KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 290–297

DATA PREDICTIONS USING NEURAL NETWORKS

CĂLIN ENĂCHESCU

Abstract. Financial and economic forecasters have witnessed the recent development of a number of new forecasting models. Traditionally, popular forecasting techniques include regression analysis, time-series, analysis, moving averages and smoothing methods, and numerous judgmental methods. ANN (Artificial Neural Networks) are members of a family of statistical techniques, as are flexible nonlinear regression models, discriminant models, data reduction models, and nonlinear dynamic systems. They are trainable analytic tools that attempt to mimic information processing patterns in the brain. Because they do not necessarily require assumptions about population distribution, economists, mathematicians and statisticians are increasingly using ANN for data analysis.

1. Introduction

Neural computing represents an alternative computational paradigm to the algorithmic one (based on a programmed instruction sequence). Neural computation is inspired by knowledge from neuroscience, though it does not try to be biologically realistic in details [4]. We will deal with neural networks organized in layers, where the information is transmitted from the first layer until the last layer. This type of feedforward multylayer neural networks is called MLP (*MultiLayer Perceptron*) [3]. Some important fact about artificial neural networks:

a) the first layer of neurons is called input layer, it is a simple buffer to store the input data.
b) the input signal is transmitted to the connected (hidden) neurons.
c) the last layer is the output of the system, and is usually called output layer.

MLPs learn in a supervised manner [10]. Learning represents the process in which input patterns are presented repeatedly and the weights are adjusted according to the learning algorithm, which in this supervised case, take the difference between the desired output and the current output into consideration.

2000 *Mathematics Subject Classification.* code, code.
Key words and phrases. Artificial Neural Networks, supervised learning, prediction of data series.

2. Mathematical aspects of supervised learning

The training set used for supervised learning has the following form:

$$T = \{(\mathbf{x}_i, \mathbf{z}_i) \mid i = 1, 2, \ldots, N\} \tag{1}$$

where $\mathbf{x}_i \in \mathbf{R}^n$ is the n-dimensional input vector, and $\mathbf{z}_i \in \mathbf{R}^m$ is the m-dimensional target vector that is provided by a trainer. $N \in \mathbf{N}$ is a constant that represents the number of training samples. In the classical supervised learning strategy [5], [10], the trainer is a static agent. Using the probabilistic distribution he selects a certain input vector $\mathbf{x}_i$, and provides the appropriate target vector $\mathbf{z}_i$. The learning algorithm [15] will compute the difference between the output generated by the neural network $\mathbf{y}_i$ and the desired target vector $\mathbf{z}_i$.

The signal error is used to adapt the synaptic weights w_{ji} using a gradient descendent strategy [7]: $w_{ji} = w_{ji} + \eta \dfrac{\partial E}{\partial w_{ji}}$ where $\eta \in (0, 1)$ is the learning rate, controlling the descent slope on the error surface which is corresponding to the error function E [8]: $E = \frac{1}{2} \sum\limits_{i=1}^{N} (y_i - z_i)^2$

3. Data series prediciton with neural networks

Usually, data accessible to the economists are sets of numeric values that describe the situations about the investigated problem in a certain moment. Numeric data sets are called *data series*, and their analysis and values prediction is called the *analysis* respectively the *prediction of data series*. Examples of data series are: the value of monthly unemployment, the value of monthly / annual inflation, the value of different stocks and the value of daily exchange between different currencies [1]. To develop these economic models we have to study the analysis of economic time series, and for the decision process we have to make the prediction for these series, using the developed models. An important class of economic data series is represented by financial data series. These series contain data that represent monetary values of some economic objects or reports on some monetary values of economic objects.

Data processing represents the process of data transformation before building the models. These transformations are conversions, classifications, filtrations or other similar processing. Many times data are not properly structured to build predictive models. In these cases it is important to remove components that represent redundant or irrelevant information [16].

The general purpose of preprocessing is to remove the observable deterministic relations. Theoretically, the purpose is to obtain some data series with mean 0 and a small variation. The first step in data preprocessing is to make comparable the components of the data series. For this purpose it is necessary to rescale the data so that the values of the data series components to have values in the interval

[0,1] or [-1,1]. The second step is to remove the primary deterministic components, that are easy to be observed. Examples of these types of components are trend and seasonality [12]. In the next step we can proceed a data filtering. The purpose of the filtering process is to remove non-trivial periodical components that have dominant effects in data series. To determine those periodical componets we can apply a Fourier transformation of the data series. To filter these componets we can build linear filters. The most common filters are the the low-pass filters, high-pass filters and band-pass filters [10].

Finally, input data vectors are analised to determine possible clustering [9], [13]. This is done through simple classification of data. If we can group the data in precise distinct classes, separate models are built for those classes. It is possible, that after clusterization the models to be equivalent. We must verify model equivalence, and in case of equivalence relations, we must build simplified data general models [14].

By analising the data corresponding to a problem it is possible to build predictive models. It is very important to test the accuracy of the generated model. Through validation we understand the testing of the model and the measurment of its performance using a measure of performance [6]. A method for spliting data in training data and in validation data is the simple division based on data feature. We can consider the data x_t with $t <= T_0$ as training data and data x_t with $t > T_0$ as validation data.

Other techniques for data prediction are based on the auto-regressive model, or on the average sliding model [16].

The prediction of data series is based on the supposition that there exist a functional relation between past, present and future data series values. Usually the supposition is that the functional relation is not completely deterministic, but also contains a stochastic component. In several cases, especially in the case of financial time series, it is assumed that the deterministic component is not dominant, the stochastic component being of great importance.

The performance measurement of predictive models is crucial for their practical application. A common measure regarding predictive models, including neural networks, is the use of square average error. Many times, prediction errors are too large in the context of real applications because of the stochastic components present in the financial series. For this reason, it is necessary to use additional performance measures to test the validity of the predictive models. Neural networks are efficient tools to detect nonlinear relations that rule, at least partially, the behaviour of time series. As the majority of financial data series contain nonlinear components, neural networks are good candidates to predict this type of series.

4. Practical implementation of the neural network

The practical part of this paper is related to the implementation of a neural network that offers the possibility to analyse and predict data series. In this simulation we have tried to approximate and to predict some financial data series. The parameters that are influencing the learning process are: the training data set, the number of epochs (number of the presentations of the training data set), learning rate, the activation functions for the neurons contained in the hidden layer, the number of neurons in the hidden layer. The architecture of the neural network used in our simulation is corresponding to a MLP neural network with one hidden layer [4]:

- The input layer contains n input neurons, n representing the dimensionality of the input space $x_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)}) \in \mathbf{R}^n$. The bias can be considered explicitly or implicitly;
- The hidden layer having a number of hidden neurons equal to the dimension of the training set $T = \{x_i, f(x_i)) \mid i = 1, 2, \ldots, N\}$. The activation functions of the hidden neurons are Green functions $G(x - x_k)$ [13]. The dimension of the hidden layer can be reduced using an unsupervised clustering algorithm;
- The output layer contains one single output layer having as activation function a linear function or a special weighted functions of the output values generated by the neurons in the hidden layer [6];

Synaptic weights:

- The weights between the input layer and the hidden layer are included in the form of the activation functions of the hidden neurons. The vector $w = (w_1, w_2, \ldots, w_N)$ represents the weights between the hidden layer and the output layer.

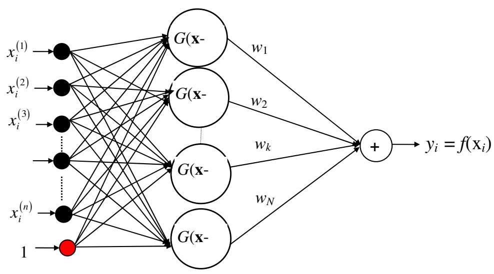

FIGURE 1. Architecture of the neural network used for simulations.

The nerural network was trained using an original learning algorithm, based on the backpropagation learning strategy [8]. For learning we have used a training set containing financial data series, like the exchange rate between RON and EUR or RON and USD. The learning set, corresponds to the time frame January 2004-June 2005, and was obtained from the official Web-site of the National Bank of

Romania http://www.bnr.ro (Banca Nationala a Romaniei). After learning, we have performed a testing phase, in order to measure the accuracy of the trained neural network.

Number of epochs	Learning error
10	0.0289682211941586
50	0.0135544478100808
100	0.00669238729756063
500	0.00268232311020435
1000	0.00192883390014049
5000	0.00163946058914474
10000	0.00157560371972362

TABLE 1. Results of the learning phase: the learning error obtained for different epochs.

In the following graphics (Fig. 2, 3, 4, 5) we have presented the ability of the neural network to approximate and to predict the finnancial data series corresponding to the exchange rate of RON versus EUR. The light-gray curve represents the real exchange rate and the dark-gray curve represents the result generated by the neural network.

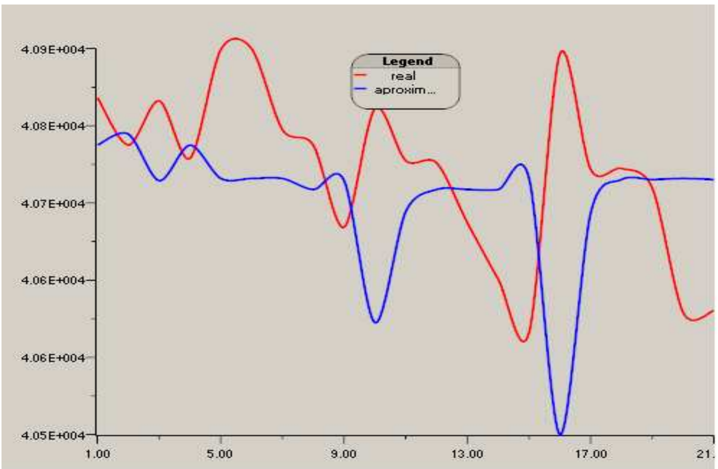

FIGURE 2. Results of the approximation and prediction made by the neural nework, after a learning process of 10 epochs.

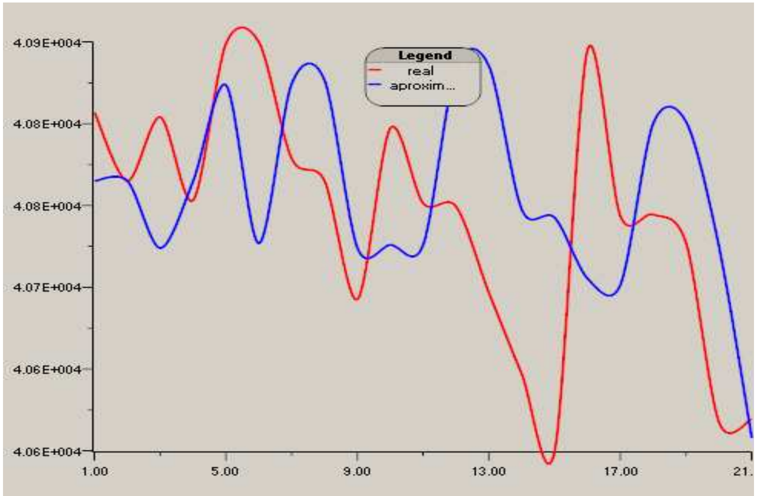

FIGURE 3. Results of the approximation and prediction made by the neural nework, after a learning process of 100 epochs.

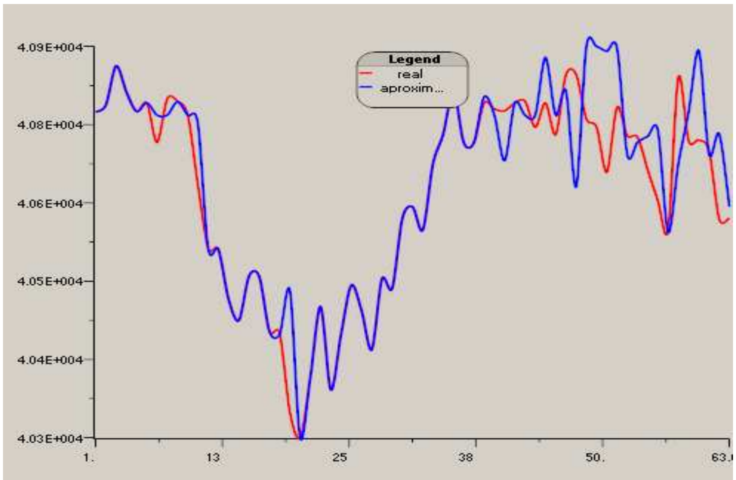

FIGURE 4. Results of the approximation and prediction made by the neural nework, after a learning process of 500 epochs.

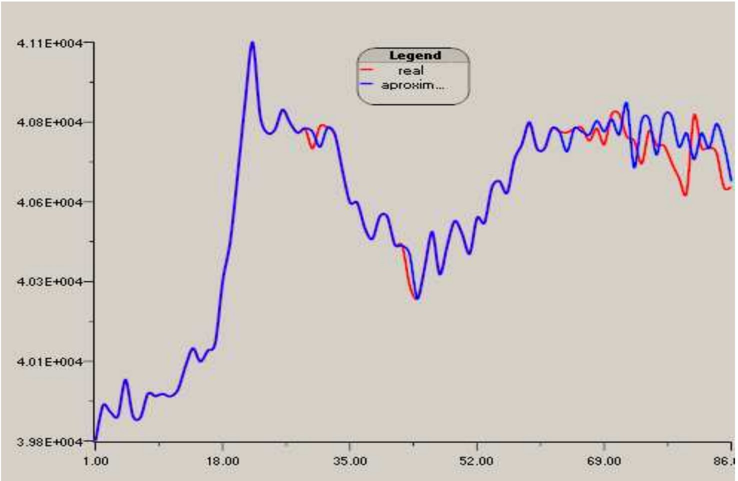

Figure 5. Results of the approximation and prediction made by the neural nework, after a learning process of 5000 epochs.

5. Conclusions

The ability to deal with many processing elements makes neural computing faster than conventional computing. In addition, parallelity makes it robust and fault-tolerant in the sense that performance does not degrade significantly even if one of the nodes fails. Researchers are concluding that most economic and financial problems are non-linear; that simple cause-and-effect relationships rarely exist; that, instead, most problems encountered are fuzzy patterns, which relate to multiple variables. There are many useful neural network models for nonlinear data analysis, such as the MLP model, and there is room for many more applications of statistics to neural networks, especially in regard to estimation criteria, optimization algorithms, confidence intervals, diagnostics, and graphical methods. As they do not require an exact specification of the functional equations, emulative neural systems can be applied to predict economic phenomena - especially unrecognized, unstructured, and non-stationary processes. Thus, ANNs (Artificial Neural Networks) are highly suitable for analyzing economic systems. ANNs have proven themselves to be adequate also for searching out and identifying non-linear relationships and for pinpointing those variables that hold the highest predictive value. After extensive training, ANN are able to eliminate substantial amounts of ambiguity in economic forecasts, although never completely overcoming indeterminacy.

References

[1] Adlar, J. Kim, Christian, R. Shelton (2002) Modeling Stock Order Flows and Learning Market-Marking from data, AIM-2002-009, MIT.

[2] András, P. (2000) Reţele neuronale pentru aproximare şi predicţia seriilor de timp, Universitatea "Babeş-Bolyai" Cluj Napoca.

[3] Enăchescu, C. (1998) Bazele teoretice ale reţelelor neuronale, Editura Casa Cărţii de Ştiinţă Cluj-Napoca.

[4] Enăchescu, C. (1997) Elemente de inteligenţă artificială. Calcul neuronal., Universitatea "Petru Maior" Trgu-Mureş, 1997.

[5] Enăchescu, C. (1995) Properties of Neural Networks Learning, 5th International Symposium on Automatic Control and Computer Science, SACCS'95, Vol.2, 273-278, Technical University "Gh. Asachi" of Iasi, Romania.

[6] Enăchescu, C.(1996) Neural Networks as approximation methods. International Conference on Approximation and Optimisation Methods, ICAOR'96, "Babes-Bolyai University", Cluj-Napoca.

[7] Enăchescu, C. (1995) Learning the Neural Networks from the Approximation Theory Perspective. Intelligent Computer Communication ICC'95 Proceedings, 184-187, Technical University of Cluj-Napoca, Romania.

[8] Hecht-Nielsen, R. (1989). Theory of the backpropagation Neural Network. IEEE International Joint Conference on Neural Networks, 1, 593-605.

[9] Girosi, F., T. Pogio (1990). Networks and the Best Approximation Property. Biological Cybernetics, 63, 169-176.

[10] Haykin, S. (1994), Neural Networks. A Comprehensive Foundation. IEEE Press, MacMillian.

[11] Hornik, K., Stinchcombe, M., White, H. (1989). Multilayer Feedforward Networks Are Universal Approximators. Neural Networks, 2, 359-366.

[12] Lackes, R. and Mack D.: Neural Networks. Basics and Applications, Springer-Verlag Berlin Heidelberg, 1998.

[13] Pogio, T., F. Girosi (1990). Networks for Approximation and Learning. Proceedings of the IEEE, Vol. 78, 9, 1481-1497.

[14] Petri, A. J. (1991). A Nonlinear Network Model for continuos learning. Neurocomputing 3, 157-176.

[15] White, D.A., (1989) Learning in artificial neural networks: A statistical perspective. Neural Computation 1, 425-464.

[16] Weigend A.: Time series prediction, Addison-Wesley, 1994.

(1)"Petru Maior" University of Tg. Mures, Nicolae Iorga street nr.1, 540088

E-mail address: `ecalin@upm.ro`