Raport Badawczy
Research Report

RB/32/2001

Neural networks foundations for traffic forecasting

M. Krawczak

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences

POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 8373578

fax: (+48) (22) 8372772

Pracę zgłosił: prof. dr inż. Roman Kulikowski

Warszawa 2001

Neural networks foundations for traffic forecasting

Maciej Krawczak*

Abstract

Most observational disciplines, such as traffic, try to infer properties of an unfamiliar system from the analysis of a measured data of its behavior. There are many highly developed techniques associated with traditional time series analysis. During the last decade, several new and innovative approaches have appeared, such as neural networks and time-delay embedding, which can give insights not available with those standard methods - however the realization of this promise is still difficult.

1. Introduction

In advanced management of traffic systems it is a crucial task to predict the future traffic conditions. A short-term prediction can help in improve smoothness, reduce crowd, change travel conditions. Models of traffic can be used for estimating the future travelling conditions and duration of them. Then drivers can be informed by e.g. radio about expected times of travelling, and about possibilities to avoid jams.

In general, traffic management systems use historical as well as current data for traffic control collected by inductive loop detectors. The data describe flows of cars in selected segments of roads. Here we will not consider any technical aspects of procedures of collecting data. After collecting and preprocessing data we are interested in extracting some knowledge involved in data in order to predict the future flows.

If a mathematical model describing a studied traffic is known, forecasting becomes a trivial task. However, if a model of the traffic is either unknown or incomplete, it is typical to attempt to forecast by building a model that takes into account only previous flows while ignoring any other exterior influence.

The collected historical data of traffic streams have a form of time series. More formally, a time series $\{x(t)\}$ can be defined as a function x of an independent variable t. Its main characteristic is that its future behavior can not be predicted exactly as in the case of a known deterministic function of t. However, the behavior of a time series can sometimes be anticipated by describing the series through probabilistic laws. Time series prediction

* Systems Research Institute of the Polish Academy of Sciences,
University of Information Technology and Management, ul. Newelska 6, 01-447 Warsaw, Poland.

problems are approached either from a stochastic perspective [krawczak] or, more recently from a neural network perspective [2, 3]. Each of these approaches has advantages and disadvantages: the stochastic methods are usually fast, but of limited applicability since they commonly employ linear models, whereas the neural networks methods are powerful enough, but the selection of an appropriate architecture and parameters is a time consuming trial and error procedure.

There are at least two reasons that might make neural networks very attractive for modeling time series: 1) The theoretical work shows that neural networks are powerful enough to uniformly approximate almost any arbitrary continuous function on a compact domain [4] similar to traditional universal approximation techniques based on Taylor function expansion, Fourier series, etc. However, neural networks can effectively construct approximations for unknown nonlinear functions by learning from examples (known outcomes of the function). The neural networks models can be complemented with other successful approximation techniques based on wavelets, kernel estimators, nearest neighbors, hinging hyperplanes, regression, partial least squares and fuzzy models [5]; 2) There is a direct relationship between the basic stochastic models for time series and neural networks models.

Our study explores:

- the possibility of designing an appropriate neural networks for time series prediction based on information obtained from stochastic modeling. Such an analysis could provide some initial knowledge regarding an appropriate data sampling rate and neural network architecture, as well as regarding the choice of initial neural network parameters;
- some tools from the chaos theory, mostly based on Takens' theorem [7] that can help to estimate the dimension m of the manifold from which the time series originated can be used to construct a neural networks model using $(2m + 1)$ external inputs [8].

The approach is to perform an initial stochastic analysis of the data and to choose an appropriate neural networks architecture, and possibly initial values for the neural networks parameters according to the most adequate linear model. This idea is supported by the fact that many non-linear systems can be described fairly well by linear models and for such systems it is a good idea to use insights from the best linear model to select the regression for the neural networks model. The motivation for this approach is that the linear stochastic modeling is much more cost effective than the selection of a neural networks architecture through a trial and error procedure. Since information is obtained from a linear model, for more complex problems the neural network might be over-dimensioned (similar performance could be obtained using a smaller net and less learning examples). However, the exhaustive trial and error procedure involved for determining such an optimal network could be costlier than the stochastic analysis.

2. Neural networks approximations of stochastic models

Stationary time series can be described time invariant characteristic parameters. This makes them very attractive in practice, since it implies that they could be represented by time invariant models. The first step in stochastic modeling is thus an attempt to stationarize the studied time series through a suitable discrete differentiation pre-processing, as described in [11].

A general linear stochastic model of a stationary time series is the autoregressive moving average model of orders p and q (ARMA(p, q). It describes the future value as a weighted sum of p previous values and the current as well as q previous values of a random process. Formally, a stationary ARMA(p, q) process with zero mean $\{x(t)\}$ is represented as:

$$x(t)=\varphi_1 x(t-1)+\varphi_2 x(t-2)+\cdots+\varphi_p x(t-p)+\alpha_t+\psi_1\alpha_{t-1}\psi_2\alpha_{t-2}+\cdots+\psi_q\alpha_{t-q}, \quad (1)$$

where $x(t-1), x(t-2),..., x(t-p)$ represent the process values at p previous time steps, α_t, α_{t-1}, ..., α_{t-q} are the current and the q previous values of a random process, usually emanating from a normal (Gaussian) distribution with mean zero and $\varphi_1... \varphi_p$, $\psi_1... \psi_q$ are the model parameters.

The ARMA(p, q)-based predictor approximates the real process value $x(t)$ by a predicted value $\hat{x}(t)$, computed as:

$$\hat{x}(t)=\varphi_1 x(t-1)+\varphi_2 x(t-2)+\cdots+\varphi_p x(t-2)+\alpha_t+\psi_1\alpha_{t-1}\psi_2\alpha_{t-2}+\cdots+\psi_q\alpha_{t-q}, \quad (2)$$

The error between the real process value $x(t)$ and the predicted value $\hat{x}(t)$ is the *residual* α_t.

The AR(p) and MA(q) models are special cases of the ARMA(p, q) model, where AR(p) is described as:

$$x(t)=\varphi_1 x(t-1)+\varphi_2 x(t-2)+\cdots+\varphi_p x(t-p)+\alpha_t, \quad (3)$$

and MA(q) is described as:

$$x(t)=\alpha_t+\psi_1\alpha_{t-1}\psi_2\alpha_{t-2}+\cdots+\psi_q\alpha_{t-q}, \quad (4)$$

A natural generalization of the linear ARMA and AR models to the nonlinear cases leads to the NARMA model

$$x(t)=h(x(t-1), x(t-2),\cdots, x(t-p), \alpha_{t-1}, \alpha_{t-2}, \alpha_{t-q})+\alpha_t, \quad (5)$$

and the NAR model

$$x(t)=h(x(t-1), x(t-2),\cdots, x(t-p))+\alpha_t, \quad (6)$$

where h is an unknown smooth function.

The NARMA and NAR models are very complex, thus being unsuitable for real life applications. Fortunately, they are closely related to more practical nonlinear models, the neural networks.

The neural network architecture considered in this paper consists of a number of relatively simple processing units called neurons. The neurons are grouped in layers, neurons in adjacent layers are interconnected through weights. Three different layer types can be distinguished: input layer - the layer that external stimuli are applied to, output layer - the layer that outputs results to the exterior world, and one or more hidden layers - intermediate computational layers between input and output layer. The neural network architectures

considered here are either feedforward, in which the signal flow is from input layer towards output layer, or recurrent, in which the feedforward signal flow is supplemented with additional feedback connections from output to input layer. Each neuron applies an activation function (usually a nonlinear smooth and bounded function) to the sum of its weighted inputs and a neuron specific parameter (called bias).

Feedforward Recurrent and NNs have been proposed [12, 2] for simulating NARMA and NAR models respectively. An invertible [1] NARMA-based predictor can be approximated as:

$$\hat{x}_t = h(x_{t-1},\cdots,x_{t-p},\alpha_{t-1},\cdots,\alpha_{t-q}) \approx \sum_{i=1}^{m} W_i f\left(\sum_{j=1}^{p} w_{ij} x_{t-j} + \sum_{j=1}^{q} w'_{ij}\,(x_{t-j} - \hat{x}_{t-j}) + \theta_i\right) + \Gamma, \qquad (7)$$

where f represents a nonlinear, smooth and bounded function and $\alpha_k = x(k) - \hat{x}(k)$, for all $k \in \{t-q,\cdots,t-1\}$. This approximation of the NARMA-based model corresponds to the recurrent neural network from Fig. 1, in which W_i; are the weights between hidden and output neurons, w_{ij} are the weights between external input neurons and hidden neurons; w'_{ij} are the weights between context input neurons and hidden neurons, θ_i are the hidden neuron biases, Γ is the bias and f is the activation function of the neurons. Similarly, a NAR-based predictor can be approximated as:

$$\hat{x}(t) = h(x(t-1),\cdots,x(t-p)) \approx \sum_{i=1}^{m} W_i f\left(\sum_{j=1}^{p} w_{ij} x(t-j) + \theta_i\right) + \Gamma, \qquad (8)$$

obtained by disconnecting the context inputs $\alpha_{t-1},\ldots,\alpha_{t-q}$ from Fig. 1.

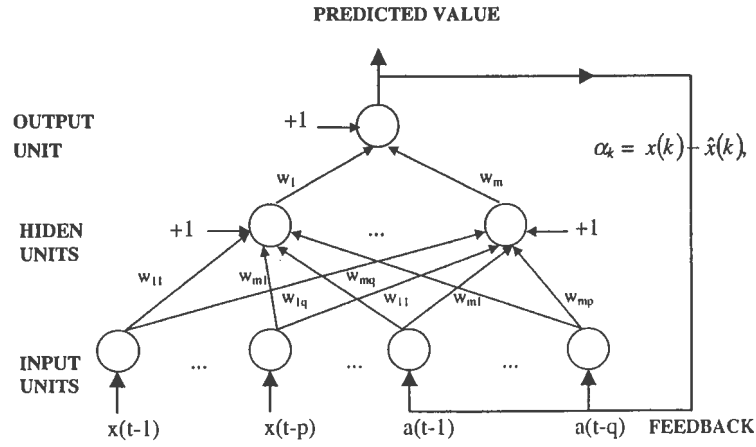

Fig. 1. Stochastic Model Approximation.

3. Chaos theory and time series prediction

It is assumed that an interesting system is observed namely the output of a traffic system is available and observed. The second assumption is such that, generally, the reasons of traffic system performance is unknown - it means the inputs to the system are not available. The aim is to analyze the system and to forecast the future behavior of it.

The main difficulties in analyzing the real world time series and related traffic systems are due to presence of nonlinearity and representation of only finite observations - these kind of difficulties justifies the use of empirical as well as nonparametric methods dealt in this paper.

For last few years much attention has been paid to the application of the chaos theory for analysis of different time dependent real systems. At the same time much interest has been given to the ability of modern computational tools such as neural networks and genetic programming.

3.1. Observation of Linear Systems

For long time in the system theory linear dynamic systems (very often time-invariant) were investigated very deeply. Let us consider a linear time invariant system defined by state-space equations:

$$\dot{x}(t) = Ax(t) + bu(t) \tag{9}$$
$$y(t) = hx(t) + dv(t) \tag{10}$$

The state vector of the system x is of the N dimension; $y(t)$ is the observed output of the system; $u(t)$ is the driving force (control or noise) applied to the system; $v(t)$ is considered as noise; the matrix A and vectors b, h, d are parameters of the system. Having the system parameters as well as the value of the state $x(t)$ it is easy to predict $y(t)$.

Now, let us consider a discrete version of the system (9)-(10), taking samples of the output at times Δt:

$$x(k+1) = \Phi x(k) \tag{11}$$
$$y(k) = hx(k) \tag{12}$$

in these equations $k=n\Delta t$ and $\Phi = \exp(A\Delta t)$. There is a question when and if $y(k)$ can determine corresponding state $x(k)$, in other words there is a question of observability of the system. For linear systems like (11)-(12) we define a so-called observability matrix:

$$O(h, \Phi) = \begin{bmatrix} h \\ h\Phi \\ . \\ h\Phi^{N-1} \end{bmatrix} \Phi x(k) \tag{13}$$

Putting

$$y(k+n)=h\Phi^{n}x(k) \tag{14}$$

we can describe an observation vector

$$[y(k), y(k+1),..., y(k+N-1)]=O(h,\Phi)x(k) \tag{15}$$

In order to find the state of the system the observability matrix $O(.)$ must be invertible.

The linear autoregression which exactly models the observed sequence has a form

$$y(k)=h\Phi^{N}O^{-1}[(k-N)\cdots y(k-1)+dv(t)]^{T}. \tag{16}$$

Now let us consider a nonlinear system, which are much more useful because most of observed time series represent nonlinear dynamic systems. In a similar way we can write a nonlinear system:

$$\dot{x}(t)=F(x(t)) \tag{17}$$
$$y(t)=h(x(t)) \tag{18}$$

where x is N dimensional state of the system, F is a nonlinear transition function, h is an observation function. Applying sampling intervals equal to Δt we can write a delay vector

$$[y(k-1), y(k-2),..., y(k-T)] \tag{19}$$

Takens theorem (1981) allows to apply the observability problem of nonlinear systems. Due to this theorem which states that under very mild conditions, if

$$T>2D+1 \tag{20}$$

(where D is the fractal dimension), then there exists (for almost all smooth function h) an one-to-one differentiable mapping Ψ between the delay vector and the state vector $x(k)$

$$\Psi(y(k-1), y(k-2),..., y(k-T))=x(k) \tag{21}$$

In this way we can write an autoregression which models the time series in the form

$$y(k)=h\circ F\circ\Psi(y(k-1), y(k-2),..., y(k-T)). \tag{22}$$

It is worth to notice that equation (21) has a similar form as equation (15). Such a form allows to apply nonlinear autoregressions to model time series.

3.2. Neural Networks Prediction

Feedforward neural networks work as an universal approximators. Before using such a network it must be trained. For an input $x(k)$ the network response (or output) is given as

$$y(k) = N(w, x(k)) \tag{23}$$

where weights w are parameters of the network. During the learning process we try to minimize the average squared error (a learning error)

$$\min_{w} \frac{1}{P} \sum_{p=1}^{P} [d(k) - N(w, x(k))]^2 \tag{24}$$

counting over a training set of examples, where $d(k)$ is the desired pattern for each input. Under some stationary and ergodic conditions the learning error (23) converges to an expectation:

$$\lim_{P \to \infty} \min_{w} \frac{1}{P} \sum_{p=1}^{P} [d(k) - N(w, x(k))]^2 \to E\|D - N(w, X)\|^2. \tag{25}$$

Here D and X are considered as random variables while the expectation is taken over the joint probability distribution.

The problem of finding optimal parameters w^* for linear systems is trivial, eg. Shanmugan (1988). For nonlinear systems, the problem is much more complex, especially using neural networks. Namely, we must use the universal approximator properties of feedforward neural networks. The formal explanation of using least square errors for neural network training can be found e.g. in Hecht-Nielsen (1990).

3.3. Chaotic series

Many interesting systems in the real world are known to be nonlinear or chaotic. Up till now by some mathematical expressions the analysis is constrained to extraction of similar tendencies. For example let us analyze one of the most famous and oldest, perhaps, equation modelling a population growth. The equation is called the logistic equation and is described by the following form (discrete case)

$$x(t+1) = bx(t)(1 - x(t)), \quad t = 0,...,300 \tag{26}$$

where b is a real value parameter. The logic equation is drawn in Fig. 2 for the parameter $b = 3.0$, and first 200 points $t = 1,...,200$, and for the initial condition $x(0) = 0.1$,

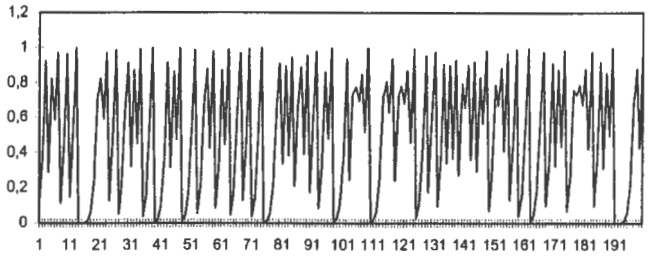

Fig. 2. The first 200 points of the logistic equation.

Another example of chaotic series is generated by the Mackey-Glass equation (1977) of the form:

$$x(t+1) = bx(t) + a\frac{x(t-s)}{1+x^c(t-s)} \tag{27}$$

where the parameters can be stated as follows: $a = 0.2$, $b = 0.9$, $c = 10$, $s = 18$, and the initial conditions are assumed to be $x(0) = x(1) = ... = x(18) = 0.7$. Time evolution of the Mackey-Glass equation for the above parameters is shown in the Fig. 3

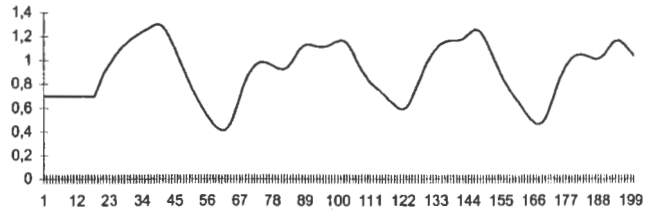

Fig. 3. The first 200 points of the Mackey-Glass equation.

The above time series are shown in order to show that simple nonlinear equations with feedback can cause very complex behavior of the plots. It will be shown little later the attractors of complex time series.

Embedding There is a pretty simple method for analyzing time series, the method is called the *time series embedding*, Ruelle (1981) and Takens (1981). The approach can be illustrated by plotting pairs of point $x(t)$ and $x(t+1)$ for the considered functions. The case of the logistic equation is shown in the Fig. 4

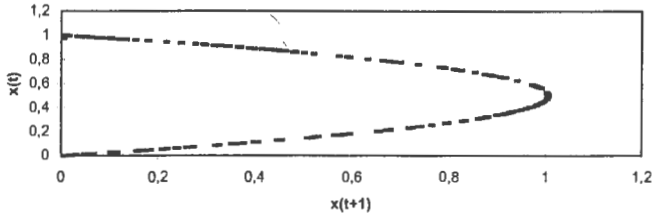

Fig. 4. Embedded logistic function.

In the case of the Mackey-Glass equation the embedding plot is shown in Fig. 5:

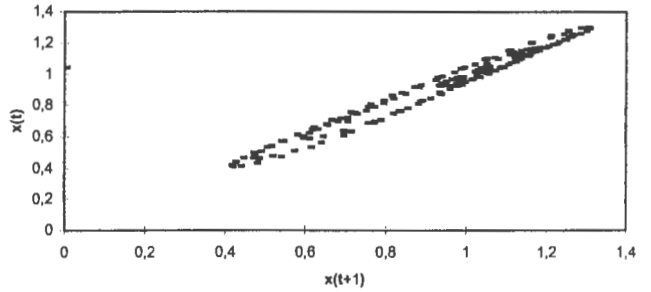

Fig. 5. Embedded Mackey-Glass function.

It should be emphasizes that even plots versus time are very complex time series than the respective embedding plots are rather simple as well as known patterns. The idea is such, namely given the point $x(t)$ it is easy to make a very good estimation of the next point $x(t+1)$ by interpolation. The similar principle can be extended to multiple dimensions:

$$X(t) = x(t), x(d+t), x(2d+t), x(3d+1), \ldots, x(nd+t) \tag{28}$$

where X is the embedded vector, d is the *separation*, and n is the *embedding dimension*.

It has been shown (Takens, 1981) that for a given chaotic series embedded properly there exists a smooth function. This function of embedding can be approximated in various ways, e.g. or by neural networks. Proper parameters that is the dimension parameter d as well as the embedding function must be found empirically. Latter we will discuss difficulties appearing during the modelling process.

3.4. Elements of the empirical chaos theory

Lyapunov exponents The main feature of chaotic systems is their high sensitivity to initial conditions. There is a way to distinguish this feature, namely by calculating the Lyapunov

exponents (Wolff, Swift, Swinney, 1985). These exponents indicate whether succeeding points laying on an attractor diverge or converge - with passing time.

The examined trajectories on the attractor are embedded in a space. The divergence between two trajectories can be measured as a difference between two n-tuples. At the beginning there is a need to define the so-called dominant average Lyapunov exponent as:

$$L = \log_2 \frac{\sum_{n=1} \frac{l_{n+1}}{l_n}}{n-1} \tag{29}$$

where n denotes the index of a sample, l is the Euclidean distance between two neighboring trajectories. There is also possible to calculate local Lyapunov exponents, for that we need samples of trajectories of attractor which should be dense. It is obvious that the rate of divergence is not constant at all along the attractor.

It is interesting to allays the Lyapunov exponents. If the Lyapunov exponents are positive it means the system is a chaotic one, while the negative exponents indicate the system behavior is reverting, the value zero of the exponents characterizing cyclic behavior of the systems. For instance, an attractor of a sinusoidal system is a circle.

Hurst exponent There is a very important measure of predictability of states of time series. This measure is named the Hurst exponent which is derived by application of so-called R/S analysis. Considering a time series X representing by n points, and choosing a number p which can be taken for convenience as $10 \le p < n/2$, then the data can be divided into n/p blocks. For any block the average is calculated, then next the maximum range of each block as well as the standard deviation of each block is calculated. The value=range/standard_deviation is calculated for any block and next average of the block is calculated. The average value rs is in relation with the Hurst exponent in the following way

$$rs = (p/2)^H \tag{30}$$

where H is the Hurst exponent.

The Hurst exponent values are between 0 and 1. We can distinguish two ranges of the exponent. A value $0.5 < H < 1$ indicates so called persistent behavior, it means a system can be considered as the values moves to one as a predictable system. While a value $0 < H < 0.5$ indicates probabilistic systems. For $H = 0$ time series must change direction every sample, for $H = 0.5$ time series moves as a random walk, while for $H = 1$ a system is a purely deterministic.

There is a relationship between one definition of the fractal dimension and the Hurst exponent, that is following expression

$$D = 2 - H .$$

Analysis and prediction of chaotic time series requires finding the above mentioned parameters - this is another difficult task. We can distinguish two class of methods: the first empirical and the second analytical one. Within the first group we must form a model of the attractor and a supervised learning algorithm of some kind is required (e.g. algorithms for learning feedforward neural networks or genetic algorithm). Within the analytical methods we

must base our consideration on the Takens theorem (1981) for determining the upper bounds of an embedding parameter (if we know the fractal dimension of the attractor). There are several methods for deriving a choice of embedding dimension, e.g. Tong (1990), Wolff, Swiftt, Swinney, (1985).

4. Conclusions

In this paper we summarize modern foundations of application neural networks for prediction of time series. Here feedforward neural networks which are universal approximators are used as a tool for modeling unknown nonlinear functions.

We have considered linear systems and associated state observability or autoregression. In a similar way we described nonlinear systems. Next we showed feedforward neural networks as universal approximators. Nonlinear systems, even very simple, can generate in some sense unstable solutions - these kind of systems are called chaotic systems. The main parameters of such systems are described and the role of neural networks as a tool for modelling.

We have studied neural networks as models for time series forecasting, and our research compares the Box-Jenkins method against the neural network method for long and short term memory series. Our work was inspired by previously published works that yielded inconsistent results about comparative performance. We have since experimented with many time series of differing complexity using neural networks. The performance of the neural networks is compared with that of the Box-Jenkins method. Our experiments indicate that for time series with long memory, both method produced comparable results. However, for series with short memory, neural networks outperformed the Box-Jenkins model. Because neural networks can be easily built for multiple-step-ahead forecasting, they present a better long term forecast model than the Box-Jenkins method. We discussed the representation ability, the model building process and the applicability of the neural net approach. Neural networks appear to provide a promising alternative for time series forecasting.

References

[1] Abu-Mostafa, Y., 1995, "Financial Applications of Learning from Hints," *in Advances in Neural Information Processing Systems* (G. Tesauro et al, editors), Vol. 7, pp. 411-418.

[2] Baker, L. and J. P. Gollub (1990): Chaotic Dynamics, an Introduction. Cambridge Univ. Press.

[3] Bowerman, B. L. and O'connell, R. T. 1987. *Time Series Forecasting: Unified Concepts and Computer Implementation, 2nd* ed. Duxbury Press, Boston, TX.

[4] Box G. and G. Jenkins, 1976, „Time Series Analysis. Forecasting and Control", Prentice Hall.

[5] Cybenko G., 1989, „Approximation by Superpositions of Sigmoidal Function", *Mathematics of Control, Signal, and Systems,* Vol. 2, pp. 303-314.

[6] Dreyfus, S. E. (1990): Artificial Neural Networks, Back Propagation, and the Kelley-Bryson Gradient Procedure. Journal of Guidance, vol. 13, no. 5, 926.

[7] Drossu R. and Z. Obradoviæ, 1997, „Efficient Design of Neural Networks for Time Series Prediction", School of Electrical Engineering and Computer Science, Washington State University, Pullman, Working Paper, 99164-2752.
[8] Dutta, S. and Shekhar, S.1988. Bond rating: A non-conservative application of neural networks. *In International Joint Conference on Neural Networks,* volume 2, pages 443-450.
[9] Fishwick P. A.1989. Neural network models in simulation: A comparison with traditional modeling approaches. *In Proceedings of Winter Simulation Conference,* pages 702-710, Washington, DC.
[10] Hecht-Nielsen, R. (1990): Neurocomputing. Addison Wesley.
[11] Hertz, J. A., A. Krogh, R.G. Palmer (1990): The Theory of Neural Computation.
[12] Hornik, K., M. Stinchombe and H. White (1989): Multilayer feedforward neural networks are universal approximators. Neural Networks, vol. 2, 359.
[13] Krawczak, M. (2000): Backpropagation versus dynamic programming approach. BULLETIN OF POLISH ACADEMY OF SCIENCES, SERIA TECHNICAL SCIENCES, vol. 48, no. 2.
[14] Lapedes A. and R. Farber, 1987, „Nonlinear Signal processing Using Neural Networks: Prediction and System Modeling", *Technical Report,* LA-UR87-2662, Los Alamos National Laboratory, Los Alamos, New Mexixco.
[15] Mackey, M. and L. Glass (1977):Oscillations and chaos in physiological control systems. Science, 197,287.
[16] Makridakis, S. e. 1982. The accuracy of extrapolation (time series methods: Results of a forecasting competition. *Journal of Forecasting,* l:lll-153.
[17] Moody, J. and J. Utans, 1994, "Architecture Selection Strategies for Neural Networks: Application to Corporate Bond Rating Prediction," *in Neural Networks in the Capital Markets* (A. N. Refenes, editor), John Wiley & Sons.
[18] Obradovic Z. and Chenoweth, T., 1996, "Selection of Learning Algorithms for Trading Systems Based on Biased Estimators," *Heuristics* Vol. 8, No. 5, in press.
[19] Potts and D.S. Broomhead, 1991, „Time Series Prediction with a Radial Basis Function Neural Network", *Proc. SPIE Vol. 1565, Adaptive Signal Processing,* pp. 255-266.
[20] Ruelle, D. (1981): Small random perturbations of dynamical systems and the definition of attractors. Commun Math. Phys., 137, 82.
[21] Rumelhart, D., McClelland, J., and the PDP Research Group 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* volume 1. MIT Press, Cambridge.
[22] Shanmugan, K. and A. Breipohl (1988): Random Signals: Detection, Estimation, and Data Analysis. John Wiley and Son, N-Y.
[23] Sharda, R. and Patil, R. 1990. Neural networks as forecasting experts: An empirical test. In *International Joint Conference on Neural Networks,* volume l, pages 491-494, Washington, D.C. IEEE.
[24] Sjöberg et al., 1994, „Neural Networks in System Identification", *Proc. 10th IFAC Symposium on System Identification (SYSID'94), C*openhagen, Denmark, Vol. 2, pp. 49-72.
[25] Takens T., 1981, Detecting Strange Attractors in Turbulence", *Lecture Notes in Mathematics,* D.Rand and L. Young eds., Springer, Berlin, pp. 366-381.
[26] Tang Z., et al., 1994, „Time Series Forecasting Using neural Networks vs. Box-Jenkins Methodology, *Artificial Neural Networks: Forecasting Time Series,* V.R. Vemuri and R.D. Rogers eds., IEEE Computer Society Press, pp. 20-27.

[27] Tong, H. (1990): Non-linear Time Series - A Dynamical System Approach. Oxford Univ. Press.
[28] Tong, H. and Lim, K. 1980. Threshold autoregression, limit cycle and cyclical data. *Journal of Royal Statistical Society* B, 42:245.
[29] Weigend, A. and N. Gershenfeld (eds) (1993): Time series prediction: Forecasting the future and understanding the past. Addison-Wesley.
[30] Weigend, A., Huberman, B., and Rumelhart, D. 1990. Predicting the future: A connectionist approach. Technical Report Stanford-PDP-90-Ol, Stanford University, Stanford, CA.
[31] Werbos P., 1992, „Neural Networks, System Identification and Control in the Chemical Process Industries", *Handbook of Intelligent Control, Neural, Fuzzy, and Adaptive Approaches,* D.A. White and D.A. Sofge eds., Van Nostrand Reinhold, pp. 283-356.
[32] Wolff, A., J. B. Swift, H. L. Swinney and J. A. (1985): Determining Lyapunov exponents from a time series. Physica 16D, 285.