© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Data Mining III*, A Zanasi, CA Brebbia, NFF Ebecken & P Melli (Editors).
ISBN 1-85312-925-9

Intrinsic characterization and scoring of classes

H-G. Zimmermann, Ch. Tietz & R. Grothmann
Siemens AG, Corporate Technology, Munich, Germany

Abstract

The paper is about classification and scoring by neural networks. Especially in marketing applications, classification tasks are often high dimensional and only small data sets are available for the estimation of the model parameters. Further on, the classes are often unbalanced and can not be separated easily from each other. Since the class characteristics are often incomplete, the handling of missing data is also a critical issue.

For these problems, we employ an autoassociator neural network, which incorporates prior knowledge about the classification task. We propose to transfer the classification into a scoring task. The reconstruction error of the autoassociator serves as a membership measure for e. g. the buyers or non-buyers in a marketing application. Within this framework, a so-called 'square' layer with a quadratic squashing function puts us in a better position to model localized structures in the data, so-called cleaning noise allows to construct a replacement for missing values in the data set.

The usefulness of our neural network architectures is illustrated by a real-world scoring application in which we separate those persons from a database who show the greatest potential to buy a new product.

1 Introduction

Classifi cation tasks arise in diverse data mining applications, such as credit scoring in fi nancial services, customer segmentation in marketing research, or fraud detection in telecommunication services [1, 2]. The spectrum of classifi cation techniques ranges from decision trees, logistic regression, genetic algorithms, or neural networks to various statistical methods [3].

Unfortunately, most of these algorithms suffer from some serious problems which come along with real-world classifi cation tasks [1]. These diffi culties are:

High dimensional spaces: Typically, real-world classifi cation problems are high dimensional and only small data sets are available for the estimation of the model parameters. The high dimensional space and the inherent sparseness of the data points, is a serious drawback for density estimation.

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Data Mining III*, A Zanasi, CA Brebbia, NFF Ebecken & P Melli (Editors).
ISBN 1-85312-925-9

Unbalanced classes: Especially in marketing applications, such as buyer and non-buyer classifi cation, the training data for the different classes are extremely unbalanced. Typically, we have far more data samples from the "non-buyer" class than from the class of "buyer" to learn the parameters of our model. As a consequence, the learning may ignore the characteristics of the class which is represented with only a few data samples. This problem becomes even more serious if the data is covered with noise.

Inseparable classes: Another well-known problem of real-world applications is that the classes can not be separated from each other. In this case, one has to use a density estimation approach rather than a geometric separation.

Missing data: Various real-world classifi cation problems involve incomplete data samples, i. e. missing components in the input vectors. Further on, the data may be covered with noise. The question arises, how to treat missing information and how to handle noisy data.

In this paper we introduce an autoassociator neural network architecture, which incorporates prior knowledge about the classifi cation task, in order to handle the outlined diffi culties. Dealing with customer classifi cation, we propose to formulate the classifi cation task in form of a scoring rather than a density estimation problem. Here, we refer to the reconstruction error of the autoassociator neural network as a membership measure for a particular class of customers, such as buyers of a certain product. The prior knowledge is integrated into the neural network in form of architectural and algorithmic enhancements. For instance, we introduce a 'square' layer with a quadratic squashing function in the decompression phase of the autoassociator network in order to model localized structures in the data. Further on, a technique called cleaning noise allows us to handle missing data. As a result, the scoring is improved by supporting higher compression levels of the autoassociator network. We demonstrate the usefulness of our neural network architecture by its application to a real-world classifi cation task of sorting out potential buyers out of a large customer database.

The organization of the paper is as follows: In sec. 2 we introduce a basic autoassociator network and formulate the associated scoring problem. Further on, we incorporate a square layer into the network in order to model localized data structures, discuss proper ways of preprocessing the raw customer data, and focus on cleaning noise for the handling of missing data. In sec. 3 we deal with an empirical study in which we extract those persons from a database who are potential buyers. Conclusions and an outlook on future research follow in sec. 4.

2 Classification & scoring by autoassociator neural networks

The goal of customer classifi cation task is to decide whether a person is a potential buyer of a new product or not. Typically, such an analysis is based on the previous spending behavior of the individual and other personal data, such as age, gender, job or family status. As it can be seen from this outline, the design of a customer classifi cation model involves two steps: First, we develop an autoassociator neural network to identify the characteristics of a particular set of persons, such as buyers or non-buyers. Second, we forecast the behavior of other persons on the latter basis

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Data Mining III*, A Zanasi, CA Brebbia, NFF Ebecken & P Melli (Editors).
ISBN 1-85312-925-9

by defi ning a score value. The higher the score value, the higher is the probability that an analyzed person is a potential buyer.

2.1 Scoring by autoassociator neural networks

Fig. 1 depicts a basic neural network architecture which solves the outlined tasks of customer characterization and scoring.

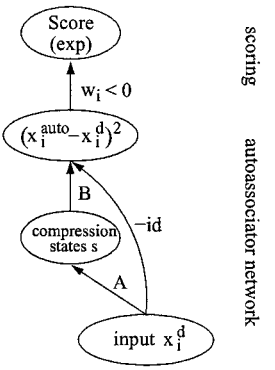


Figure 1: Scoring customers by an autoassociator neural network approach.

The network architecture of Fig. 1 is composed of two parts: The lower part of the architecture consists of an autoassociator neural network which used to identify the characteristics of a particular set of persons (e. g. buyers *or* non-buyers). The upper part is responsible for the estimation of the scoring value of each person.

Suppose, that we have a certain set of customers who are buyers of a new product. These customers should have identical characteristics concerning their e. g. previous spending behavior or other personal data. Persons who share these characteristics are likely to be potential customers as well, while persons who show a different behavior are most likely non-buyers of the new product.

We identify the major characteristics of the customer set by a coordinate transformation. Such a coordinate transformation can be realized by a neural network in form of a so-called autoassociator (bottleneck) neural network [8]. Inputs x_i^d to the autoassociator network are the customer records including personal data or other detailed information about the customers' behavior. Although the compression states of the bootleneck have no direct interpretation, the bottleneck is used to fi lter major characteristics the customers have in common. The dimensionality of the internal compression state $dim(s)$ is chosen smaller than the dimensionality of the input vector x ($dim(s) < dim(x)$). The compressor matrix A identifi es the characteristics of the customers from the data records, while the reconstruction of the customer data records is done by the decompresser matrix B. The compression state is designed as an hidden layer with hyperbolic tangent squashing [6].

We transfer the customer data records with a fi xed negative of an identity matrix $-id$ to the output layer of the bottleneck. Thus, the output layer of the autoassociator network provides us with the *reconstruction error* $(x_i^{\text{auto}} - x_i^d)^2$ of each customer data record. Since our goal is a perfect reconstruction of the customer

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Data Mining III*, A Zanasi, CA Brebbia, NFF Ebecken & P Melli (Editors).
ISBN 1-85312-925-9

data, the target values of the output layer are fixed at zero. We use the squared errors as cost function [6, 8].

The reconstruction error $(x_i^{\text{auto}} - x_i^d)^2$ of the autoassociator neural network serves as a membership measure for customers who share important characteristics. From this it follows, that the reconstruction error of potential customers should also be small. On the other hand, the characteristics of persons who show a different behavior should lead to large reconstruction errors. Based on these assumptions, we define a scoring value, which indicates the probability of a particular person to be a potential customer. The scoring value ranges between 0 and 1, with 0 denoting a non-buyer and 1 marking a potential buyer respectively.

The calculation of the score value for each person is as follows: The reconstruction error of the autoassociator network is fed into an output layer, which computes the scoring value. The weights w_i of the associated connector are limited to negative values ($w_i < 0$). We use the exponential function as activation function for the 'score' output layer. In combination with the negative weights w_i the exponential squashing yields a proper definition of our scoring value: Large reconstruction errors result in low scoring values, while small errors are associated with high scores. The target values of the 'score' output layer correspond to the actual scores of the customers. We employ the absolute value cost function [6].

Most remarkable, this approach can also be utilized if only data records of *one* class (e. g. buyers) are available. In this case, we propose the following *four* steps:

1. Train the autoassociator network for different compression levels $dim(s)$ until convergence. Freeze matrices A and B.
2. Train the scoring weights $w_i < 0$. By definition, all target values of the 'score' output layer are equal to 1. The scoring weights $w_i < 0$ should be initialized with weight values of -1. We stop the training if some of the weights w_i drop below -0.5. This prevents the trivial solution of all scoring weights w_i being (nearly) equal to zero. According to our experiments, this procedure improves the scoring by removing unreliable inputs.
3. Identify the optimal compression level $dim(s)$ by a distribution analysis of the scoring abilities of the different networks.
4. Compare the results with those of randomly generated customer data.

The outlined procedure directly addresses the problem of unbalanced classes. Since misleading features are sorted out during the learning of the scoring weights, the identification of potential customers is enforced and the scoring is reevaluated.

2.2 Local data modeling by autoassociators with extended decompression

Depending on the global and local segmentation properties of the customer data, we propose to extend the decompression phase of the autoassociator network with a square layer. The resulting neural network architecture is depicted in Fig. 2.

The functioning of the extended neural network architecture of Fig. 2 is similar to the basic autoassociator network of Fig. 1. However, the additional 'square' hidden layer incorporating a quadratic squashing function puts us in a better position to model localized data: The square layer in the decompression phase of autoassociator includes a similarity or discrimination analysis into the network. By this

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Data Mining III*, A Zanasi, CA Brebbia, NFF Ebecken & P Melli (Editors).
ISBN 1-85312-925-9

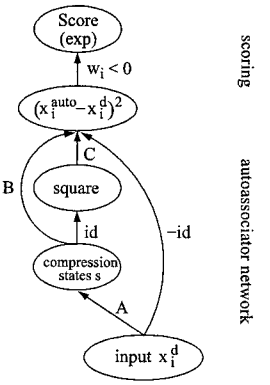


Figure 2: Scoring by an autoassociator network with extended decompression.

we combine the global segmentation properties of multi-layer perceptrons with the local segmentation properties of RBF networks. Further details on the properties of the square layer can be found in Neuneier et al. (1998) [6]. In our experiments we found that a square layer in the compression phase is *not* necessary.

2.3 Preprocessing of raw customer data records

The preprocessing of the raw customer data records is of importance, because the data records are essential for the characterization of e. g. buyers and non-buyers. The raw data records may contain e. g. information about the customers personal and demographic status as well as data about their previous spending behavior. Examples of such input signals are the customers age, marital status, information about their residence as well as specifi cations of the products the customers already use. As it can be seen from these examples, the data records typically consist of *heterogeneous* data types: Besides cardinal input signals, one has to deal with ordinal, nominal or bi-nominal raw input signals. After many experiments with real world customer data, we have settled with the simple transformations of Tab. 1 for the different types of input signals.

Table 1: Preprocessing functions for different types of raw input data.

type of raw input	**preprocessing transformation (pseudo code)**
cardinal input x	INPUT = scale(x)
nominal input x	INPUT = IF ($x = i$) THEN 1 ELSE 0 $\forall i = 1, ..., n > 2$
ordinal input x	INPUT = scale(x) INPUT = IF ($x = i$) THEN 1 ELSE 0 $\forall i = 1, ..., n > 2$
bi-nominal input $x \in \{a, b\}; p(a) \leq p(b)$	INPUT = IF ($x = a$) THEN 1 ELSE $-p(a)$ / $p(b)$

As shown in Tab. 1, we propose a relative simple preprocessing. Since the measurement unit of a cardinal input may affect the analysis, cardinal inputs are scaled

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Data Mining III*, A Zanasi, CA Brebbia, NFF Ebecken & P Melli (Editors).
ISBN 1-85312-925-9

such that the transformed input has a mean value of 0 and a variance of 1. In case of a nominal input, we suggest to create a new binary variable for each of the n nominal states. An ordinal variable can either be discrete or continuous. Depending on its characteristic, such an input can either be treated like a cardinal or a nominal input. Concerning bi-nominal variables, we generate a new variable for each state a, b by taking into account the distribution $p(a)$ and $p(b)$ of the raw states a and b.

Finally note, that the handling of missing values should be done before the preprocessing transformations are calculated. In case of cardinal and continuous ordinal inputs we suggest to replace missing values by the sample mean. Missing values in nominal and bi-nominal variables should be handled such that the resulting preprocessed input value of a missing value is always zero. For instance, missing values of nominal and bi-nominal variables could be coded as -999.

2.4 Handling missing data by cleaning noise

In classifi cation tasks, one typically has to deal with missing values, outliers or noise. As a remedy, we propose *cleaning* and the combination of cleaning with an adaptive noise term (*cleaning noise*). Cleaning computes an individual correction term for each training pattern. The adaptive noise procedure generates a potentially asymmetric and dependent noise distribution which also covers the observed data. Both techniques are explained in the following. A more comprehensive approach to the training of neural networks can be found in Neuneier et al. (1998) [6].

2.4.1 Cleaning reviewed

Training neural networks, one typically assumes that the input data is noise-free and thus, forces the network to fi t the data exactly. However, this assumption is often violated due to corrupt data. Cleaning considers the input data as being corrupted by noise whose distribution has to be learned. Individual correction terms are calculated for each training pattern during the cleaning (Fig. 3).

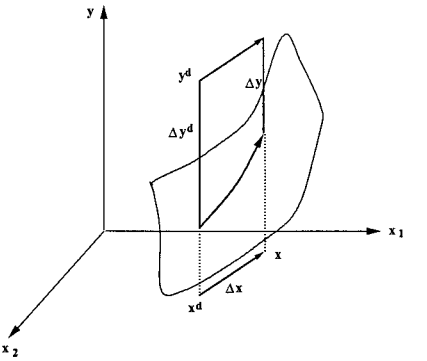


Figure 3: Cleaning. A small shift in the input data decreases the output error.

The cleaning error function for a data pattern i is given by the sum of two terms assuming identical variance levels for input and output:

$$E_i^{y,x} = \frac{1}{2}\left[\left(y_i - y_i^d\right)^2 + \left(x_i - x_i^d\right)^2\right] = E_i^y + E_i^x \qquad (1)$$

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Data Mining III*, A Zanasi, CA Brebbia, NFF Ebecken & P Melli (Editors).
ISBN 1-85312-925-9

with x_i^d, y_i^d as the observed data point. Learning with standard backpropagation [4], the network output $y(x_i, w)$ determines the weight adaptation as usual,

$$w^{(\tau+1)} = w^{(\tau)} - \eta \frac{\partial E^y}{\partial w} . \tag{2}$$

We also must memorize correction vectors Δx_i for all input data of the training set in order to present the cleaned input x_i to the network,

$$x_i = x_i^d + \Delta x_i . \tag{3}$$

The update rule for the corrections, initialized with $\Delta x_i^{(0)} = 0$ can be derived from typical adaptation sequences $x_i^{(\tau+1)} = x_i^{(\tau)} - \eta \frac{\partial E^{y,x}}{\partial x}$ leading to

$$\Delta x_i^{(\tau+1)} = (1-\eta)\Delta x_i^{(\tau)} - \eta(y_i - y_i^d)\frac{\partial y}{\partial x} . \tag{4}$$

All of the necessary quantities, i. e. $(y_t - y_t^d)\frac{\partial y(x,w)}{\partial x}$ are computed by standard backpropagation anyway [4]. We found that the algorithms work well if the same learning rate η is used for both the weight and cleaning updates.

Although the network may learn an optimal model for the cleaned data, there is no easy way to work with cleaned data on a test set because for this data we do not know the output target difference for computing eq. 4. Hence, the model is evaluated on a test set with a different noise characteristic compared to the training set. On this problem, we propose a combination of learning with noise and cleaning.

2.4.2 Cleaning noise reviewed

Typically, training with noisy inputs involves taking a data point and adding a random variable drawn from a fixed or adaptive distribution. This new data point x_i is used as an input to the network. If we assume, that the data is corrupted by outliers and other influences, it is preferable to add the noise term to the cleaned input. For the case of Gaussian noise the resulting new input is:

$$x_i = x_i^d + \Delta x_i + \xi\phi, \tag{5}$$

with ϕ drawn from the normal distribution. The cleaning of the data leads to a corrected mean of the data and therefore to a more symmetric noise distribution, which also covers the observed data x_i.

We propose a variant which allows more complicated and problem dependent noise distributions:

$$x_i = x_i^d + \Delta x_i - \Delta x_k, \tag{6}$$

where k is a random number drawn from the indices of the memorized correction vectors $[\Delta x_i]_{i=1,\ldots,N}$. By this, we exploit the distribution properties of the correction vectors in order to generate a possibly asymmetric and/or dependent noise distribution, which still covers $x_i = x_i^d$ if $k = i$.

One might wonder why we want to disturb the cleaned input $x_i^d + \Delta x_i$ with an additional noisy term Δx_k. The reason for this is that we want to benefit from

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Data Mining III*, A Zanasi, CA Brebbia, NFF Ebecken & P Melli (Editors).
ISBN 1-85312-925-9

representing the whole input distribution to the network instead of only using one particular realization. This approach supplies a solution to the cleaning problem when switching from the training set to a test set.

We propose to employ the cleaning and the cleaning noise procedure right from the beginning of the neural network training. Given the structural design of the network, cleaning noise allows to construct a replacement for missing values in the customer data records. In our experiments, we found that cleaning noise supports higher autoassociator compression levels and improves the scoring.

3 Empirical study

The customer classifi cation problem which is presented in this section has been carried out as a feasibility study for a German publishing company. The goal of the study is to identify those people from a database who are likely to respond to a mailing (in terms of buying a product or not). The feasibility study is based on the data records of 45.000 persons. A data record includes 54 data fi elds, which contain personal data and information about the previous spending behavior.

On this classifi cation task, we apply the neural networks and learning techniques outlined in the previous section. For the learning of the neural network parameters we selected 4.000 customers out of the database, i. e. the autoassociator networks are trained by using only customer data. The evaluation of the neural networks is performed on the remaining set of data records (approx. 41.000 persons). The preprocessing of the raw data records is in accordance with the suggestions of sec. 2.3. The models are trained until convergence using standard backpropagation and the *vario-eta* learning rule [8, 4]. In our experiments we found that an autoassociator compression level of $dim(s) = 27$ yields proper classifi cation results. Due to space limitations, we only present the results of selected models.

Fig. 4 compares the scoring error distribution of *two* autoassociator neural networks on the test set. The underlying neural network architecture of both models is given by the standard autoassociator network of Fig. 1. The only difference between the networks is that we applied cleaning noise only to one model.

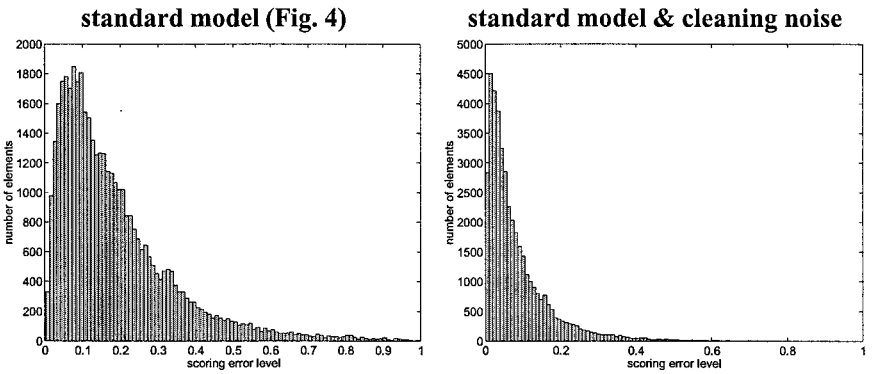


Figure 4: Scoring error distribution of autoassociator networks (test set).

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Data Mining III*, A Zanasi, CA Brebbia, NFF Ebecken & P Melli (Editors).
ISBN 1-85312-925-9

As shown in Fig. 4, cleaning noise enables us to improve the scoring accuracy. The scoring error of the standard model incorporating cleaning noise is defi nitely lower than those of the standard model without cleaning noise. Hence, we may conclude that cleaning noise supports the training of the neural network [6].

The performance of the different autoassociator networks is measured by calculating the so-called lift [1]. The lift is a common performance measure in classifi cation tasks. The lift measures "the change in the concentration of a particular class when the model is used to select a purposefully biased sample from the general population" [1]. According to Berry et. al (1996) [1], the lift is defi ned as

$$\text{lift} = \frac{P(\text{class}|\text{sample})}{P(\text{class}|\text{population})} . \tag{7}$$

The higher the lift value of a particular model, the better is its classifi cation performance [1]. The lift values calculated for our autoassociator neural networks are shown in Tab. 2. All computations are performed on the test set.

Table 2: Evaluating the lift values of different scoring models.

Percent of top scores	Standard model (Fig. 1)	Optimized scoring weights	Standard model & cleaning noise	Standard model & cleaning
5%	26.30%	21.25%	24.38%	30.63%
10%	15.31%	15.00%	18.75%	22.81%
20%	14.06%	13.75%	14.84%	18.44%
30%	14.58%	14.48%	15.11%	15.31%
50%	11.44%	11.12%	12.00%	12.38%

As can be seen from Tab. 2, the standard autoassociator networks (see Fig. 1) with cleaning and cleaning noise are able to achieve the best classifi cation performance. Compared to the performance of the standard model, the results re-indicate that cleaning (noise) supports the learning of the networks. The results outlined here suggest that one could increase of the overall effi ciency of the mailing by the application of our autoassociator neural networks. From this point of view, the application of our models to other classifi cation problems, such as credit scoring in fi nancial services, seems to be promising.

4 Conclusions and future research

We developed autoassociator neural networks for customer classifi cation and scoring problems. The classifi cation task is transferred into a scoring problem. The reconstruction error of an autoassociator network serves as a membership measure to separate potential buyers from non-buyers. Most remarkably, the inclusion of prior knowledge into the neural networks by architectural and algorithmic enhancements allows us to handle common diffi culties of real-world classifi cation tasks. For instance, the application of cleaning (noise) supports the handling of

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Data Mining III*, A Zanasi, CA Brebbia, NFF Ebecken & P Melli (Editors).
ISBN 1-85312-925-9

missing values in the customer data records, while a square layer enables us to model localized structures in the data. We applied our approach to a real-world classifi cation problem of a German publishing company. The results are promising and suggest the transfer of the modeling framework to other applications, such as credit scoring in fi nancial services.

We have lots of extensions and further analyses under consideration. For instance, one may apply input pruning techniques in order to perform a feature selection which supports the classifi cation. Furthermore, one may apply recurrent neural networks to solve classifi cation and scoring tasks [7, 8].

References

[1] Berry M. and Linoff G.: *Data Mining Techniques For marketing, Sales and Customer Support.* John Willey & Sons, Inc, 1996.

[2] Berson S., Thearling K. and Smith S. J.: *Building Data Mining Applications for CRM*, McGraw-Hill Professional Publishing 1999.

[3] Chen M., Han J. and Yu P.: *Data mining: An overview from database perspective.* IEEE Transactions on Knowledge and Data Eng., Vol. 8, No. 6, Dec. 1996, p. 866 - 883.

[4] Haykin S.: *Neural Networks. A Comprehensive Foundation.* Macmillan College Publishing, New York, 1994. Second edition 1998.

[5] Mueller A., Zimmermann H. G. and Neuneier R.: *Robust Generation of Symbolic Prosody by a Neural Classifier Based on Autoassociators.* in: Proceedings Int. Conf. on Acoustic Speech and Signal Processing (ICASSP), 2000.

[6] Neuneier R. and Zimmermann H. G.: *How to Train Neural Networks.* in: *Neural Networks: Tricks of the Trade*, Springer, Berlin, 1998, p. 373 - 423.

[7] Zimmermann, H.G. and Neuneier, R.: *Neural Network Architectures for the Modeling of Dynamical Systems*, in: A Field Guide to Dynamical Recurrent Networks, Eds. Kolen, J.F., Kremer, St., IEEE Press 2001.

[8] Zimmermann H. G., Neuneier R. and Grothmann R.: *Modeling of Dynamical Systems by Error Correction Neural Networks*, in: Modeling and Forecasting Financial Data, Techniques of Nonlinear Dynamics, Eds. Soofi , A. and Cao, L., Kluwer Academic Publishers, 2002.