constructive models of networks calculating PP indices is given in the final section.

Gradient based maximization, like most iterative optimization procedures, does not guarantee an optimal solution. However, multistart gradient approach has been quite effective in searching for interesting projections. Although solutions are not always unique they may sometimes provide additional insight into the structure of data. Calculation of function (2) requires $O(n^2)$ operations. Various "editing techniques" used for the nearest neighbor methods with very large number of vectors [12] may decrease this complexity to $O(n \log n)$. This may be done by sorting projected vectors and restricting computations of the sum in Eq. (1) only to vectors $\boldsymbol{x}_i$ with $G(\boldsymbol{w}(\boldsymbol{x} - \boldsymbol{x}_i)) > \epsilon$. The cost is further decreased if centers of projected clusters are defined and a single sum $G(\boldsymbol{w}(\boldsymbol{x} - t))$ used. Simple gradient descent methods may be replaced by second-order approaches, or if non-differentiable $G(x)$ is taken (ex. triangular, or trapezoidal function), by stochastic or systematic search-based methods [13]). All these improvements are important but technical, while here the potential of the model that is using localized projected clusters will be stressed.

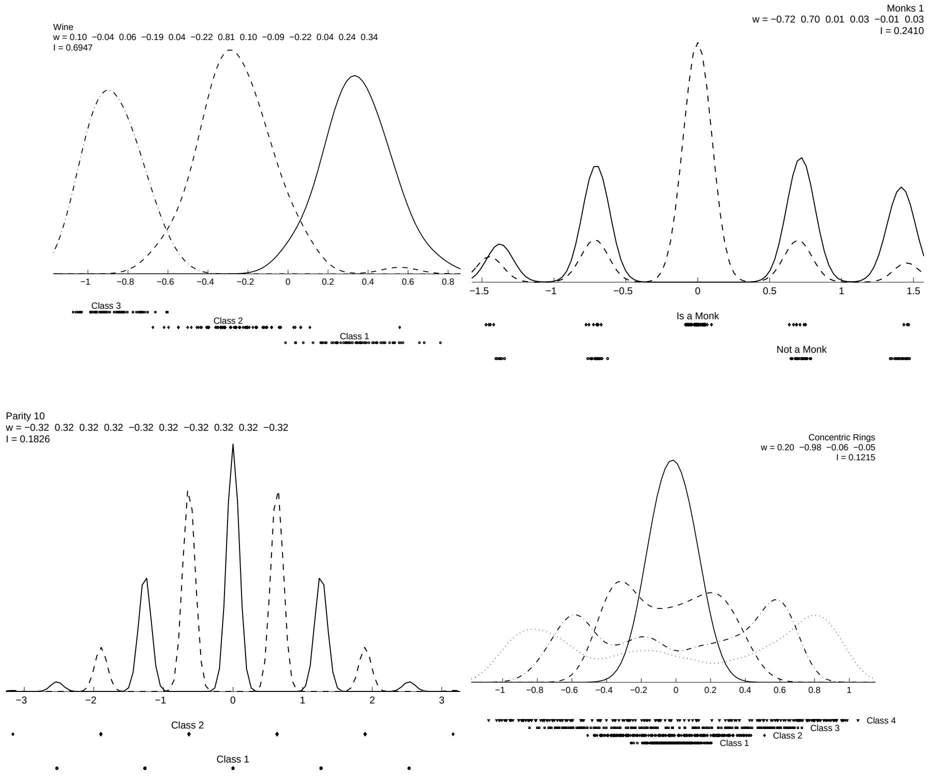

Fig. 1. Examples of projections found by maximization of the projection index using gradient descent method for Wine dataset (top-left), Monk's 1 problem (top-right), 10 bit Parity (bottom-left) and Concentric Rings with noise (bottom-right).

Figure 1 presents examples of projections that give maximum value of the $QPC(\boldsymbol{w})$ index for four very different datasets. All projections were obtained taking Eq. (3) for $G(x)$, with $b = 3$, with simple gradient descent maximization initialized 10 times, selecting after a few iterations the most promising solution that is trained until convergence. Values of weights and the value of $QPC(\boldsymbol{w})$ are shown in the corresponding figures. Positions of vectors from each class are shown below the projection line. Smoothed histograms may be normalized and taken as estimations of class conditionals $p(x|C)$, from which posterior probabilities $p(C|x) = p(x|C)p(C)/p(x)$ may easily be calculated.

The top left figure shows the Wine dataset from the UCI repository [14], with 13 features and 3 classes. It can easily be classified using a single linear projection that gives three groups of vectors (one for each class) with almost perfect separation between them. The top right figure shows symbolic Monk 1 datasets [14], with 6 symbolic features and two classes. All vectors of the Monk 1 problem can be classified correctly with two simple rules. Large cluster of vectors in the middle of the projection presented in Fig. 1 (first two coefficients are equal, others are essentially zero) corresponds to a rule: if head shape = body shape then object is called a Monk. To separate the remaining cases a second projection is needed (see below). These logical rules may also be extracted from an MLP networks [15].

The lower left figure shows parity problem in 10 dimensions, with 512 even and 512 odd binary strings. This problem is 11-separable, with a maximum value of projection index obtained for diagonal direction in the 10 dimensional hypercube, therefore all weight have the same value. Although a perfect solution using one projection has been found some clusters at extreme left and right of this projection are quite small and additional directions are worth exploring. MLP or RBF networks will have to create quite complex data models using sufficient number of hyperplanes or radial functions, and converges to a good solution will be in this case quite unlikely.

In the last example an artificial Concentric Rings dataset has been constructed. It has 4 classes, each with 200 samples, with 4 features, the first and second feature defining points inside 4 concentric rings, and the third and fourth containing uniformly distributed random numbers. For this dataset best projection that maximizes the QPC index reduces influence of noisy features, with weights for dimensions 3 and 4 close to zero. This shows that the QPC index may be used for feature selection, but also that linear projections have limited power, and in this case will lead to a rather complex solution requiring many projections at different angles, while using radial functions allows for creation of much simpler network. The need for networks with different types of transfer functions [10, 11] has been stressed some time ago but still there are no programs capable of finding the simplest data models in all cases.

3 Visualization

Additional directions for interesting projections can be found in several ways. First, the projection pursuit approach [4, 5] orthogonalizes all data to the first direction $\boldsymbol{w}$, repeating the same procedure to generate the second projection. This warrants that different direction is found at each such iteration. Second, one can focus on clusters of vectors that overlap in the first projection, and use only the subset of these vectors to maximize the PP index to find the second direction. The third possibility is to search for the next linear projection with additional penalty term that will punish solutions similar to those found earlier:

$$QPC(\boldsymbol{w};\boldsymbol{w}_1) = QPC(\boldsymbol{w}) - \lambda f(\boldsymbol{w},\boldsymbol{w}_1)\ . \tag{5}$$

The value of $f(\boldsymbol{w},\boldsymbol{w}_1)$ should be large if the current $\boldsymbol{w}$ is close to previous direction $\boldsymbol{w}_1$. For example, some power of the scalar product between these directions may be used:

$$f(\boldsymbol{w},\boldsymbol{w}1) = ({\boldsymbol{w}_1}^T \cdot \boldsymbol{w})^2\ . \tag{6}$$

Parameter λ controls the influence of additional term on the optimization process.

Scatterplots of data vectors projected on two directions may be used for visualization. Figure 2 presents such scatterplots for the four datasets used in the previous section. The second direction $\boldsymbol{w}$, found by gradient descent optimization of function

(5) with $\lambda = 0.5$, is used for the horizontal axis. The final weights of the second direction, value of the projection index $QPC(\boldsymbol{w})$ and the inner product of $\boldsymbol{w}_1$ and $\boldsymbol{w}$ are shown in the corresponding graphs.

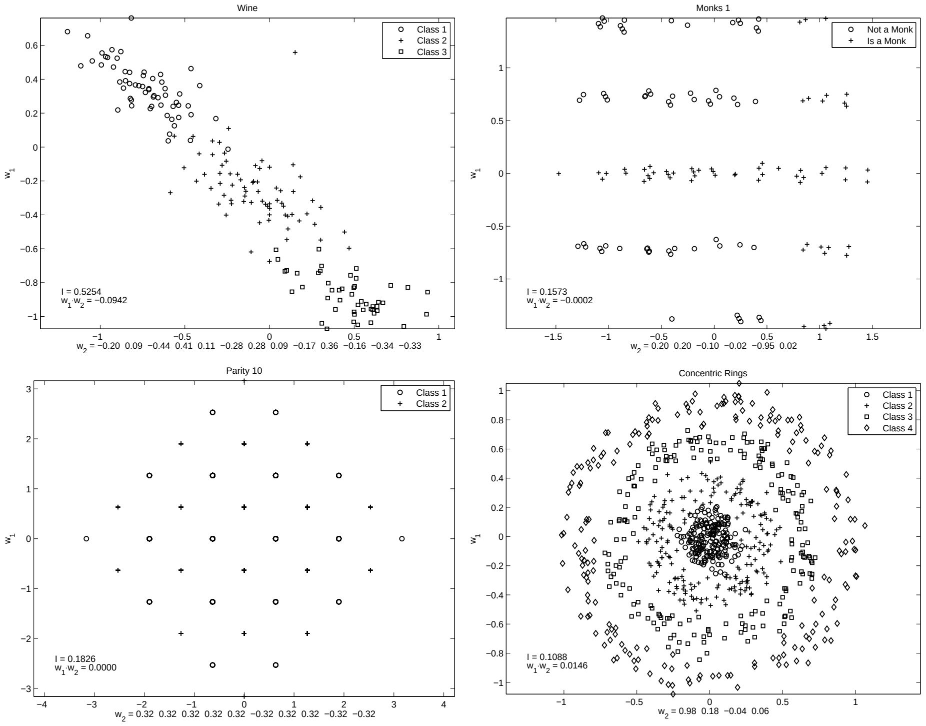

Fig. 2. Examples of scatterplots created by projection on two directions for the Wine dataset (top-left), Monk 1 problem (top-right), 10 bit parity (bottom-left) and the noisy Concentric Rings data (bottom-right).

Fig. 1 shows that in the Wine problem two classes are perfectly separated by the measure of flavanoids, a feature has been dominating and was almost sufficient to separate all three classes. Two dimensional solution for Monk's 1 forms separate and compact groups of vectors. The 5th feature (which forms the second rule describing this dataset: if it is 1 then object is a Monk) has significant value, and all unimportant features have weights equal almost to zero, allowing for simple extraction of correct logical rules. In case of the 10-bit parity problem each diagonal direction of a hypercube representing Boolean function gives a good solution with large cluster in the center. Two such orthogonal directions have been found, projecting each data vector into large pure cluster, either in the first or in the second dimension. Results for the noisy Concentric

Rings dataset shows that maximization of the QPC index has caused vanishing of noisy and uninformative features, and has been able to discover two-dimensional relations hidden inside this data. Although linear projections in two directions cannot separate this data, such dimensionality reduction is sufficient for any similarity-based method, for example the nearest neighbor method, to solve perfectly the problem.

4 Constructive Neural Network

Weights obtained from maximization of the QPC index (2) may be useful in several machine learning tasks. Reduction of dimensionality to one dimensions allows for estimation and drawing class-conditional and posterior probabilities, but may be not sufficient for optimal classification. Reduction to two or three dimensions allows for visualization of scatterograms, showing interesting structures hidden in the high-dimensional datasets and suggesting how to handle the problem in a simplest way: adding linear output layer (Wine), localized functions, using intervals (parity), or nearest neighbor rule (Concentric Rings). Reduction to higher number of dimensions will be very useful as a pre-processing for final classification.

Coefficients of the projection vectors may be used directly for feature ranking and feature selection models, because maximization of the QPC index gives negligible weights for noisy or insignificant features, while important attributes have distinctly larger values. This method might be used to improve learning for many machine learning model that are sensitive to feature weighting, such as kNN. Interesting projections may also be used to initialize weights in many neural network architectures, including MLP networks. The QPC index Eq. (1) defines specific representation for the hidden layer of a constructive neural network. This may be used in several ways to construct nodes of the network.

In most cases, projections that maximize $QPC(\boldsymbol{w})$ contain at least one large cluster. The center of this cluster can be directly estimated during maximization of the $Q(\boldsymbol{x};\boldsymbol{w})$ index, because it is associated with some vector $\boldsymbol{x}_i \in \mathcal{X}$ that gives the maximum value of $Q(\boldsymbol{x}_i;\boldsymbol{w})$ in Eq. (1). Consider the node M implementing the following function:

$$M(\boldsymbol{x}) = \begin{cases} 1 & \text{if} \quad |G(\boldsymbol{w}(\boldsymbol{x}-\boldsymbol{t})) - \theta| \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where the weights $\boldsymbol{w}$ are obtained by maximization of projection index, and $\boldsymbol{t}$ is the center of cluster associated with maximum $Q(\boldsymbol{x};\boldsymbol{w})$. This node splits input space into two disjoint subspaces, with output 1 for each vector that belongs to the cluster and 0 for all other vectors. It is fairly easy to solve the parity-like problems with such nodes, summing the output of all nodes that belong to the same class.

Further adjustments of weights and center of the cluster can enlarge the cluster and give better separation between classes. This can be done by maximizing $Q(\boldsymbol{t};\boldsymbol{w})$ with respect to weights $\boldsymbol{w}$ and cluster center $\boldsymbol{t}$, or by minimization of an error function:

$$E(\boldsymbol{x}) = E_{\boldsymbol{x}}||G\left(\boldsymbol{w}(\boldsymbol{x}-\boldsymbol{t})\right) - \delta(c_{\boldsymbol{x}}, c_{\boldsymbol{t}})|| \tag{8}$$

where $\delta(c_{\boldsymbol{x}}, c_{\boldsymbol{t}})$ is equal to 1 when $\boldsymbol{x}$ belongs to the class associated with cluster with center $\boldsymbol{t}$, and 0 if not (some possible expansions of this error function making it more sensitive for a number of separated vectors and purity of solution may be considered; see [2] for details).

This method has twice as many parameters to optimize (weights and center), but computational cost of calculation of function $Q(\boldsymbol{t};\boldsymbol{w})$ is linear with respect to the number of vectors $O(n)$, and since only a few iterations are needed this part of learning is quite fast. Final neuron should give good separation between the largest possible group of vectors with the same labels, and the rest of the dataset. Next node is created in the same manner, but before learning vectors correctly handled by previous nodes should be removed from the training dataset. This procedure is called general sequential constructive method [16], and it leads, in a finite number of steps, to creation of neural network which classifies all samples of a given multiclass dataset, where each neuron derived by this method is placed in hidden layer, and weights in the output layer are determined from a simple algebraic equation (for details see [16]). Although we do not have space here to report detailed results they are an improvement over already excellent results obtained in [2] and similar to [7].

5 Discussion

Projection pursuit networks that reduce dimensionality and use clustering, such as the QPC networks described in this paper, are able to find the simplest data models (including logical rules) in case of quite diverse and rather complex data. They create interesting features, allowing for visualization and classification of data. Should such networks be called neural?

Multilayer perceptrons use threshold neurons that have neurobiological inspirations, but backpropagation algorithm is quite hard to justify from biological perspective. Basis set expansion networks (such as RBF networks) have roots in approximation theory rather than biology. Inspirations for the projection pursuit networks should be searched at a higher level than single neurons. Non-local projections that form low-dimensional clusters may be realized by neural columns that are activated by linear combinations of incoming signals, learn to remember groups of vectors that give similar projections, and learn better weights to increase their excitations, inhibiting at the same time competing columns. Different mechanisms may then be used to extract interesting transformation from such columns, reducing noise in data, selecting relevant information, learning to estimate similarity of responses. A column may learn to react to inputs of specific intensity, solving complex logical problem by clustering data in low-dimensional projections that are not linearly separable.

Linear separability is not the best goal of learning. The QPC index helps to solve problems that go well beyond capabilities of standard neural networks, such as the parity or the noisy Boolean functions. The class of PP networks is quite broad. One can implement many transformations in the hidden layer, explicitly creating hidden representations that are used as new inputs for further network layers, or used for initialization of standard networks. Brains are capable of deep learning, with many specific transformations that lead from simple contour detection to final invariant object recognition. Studying PP networks should bring us a bit closer to powerful methods for deep learning.

References

1. Duch, W.: k-separability. Lecture Notes in Computer Science **4131** (2006) 188–197
2. Grochowski, M., Duch, W.: Learning highly non-separable Boolean functions using Constructive Feedforward Neural Network. Lecture Notes in Computer Science **4668** (2007) 180–189
3. Haykin, S.: Neural Networks - A Comprehensive Foundation. Maxwell MacMillian Int., New York (1994)
4. Friedman, J.: Exploratory projection pursuit. Journal of the American Statistical Association **82** (1987) 249–266
5. Jones, C., Sibson, R.: What is projection pursuit. Journal of the Royal Statistical Society A **150** (1987) 1–36
6. Fisher, R.: The use of multiple measurements in taxonomic problems. Annals of Eugenics **7** (1936) 179–188
7. Sugiyama, M.: Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. Journal of Machine Learning Research **8** (2007) 1027–1061
8. Duch, W.: Similarity based methods: a general framework for classification, approximation and association. Control and Cybernetics **29** (2000) 937–968
9. Duch, W., Adamczak, R., Diercksen, G.: Classification, association and pattern completion using neural similarity based methods. Applied Mathemathics and Computer Science **10** (2000) 101–120
10. Duch, W., Jankowski, N.: Survey of neural transfer functions. Neural Computing Surveys **2** (1999) 163–213
11. Duch, W., Jankowski, N.: Transfer functions: hidden possibilities for better neural networks. In: 9th European Symposium on Artificial Neural Networks, Brusells, Belgium, De-facto publications (2001) 81–94
12. Shakhnarovish, G., Darrell, T., Eds., P.I.: Nearest-Neighbor Methods in Learning and Vision. MIT Press (2005)
13. Kordos, M., Duch, W.: Variable Step Search MLP Training Method. International Journal of Information Technology and Intelligent Computing **1** (2006) 45–56
14. Asuncion, A., Newman, D.: UCI repository of machine learning databases (2007)
15. Duch, W., Adamczak, R., Grąbczewski, K.: A new methodology of extraction, optimization and application of crisp and fuzzy logical rules. IEEE Transactions on Neural Networks **12** (2001) 277–306
16. Muselli, M.: Sequential constructive techniques. In Leondes, C., ed.: Optimization Techniques, vol. 2 of Neural Network Systems, Techniques and Applications. Academic Press, San Diego, CA (1998) 81–144