ON NEURAL NETWORK CLASSIFIERS WITH SUPERVISED TRAINING

Marius Kloetzer and Octavian Pastravanu

Department of Automatic Control and Industrial Informatics
Technical University "Gh. Asachi" of Iasi
Blvd. Mangeron 53A, Iasi, 700050, Romania
Phone / Fax: +4(0)-0232-230751
E-mail: kmarius@ delta.ac.tuiasi.ro, opastrav@delta.ac.tuiasi.ro

Abstract: A study on classification capability of neural networks is presented, considering two types of architectures with supervised training, namely Multilayer Perceptron (MLP) and Radial-Basis Function (RBF). To illustrate the classifiers' construction, we have chosen a problem that occurs in real-life experiments, when one needs to distinguish between overlapping and Gaussian distributed classes. An amply commented comparative study is elaborated between MLP- and RBF-type classifiers, in order to reveal advantages and disadvantages encountered when the two types of neural network architectures are used.

Key words: classification, decision boundary, neural networks, supervised training

INTRODUCTION

The approximation capability of some neural network topologies made them a popular tool for nonlinear system identification and classification tasks. A classification task can be regarded as a process where each presented input (pattern) is assigned to one of a predefined number of classes (categories).

The purpose of this paper is to illustrate the usage of the (Multilayer Perceptron) MLP and (Radial-Basis Function) RBF in pattern classification and to elaborate a comparative analysis of their efficiency, relying on some relevant case studies. The MLP and RBF networks are feed-forward architectures which are trained in a supervised manner and they are both topologies capable of universal approximation (Cybenko 1989; Park et al. 1991).

The design of a neural network classifier is made in two distinct steps, namely the training session and the usage as a classifier. The classification performed by a feed-forward neural network can be regarded as a feature extraction – performed by the neurons placed in the hidden layers – followed by a classification performed by the output neurons.

The organization of the material presented in the paper corresponds to the following plan. The exposition starts with the systematic construction of neural classifiers (second section). It continues with the case studies which consider as classification task the separation between Gaussian distributed overlapping classes, for which the probability of correct classification using the Bayessian classifier can be estimated (Haykin 1999). The case studies take into account various training parameters and topologies of the networks, including redundancy on the network's output layer, in order to develop a comparative study between MLP- and RBF- type classifiers (fourth section). The quality of the neural classifiers is interpreted from the point of view of the training time and from the point of view of the probability of correct classification.

The organization of the presented material can be regarded as containing the steps to be taken in order to design a neural classifier suitable to the classification tasks required by real-life experiments.

THEORETICAL PRELIMINARIES

The neural architectures used in the two steps involved in the design of a classifier – the training session and the usage as a classifier – are slightly different. The neural architecture used in the first step is a standard two-layer topology, either MLP or RBF, but in the second step a new layer must be added (Duda et al. 1973; Fukunaga 1990; Richard et al. 1987).

During the training session of a neural classifier a set of input vectors (which are relevant for a certain experiment) is presented to the network along with their corresponding categories, each class being coded in a binary mode (usually combinations of a positive value, namely 1, and a negative value, namely -1).

In order to exploit the trained network as a classifier, a new layer with bipolar *step activation functions* must be added at the output of the standard MLP and RBF topologies. The number of perceptrons in the new layer

is equal to the number of outputs of the trained network. The architecture of the obtained neural classifier is presented in figure 1.

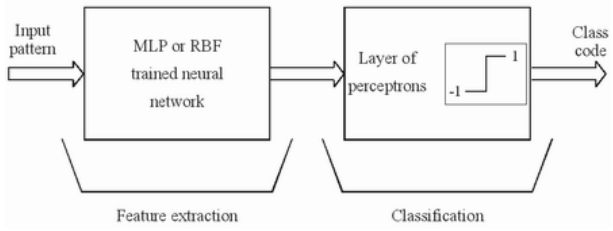


Figure 1: Modular architecture of a neural classifier

The two layers of the trained MLP or RBF network extract a feature from the presented input, and the added output layer transposes the feature vector into a code corresponding to one of the predefined classes. In order to obtain a correct mapping, the weight matrix of the new layer will be equal to the unity matrix, and the perceptrons will have zero biases.

ILLUSTRATIVE CASE STUDIES

In order to study the quality of the MLP- and RBF- type neural classifiers, the following classification task is considered. The objective is to distinguish between two overlapping equiprobable classes. Each class contains two-dimensional Gaussian distributed patterns, the first class having the mean vector $\boldsymbol{m}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and the standard deviation $s_1 = 1$, and the second class having the mean vector $\boldsymbol{m}2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$ and the standard deviation $s_2 = 2$.

The optimum (Bayesian) *decision boundary* for this kind of problem (Lippmann 1987; Haykin 1999) is a circle with the center located at:

$$\boldsymbol{x}_{\mathrm{B}} = \frac{s_2^2 \boldsymbol{m}_1 - s_1^2 \boldsymbol{m}_2}{s_2^2 - s_1^2} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \qquad (1)$$

and with the radius:

$$r_{\mathrm{B}} = \sqrt{\frac{s_1^2 s_2^2}{s_2^2 - s_1^2}\left[\frac{\|\boldsymbol{m}_2 - \boldsymbol{m}_1\|^2}{s_2^2 - s_1^2} + 4\ln\left(\frac{s_2}{s_1}\right)\right]} \approx 3{,}42 \qquad (2)$$

Figure 2 presents the Bayesian decision boundary and a set of 500 vectors from each class, plotted by using different symbols. The probability of correct classification of the Bayes classifier was estimated by performing a computer experiment which takes into account a large number of vectors from each class (namely 1 million), the obtained value being $p_{\mathrm{B}} \tilde{} 93.73\%$.

The following two subsections illustrate the development of MLP- and RBF- type classifiers, respectively, for which the probability of correct classification is estimated by considering a set of vectors "unseen" during the training phase. Both the *time necessary for training* the network (t_{train}) and the comparison between the *probability of correct classification* ($p_{classif}$) and p_{B} reflect the quality of a neural classifier.

Due to the random nature of the input patterns, all the numerical results to be given represent the mean of a series of 5 experiments.

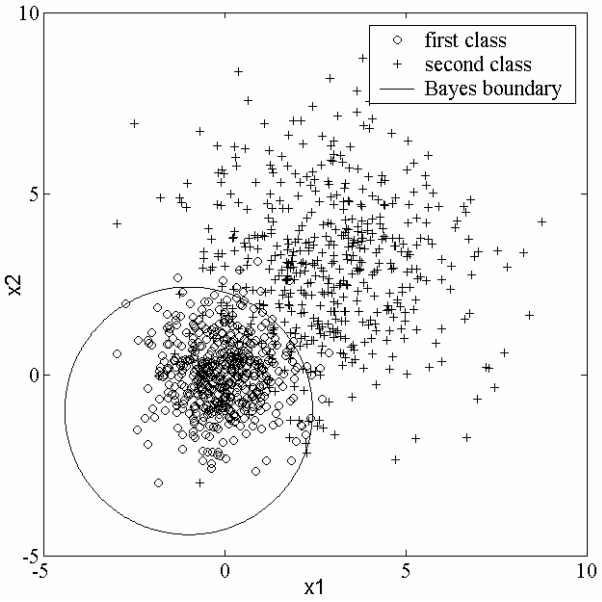


Figure 2: The distribution of the two classes and the Bayesian decision boundary

Usage of an MLP-Type Classifier

The examples presented in this subsection illustrate the dependence of the quality of the MLP classifier on the network architecture and the training parameters.

The MLP architecture to be used in classification tasks is a two layer feed-forward neural network, with *transfer (activation) functions* of the input and output layer respectively tf_1 and tf_2. There are four different MLP topologies recommended in literature, the differences between them consisting in the transfer functions, as follows:

a) tf_1 = tansigmoid function, tf_2 = tansigmoid function;
b) tf_1 = tansigmoid function, tf_2 = linear function;
c) tf_1 = tansigmoid function, tf_2 = logsigmoid function;
d) tf_1 = logsigmoid function, tf_2 = logsigmoid function.

Table 1 presents the results obtained by using the four MLP topologies, each network having two input neurons and one output neuron and the same training parameters.

obtained results	MLP topology			
	a)	b)	c)	d)
t_{train} (s)	16.7	14.6	15.5	15.3
$p_{classif}$ (%)	89.97	92.81	82.31	79.25

Table 1: Results obtained using four different MLP topologies

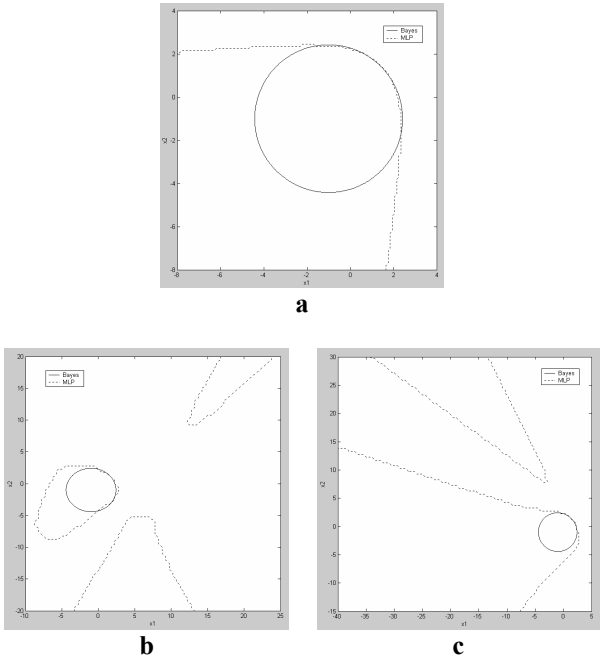
The best results in this case were obtained by using the MLP classifier with tansigmoidal neurons in the first layer and with linear nodes in the output layer. For this reason, all the experiments to be presented in this subsection will use this architecture.

The design of a neural classifier which presents redundancy in the output layer can be a useful option in some cases. For the proposed classification task, this would be the case of an MLP network with more than one neuron in the output layer. The following results were obtained by using a network with two output linear neurons (and two input tansigmoidal neurons):

- Probability of correct classification, $p_{classif}$ = 92.62 %;
- Probability to detect an error (security), p_{detect} = 96.64 %;
- Training time, t_{train} = 84.8 s.

The usage of such a redundant architecture leads to a more secure classifier, the probability of correct classification being close to the one of a non-redundant topology; the disadvantage is the increased time necessary for training the network.

A large set of experiments was performed in order to study the dependence of the quality of an MLP classifier on the number of input neurons and on the number of epochs used in training; the obtained results are presented in the next section. In the case of varying the number of input neurons it is interesting to observe the position of the decision boundary of the MLP classifier versus the Bayesian one (figure 3).



a **b** **c**

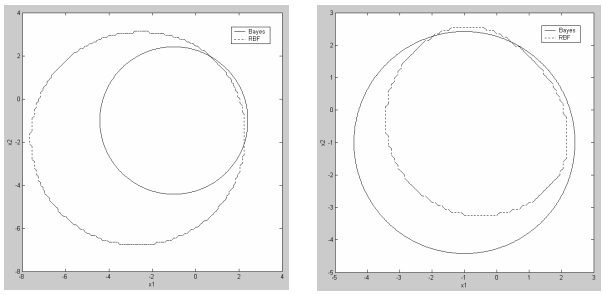
Figure 3: The boundary of the MLP classifier (dashed line) versus the Bayesian boundary (solid line):
a. MLP network with 2 input neurons;
b. MLP network with 8 input neurons;
c. MLP network with 10 input neurons.

Usage of an RBF-Type Classifier

The standard topology of an RBF neural network exhibits, on the first layer, a collection of nodes with a Gaussian-type transfer function, the second layer consisting of linear neurons.

The design of an RBF neural network can be understood as a curve-fitting problem in a high-dimensional space; the learning process is equivalent to finding a surface that provides the best fit to the training data, according to the desired accuracy; the spread of radial basis functions determines the smoothness of the approximation. The training algorithm adds neurons to the input layer of the network until the specified mean squared error goal is met.

Figure 4 reveals the decision boundary of an RBF network versus the Bayesian one, when using the RBF architecture as a classifier for the previously mentioned task. Unlike the MLP case, the boundary imposed by an RBF classifier has nearly the same shape, regardless the number of the radial neurons.



a **b**

Figure 4: The boundary of the RBF classifier (dashed line) versus the Bayesian boundary (solid line):
a. RBF network with 2 input neurons;
b. RBF network with 5 input neurons

As for the MLP architectures, the influence of training parameters of an RBF network on the quality of classification is presented in the next section.

COMPARISON BETWEEN MLP- AND RBF-TYPE CLASSIFIERS

MLP and RBF networks are both universal approximators, property which can be used, as shown, in classification tasks. For this reason it is worth to develop a comparative study with regard to their exploitation as classifiers. Therefore, RBF and MLP classifiers have been constructed for the same classification task, namely that one considered in the previous section.

All the results commented below have been obtained by *batch training*, considering that each class is known by 500 patterns. To ensure the relevance of the comparisons, all the training conditions of MLP networks have used a unique value for the error goal (namely 0) and a unique value (namely 1) to initialize weights and biases. Interest will first focus on the main

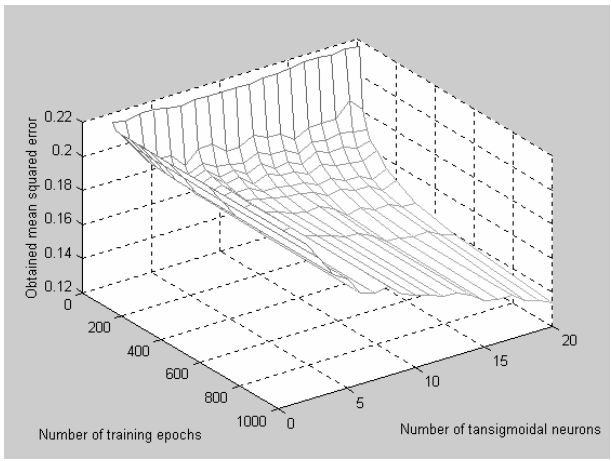
features of training procedures, and, afterwards, emphasis is placed on the quality of the designed classifiers. All the simulation experiments were conducted under Neural Network Toolbox provided in Matlab software (The MathWorks Inc. 2001)

Comments on Network Training

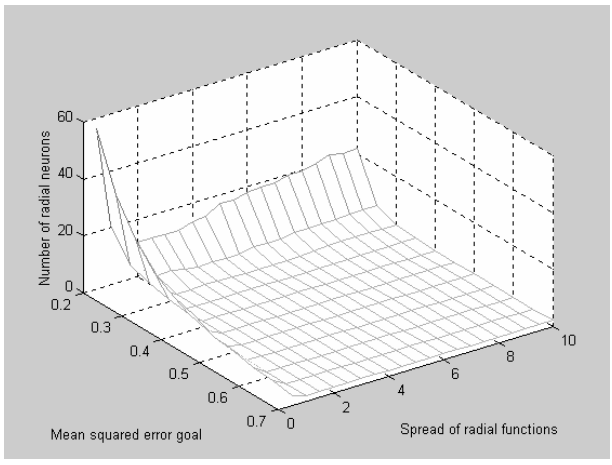
The results of network training are analyzed for various conditions used in the learning process. Thus, for MLP topology, the *achieved mean squared error* is regarded as the result of training (which depends on two key parameters: *the number of training epochs* and *the number of tansigmoidal neurons*). For RBF architecture, the *number of radial neurons* is regarded as the result of training (which depends on two key parameters: *the mean squared error goal* and *the spread of radial functions*). This point of view in understanding the role of the training conditions allows a comprehensive interpretation based on the three dimensional plots given in figure 5 (a – for MLP network, b – for RBF network).

From figure 5.a, one can observe that for the MLP classifier the result of training (obtained mean squared error) is influenced in the first instance by the number of sigmoidal neurons, for a medium number of epochs (greater than 200).

In order to compare the computation requested by the training of MLP- and RBF-type classifiers, figure 6 displays the dependence of the training time on the same training parameters previously considered. The numerical values of training times make sense only if all experiments are conducted on the same computer and in similar conditions. For MLP network, the training time increases when the number of sigmoidal neurons and/or the number of training epoch increase (figure 6.a). The plots in figures 5.b and 6.b have nearly the same shape because the time necessary to train an RBF network depends on the number of radial neurons to be added in order to obtain the desired error. Beside the probability of correct classification, the training time reflects the quality of the constructed classifier, which is discussed in the next subsection.

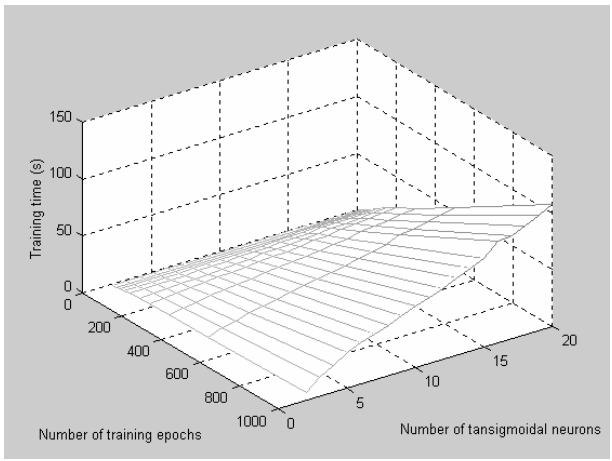


a

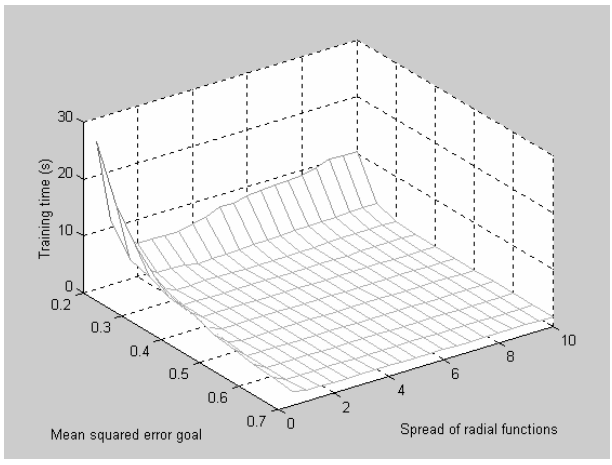


b

Figure 5: Graphical interpretation of the key elements characterizing the network training:
a. achieved mean squared error of MLP network depending on number of epochs and number of sigmoidal neurons;
b. number of input neurons of RBF network depending on mean squared error goal and spread of radial functions.



a

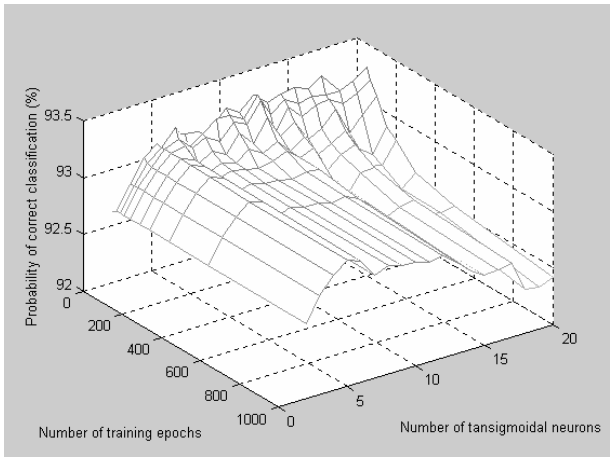


b

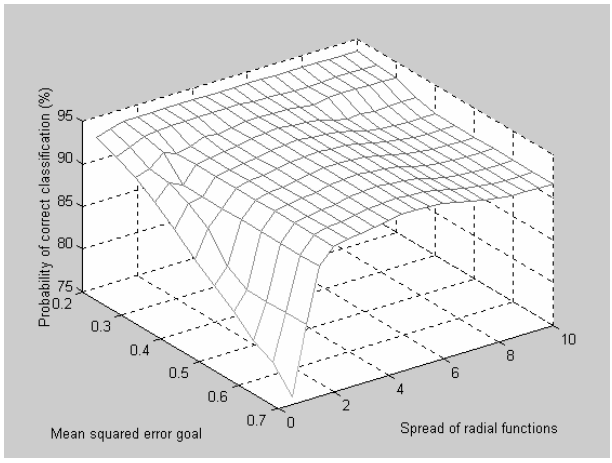
Figure 6: Dependence of the training time on the training parameters:
a. number of epochs and number of sigmoidal neurons for MLP network;
b. mean squared error goal and spread of radial functions for RBF network.

Comments on Classification Quality

The probability of correct classification of the trained classifiers is evaluated in terms of simulation results obtained for input vectors which have not been presented to the network during the training session. A graphical interpretation of these results can be given along the same lines as in the previous subsection, by considering the three-dimensional plots depicted in figure 7. According to our current interest, the surfaces plotted in these figures reflect the dependence on the *training conditions* (investigated in the previous subsection) of the classifier main quality, expressed (for both RBF and MLP networks) as the *probability of correct classification resulting from simulation*. The plots given in figure 7, together with those presented in figure 6, allow comparing the quality of the MLP- and RBF-type classifiers from the point of view of the probability of correct classification and of the training time, respectively.



a



b

Figure 7: Graphical interpretation of the neural classifiers' quality depending on the training parameters:
a. probability of correct classification of MLP network depending on number of epochs and number of tansigmoidal neurons;
b. probability of correct classification of RBF network depending on mean squared error goal and spread of radial functions.

The direct visual examination of figure 7.a shows that the quality of MLP classifier decreases by using a large number of training epochs, fact which suggests that the size of the training data set is too small for the network to get a correct generalization. In accordance with figures 5.a and 7.a, it is worth noticing that the smaller achieved error in training does not guarantee that the obtained classifier is the best. Anyway, the quality of the MLP classifier is not significantly affected by the training parameters, the difference between minimum and maximum probability of correct classification being smaller than 1%.

By using the RBF architecture, one can obtain a better probability of correct classification, but an inadequate choice of training parameters of the radial basis network can lead to bad performances of this classifier. Moreover, by comparing the plots in figures 5.b and 7.b, one can observe that an RBF classifier with many input neurons can have nearly the same quality as an RBF classifier with a significantly smaller number of neurons. All the remarks referring to MLP- and RBF- type classifiers are actually founded on precise numerical information, used for constructing the graphical plots.

When using a large set of training data is available, the MLP network generalizes well, and the result of an accurate training is usually able to provide trustable information about the quality of the obtained classifier. The situation is presented in figure 8, using 5000 vectors from each class in order to train the network. As expected, the training time is approximately ten times bigger than in the previous case. Unfortunately, in real-life experiments a large set of training data it might be difficult to obtain or improper to use because of the required computation time.

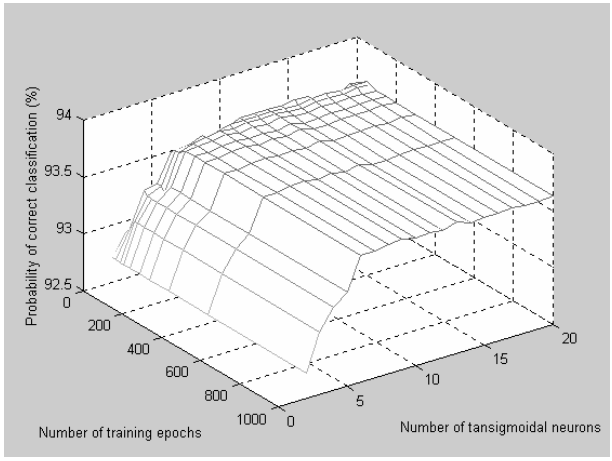


Figure 8: Probability of correct classification of an MLP network (trained using a large set of data) depending on number of epochs and number of tansigmoidal neurons

The remarks issuing from the presented comparison can be compared with the conclusions of a study involving the usage of MLP- and RBF- architectures in identification tasks (Kloetzer et al. 2001, 2002).

CONCLUSIONS

The purpose of this paper was to illustrate the usage of supervised trained neural networks in classification tasks. The problem chosen in order to construct neural classifiers reflects a situation that can occur in real-life experiments, namely the need to distinguish between overlapping and Gaussian distributed classes. The experiments were conducted in a manner that makes possible the development of an amply commented comparative study between MLP- and RBF- type classifiers. The MLP network provides a good quality and its performance is not much influenced by the fine tuning of the training parameters, making this type of classifier the optimum tool for less experienced users. It is possible to obtain a better RBF classifier (from the point of view of complexity and probability of correct classification), but this requires either many tests or a wide experience.

REFERENCES

Cybenko, G., 1989, "Approximation by superpositions of a sigmoidal function", *Mathematics of Control, Signals and Systems*, vol. 2, pp. 303 – 314.

Duda, R.O. and Hart, P.E., 1973, *Pattern Classification and Scene Analysis*, New York: Wiley.

Fukunaga, K., 1990, *Statistical Pattern Recognition*, 2nd Edition, New York: Academic Press.

Haykin, S., 1999, *Neural Networks. A Comprehensive Foundation*, 2nd Edition, New Jersey: Prentice Hall.

Kloetzer, M., Ardelean, D. and Pastravanu, O., 2001, "Developing Simulink tools for teaching neural-net-based identification", *Med'01: The 9th Mediterranean Conference on Control and Automation*, pp. 63 (abstract), paper on CD-ROM.

Kloetzer, M., Ardelean, D. and Pastravanu, O., 2002, "Crearea unei biblioteci Simulink pentru exploatarea retelelor neuronale în identificare", *Revista Româna de Informatica si Automatica*, vol. 12, no. 2, pp. 53 – 64.

Lippmann, R.P., 1987, "An introduction to computing with neural nets", *IEEE ASSP Magazine*, vol. 4, pp. 4 – 22.

Park, J. and Sandberg, I.W., 1991, "Universal approximation using radial-basis-function networks", *Neural Computation*, vol. 3, pp. 246 – 257.

Richard, M.D. and Lippmann, R.P., 1987, "Neural network classifiers estimate Bayesian a posteriori probabilities", *Neural Computation*, vol. 3, pp. 461 – 483.

* * *, The MathWorks Inc., 2001, *Neural Network Toolbox 4.0.1*, MATLAB 6.1 (Release 12.1).