13. A NEURAL NETWORK MODEL FOR TIME-SERIES FORECASTING

Nicolae MORARIU[*]
Eugenia IANCU[**]
Sorin VLAD[***]

Abstract

The paper presents some aspects regarding the use of pattern recognition techniques and neural networks for the activity evolution diagnostication and prediction by means of a set of indicators. Starting from the indicators set there is defined a measure on the patterns set, measure representing a scalar value that characterizes the activity analyzed at each time moment. A pattern is defined by the values of the indicators set at a given time. Over the classes set obtained by means of the classification and recognition techniques is defined a relation that allows the representation of the evolution from negative evolution towards positive evolution. For the diagnostication and prediction the following tools are used: pattern recognition and multilayer perceptron. The paper also presents the REFORME software written by the authors and the results of the experiment obtained with this software for macroeconomic diagnostication and prediction during the years 2003-2010.

Key words: time-series, pattern recognition, neural networks, multilayer perceptron, diagnostication, forecasting
JEL Classification: C45, C53

1. Introduction

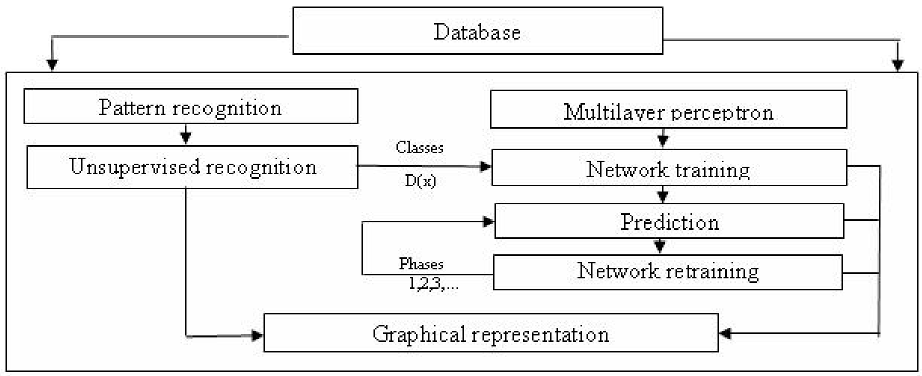
The assessments of the development level for a specific activity can be carried out by using the analysis of the evolution of the indicators describing both the quantitative level as well as the qualitative mutations in time. The problems related to diagnostication and prediction are solved using pattern recognition techniques and the multilayer perceptron, implemented by the REFORME software. The conceptual

[*] Ph.D. Lecturer, "Stefan cel Mare" University of Suceava, Economic Science and Public Administration Faculty, nmorariu@eed.usv.ro
[**] Lecturer, "Stefan cel Mare" University of Suceava, Economic Science and Public Administration Faculty, eiancu@seap.usv.ro
[***] Lecturer, "Stefan cel Mare" University of Suceava, Economic Science and Public Administration Faculty, sorinv@seap.usv.ro

diagram for diagnostication, prediction and graphical representation is presented in Figure 1.

Figure. 1

The conceptual diagram for diagnostication and prediction



The data corresponding to the evolution in time of the activity considered are processed using the methods already mentioned, methods assessing the overall evolution trend of the indicators. The output is a qualitative variable (classes or D(x)) representing the result of the assessment.

The database includes the data referring to the specific indicators and the achievements per indicators.

2. Pattern clasification and recognition techniques

2.1. Pattern recognition

If we note x_1, x_2, … , x_n the characteristics set and x is a form defined by these characteristics, then the x form can be considered as a vector $x(x_1, x_2, \dots, x_n)$. The N parameters of the vector x are subjected to different processing methods: data normalizing, linear and nonlinear transformations, reducing and pattern selection, [3], [6], [7].

If the values of the parameters are of different magnitude, then the parameters with high absolute values will have a greater influence over the classification results and the values of the parameters must have the same order of magnitude. The method frequently used is the domain adjusting method, the following transformation of the parameters values being applied:

$$x_{i,new} = \frac{\left(x_{i,old} - x_{i,min}\right)}{x_{i,max} - x_{i,min}} \tag{1}$$

If P_1, P_2,…,P_m are the reference patterns (the prototypes) and C_1,C_2…,C_m are the corresponding classes, then the minimum distance classifier will assign the input pattern X to the class C_i if the distance $d = |X - P_i|$ is minimum. The most frequently used distances are derived from the general Minkovski distance:

$$d_{Minkovski} = \left[\sum_{1}^{n} (x_i - p_i)^k \right]^{1/k} \quad (2)$$

For k=2, the Euclidean distance is obtained.

For the determination of the class to which a pattern belongs, starting from a training set with known classification, one of the most known methods based on the minimum distance principle is the nearest neighbor method. Let $F = \{f_1, f_2, ..., f_n\}$ be a set of training patterns and C_1, C_2,…,C_p the classes in which the set F was divided using a classification algorithm. The rule of the nearest neighbor can be mathematically written as:

$$if\ d(f, f_a) \le d(f, f_k), k = 1,2...n\ and\ f_a \in C_i\ then\ f \in C_i \quad (3)$$

2.2. Neural networks

An artificial neural network with an input layer and an output layer divides the input vectors in two semi plans. Solving complex problems implies the need of complicated decision regions, problem that can be solved by using networks with one or many extra layers between the input and the output layer [1], [10], [12].

When chaotic time series are involved, prediction is a difficult problem and can be viewed as temporal pattern recognition task, for which purpose neural networks suit very well.

The predicted value x(tk+1) of a variable x at a future time t_{k+1} is based on k previous values x(t1), x(t2), …, x(tk). Figures 2, 3 shows neural network structures for univariate and multivariate prediction [2]:

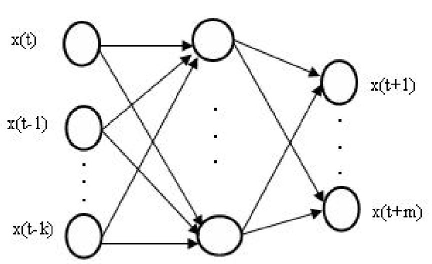


Figure 2. Connectionist univariate time-series prediction

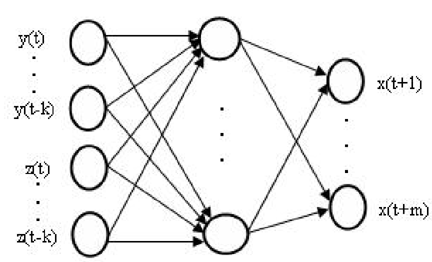


Figure 3. Connectionist multivariate (two) time-series prediction

The dependent variable subject to prediction can be different from the past data variables (independent variables), but both are on the time scale.

The multivariate prediction implies the prediction of both the dependent variable x and the prediction of the independent variables y, z at the moment t+1 starting from their precedent values. In this case the prediction of the variables y and z can be achieved with a neural network designed for the univariate prediction.

The backpropagation method and the sigmoidal activation function are the most important and used method for multilayer feed forward neural networks training that minimizes the mean squared error using the gradient method.

3. REFORME – A SOFTWARE PRODUCT FOR PATTERN RECOGNITION AND NEURAL NETWORKS

REFORME is a software product conceived by the author [4] of this paper, designed for pattern classification and recognition using specific techniques.

In this software we use pattern recognition and classification techniques referred earlier in the section 2.

The program has two components: pattern recognition and the multilayer perceptron.

This software allows the combined use of the two pattern and classification techniques for a common data set imported from Excel each of the imported pattern being one row in the spreadsheet.

The module “pattern recognition” has the following tasks:

- normalizes the inputs by means of the domain adjusting method;
- classifies unsupervised using the threshold algorithm with a classifier minimum distance based for the type of distance selected (Euclidean, Manhattan, Hamming) and specified threshold value. The classes resulted will be numbered from negative evolution toward positive evolution using an algorithm that induces the relation “<” over the classes set.
- determines to which class belongs an unknown pattern and also determines the pattern at the minimum distance from the unknown pattern by means of the nearest neighbor rule.

The resulted classes can be also used to train a neural network (supervised learning) that can be subsequently used as predictor.

The module “multilayer perceptron” accomplishes the following tasks: defines the network architecture, learns supervised using the back propagation error algorithm, test and evaluation network, pattern recognition based on the results obtained in the training step.

According to the values communicated for the number of neurons I on the input layer, the number of neurons H on the hidden layer, the number of neurons O on the output layer, will be generated the WI(HxI) and WO(OxH) random subunit weight matrix corresponding to the connections between the input layer and the hidden layer, the hidden layer and the output layer respectively.

For the supervised learning phase the admissible error and the learning rate are specified a priori (but the backprop algorithm can be used also with a variable learning rate). A number of epochs are executed until the error condition is satisfied or until the specified number of epochs is completed. From one epoch to another the patterns are randomly sorted and the weight matrix will be replaced with the new weights calculated using the backpropagation method.

After the completion of a specified number of epochs, the training phase can be resumed with another number of epochs and another learning rate. The mean error at the end of each epoch is displayed on screen on a visible area of the Sheet1.

The assessment – test phase comprises the assessment of the result during the training process for a test data set.

The network previously trained with a specified pattern set is used during the pattern recognition phase, with the weights resulted from the training phase.

4. GRAPHICAL REPRESENTATION OF THE ANALYZED ACTIVITY EVOLUTIONS

D(x) represents a measure of the activity described by the pattern x.

For each form belonging to the class c the similar method is carried out, M(c) being defined as: A graphical representation of the evolution of the activity in a coordinate system xOy can be obtained as follows: each pattern represents a point on a plane, the axis x being the time period corresponding to the pattern and y is the class that the pattern belongs to as shown in Figure 6. The accuracy of the representation depends on the number of classes. This representation implies first to sort and then renumber the classes taking into consideration the evolution direction (from negative evolution towards positive evolution).

To order the classes, for each class c is assigned a number M(c) that is calculated as presented below.

Let $x(x_1, x_2,\ldots,x_n)$ be a pattern that belongs to the class c, with normalized parameters x_i. For each parameter x_i of the pattern x is assigned a weight p_i representing the importance of the parameter x_i (the weights can be computed for instance as the partial correlation coefficients or can be set up by the expert).

For the pattern x, D(x) is calculated using the equation:

$$D(x) = \sum_{i=1}^{n} p_i x_i \tag{4}$$

$$M(c) = \left(\sum_{x \in c} D(x) \right) / p \tag{5}$$

where p is the number of forms belonging to class c.

The class c1 is in relation "<" toward the class c2 if $M(c_1)<M(c_2)$.

An order relation over the class set has been defined. Renumbering the classes and taking into account this order relation, a plot can be drawn representing the evolution of the activity analyzed as shown in Figure 6 and Figure 8.

Given two time intervals t_1 and t_2, $t_1<t_2$ and $D_{t1}(x)$, $D_{t2}(x)$:

- If $D_{t1}(x)<D_{t2}(x)$ then the activity defined by the patterns x has a positive evolution at the moment t_2 toward t_1;
- If $D_{t1}(x)=D_{t2}(x)$ then the activity defined by the patterns x is stationary at the moment t_2 toward t_1;

- If $D_{t1}(x)>D_{t2}(x)$ then the activity defined by the patterns x has a negative evolution at the moment t_2 toward t_1.

A similar interpretation can be done considering M(c). A much accurate representation of the activity evolution in a coordinate system xOy can be obtained as follows: each pattern represents a point in plane, x being the time range that corresponds to the pattern and y is D(x) as shown in Figure 7.

5. EXPERIMENTAL RESULTS – MACROECONOMIC FORECASTING

We consider the next macroeconomic indicators. ([9]):

Table 1

The prediction of the main macroeconomic indicators between 2006-2010 (modification toward the previous year)

CODE	INDICATOR	U/M	2003	2004	2005	2006	2007	2008	2009	2010
I1	Gross Domestic Product	billion RON	197.6	246.4	287.2	330.3	372.3	414.2	453.3	492.1
I2	Goods export (average)	million. EURO	15614	18935	22255	25500	29150	33100	37300	41800
I3	Goods import (CIF)	million EURO	21201	26281	32569	38620	44300	50450	56900	63500
I4	Goods import (FOB)	million EURO	19569	24258	30061	35650	40910	46600	52550	58650
I5	Occupational populations (average)	thousands pers.	8274.6	8087.7	8095	8105	8120	8135	8150	8165
I6	Average number of employees	thousands pers.	4590.9	4468.8	4575	4675	4760	4825	4895	4945
I7	Gross Income (average)	RON	664	818	958	1080	1200	1310	1430	1545
I8	Number of unemployment (the ending of the year)	thousands pers.	658.9	557.9	523	520	508	495	485	480

Taking into consideration the specific problems regarding macroeconomic forecasting, the indicators represent the parameters of the pattern and a pattern X_t is defined by the values of the parameters for an year.

For diagnostication is considered the interval between 2003 and 2006, and for prediction the interval between 2007 and 2010. The input data are processed in Sheet3. Each row of the datasheet represents a pattern and characterizes the degree of economical development for the year considered. After the data normalization, classification (unsupervised recognition using the threshold algorithm) with REFORME software, the distance used being the Euclidean distance and the threshold value =1, results the division of the patterns into 4 classes, the ordering and the renumbering of the classes as well as the representation of the plot for the evolution using the two solutions (classes, D(x)).

The results obtained are presented in the figures 4, 6, 7.

Figure 4

The result after the data normalization, classification (unsupervised recognition using the threshold algorithm) using Euclidean distance and the threshold value =1

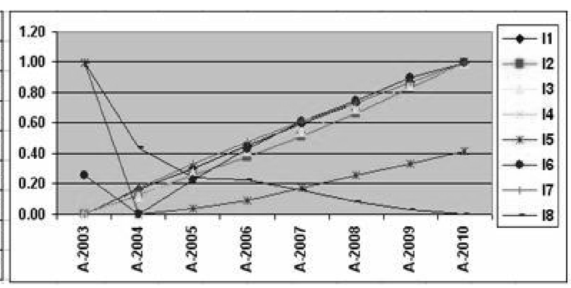
B1 (A=Learning, T=Test evaluation, P=Recognition, AT=Learning+Test, X=Unutilised)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ID_Pattern	(A=	I1	I2	I3	I4	I5	I6	I7	I8	Class	D(x)	M©
2	A-2003	A	0.0000	0.0000	0.0000	0.0000	1.0000	0.2564	0.0000	1.0000	2	2.26	1.98
3	A-2004	A	0.1657	0.1268	0.1201	0.1200	0.0000	0.0000	0.1748	0.4354	1	1.14	2.28
4	A-2005	A	0.3042	0.2536	0.2688	0.2685	0.0391	0.2230	0.3337	0.2404	1	1.93	
5	A-2006	A	0.4506	0.3775	0.4118	0.4115	0.0926	0.4330	0.4722	0.2236	1	2.87	
6	A-2007	A	0.5932	0.5169	0.5461	0.5461	0.1728	0.6115	0.6084	0.1565	3	3.75	4.62
7	A-2008	A	0.7355	0.6678	0.6915	0.6917	0.2531	0.7480	0.7333	0.0838	3	4.60	
8	A-2009	A	0.8683	0.8282	0.8440	0.8439	0.3333	0.8950	0.8695	0.0279	3	5.51	
9	A-2010	A	1.0000	1.0000	1.0000	1.0000	0.4136	1.0000	1.0000	0.0000	4	6.41	6.41
10													
11	Minim		197.60	15614.00	21201.00	19569.00	8087.70	4468.80	664.00	480.00			
12	Maxim		492.10	41800.00	63500.00	58650.00	8274.60	4945.00	1545.00	658.90			

Figure 5

The evolution of the indicators I1-I8

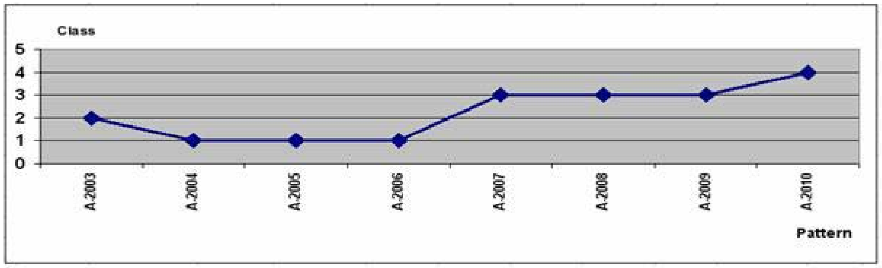
ID Pattern	I1	I2	I3	I4	I5	I6	I7	I8
A-2003	0.00	0.00	0.00	0.00	1.00	0.26	0.00	1.00
A-2004	0.17	0.13	0.12	0.12	0.00	0.00	0.17	0.44
A-2005	0.30	0.25	0.27	0.27	0.04	0.22	0.33	0.24
A-2006	0.45	0.38	0.41	0.41	0.09	0.43	0.47	0.22
A-2007	0.59	0.52	0.55	0.55	0.17	0.61	0.61	0.16
A-2008	0.74	0.67	0.69	0.69	0.25	0.75	0.73	0.08
A-2009	0.87	0.83	0.84	0.84	0.33	0.90	0.87	0.03
A-2010	1.00	1.00	1.00	1.00	0.41	1.00	1.00	0.00



One can observe from Figure 5 that the image representing the evolution of each indicator is not relevant for the overall analysis of the activity. A relevant representation of the activity evolution can be obtained using the classes obtained through pattern recognition techniques, or using the scalar measure D(x) defined above. The representation using classes is increasingly accurate as the number of classes is greater (the ideal case is when each form defines a new class, this conclusion is obvious when analyzing the Figures 6, 7, 8).

Figure 6

Graphical representation of analyzed activity using four classes



M(1)=1.98 M(2)=2.28 M(3)=4.62 M(4)=6.41

Figure 7

The graphical representation of analyzed activity using D(x)

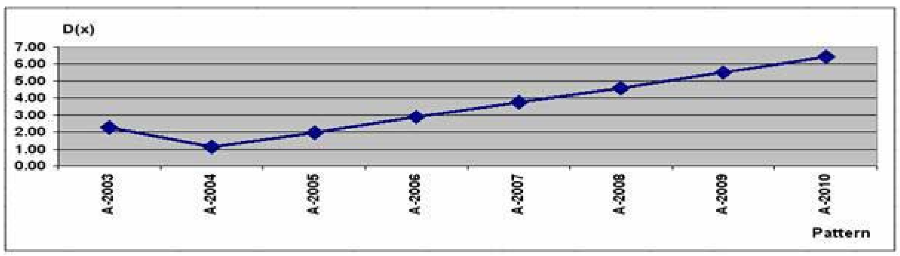
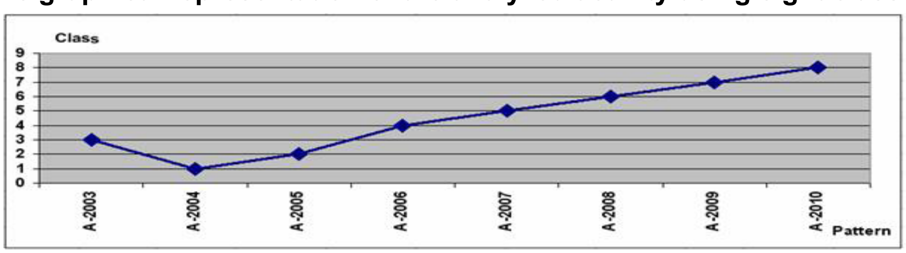


Figure 8

The graphical representation of the analyzed activity using eight classes



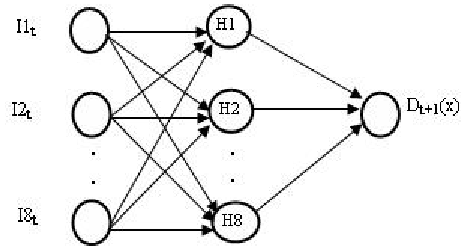
For a better approximation one solution would be the design of specific models for nonlinear systems [2], [5], [8].

For the prediction of the values corresponding to the next year the following neural architecture is defined: 8 neurons on the input layer, 8 neurons on the hidden layer and one neuron on the output layer. The network architecture is presented in Figure 9.

For training, the network uses the inputs for time t and the t+1 output is obtained.

Figure 9

The network arhitecture for the next year prediction



The results obtained with the software REFORME (using sigmoid activation function for the hidden layer and identity function for the output layer) are presented below.

The training process is carried out in three phases as well presented in the figures 10, 11, 12, 13. The training procedure begins with the training for a number of years and then the prediction for the next years is carried out. The training is then restarted including in the training set the values predicted and the next prediction for the next interval is carried out.

Figure 10

The results of network training and prediction - phase 1

Avg.error:, 9.75E-07, Learning rate = 0.3, No. Epochs = 9000, No. hidden neurons = 8

ID_Pattern	D(x)	Required D(x)	Calculated D(x)	Predicted D(x)
A-2003	2.26			
A-2004	1.14	1.14	1.14	
A-2005	1.93	1.93	1.93	
A-2006	2.87	2.87	2.87	
P-2007	3.75			3.75
P-2008	4.60			4.59
P-2009	5.51			5.32
P-2010	6.41			5.95

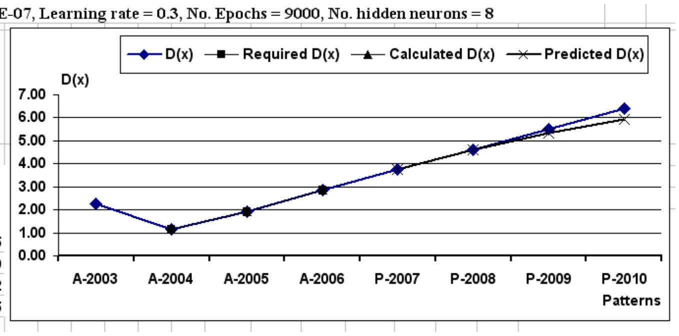


Figure 11

The results of network retraining and prediction - phase 2

Avg.error:, 4.03E-07, Learning rate = 0.3, No. Epochs = 2430, No. hidden neurons = 8

ID_Pattern	D(x)	Required D(x)	Calculated D(x)	Predicted D(x)
A-2003	2.26			
A-2004	1.14	1.14	1.14	
A-2005	1.93	1.93	1.93	
A-2006	2.87	2.87	2.87	
A-2007	3.75	3.75	3.75	
P-2008	4.60			4.6
P-2009	5.51			5.35
P-2010	6.41			6.00

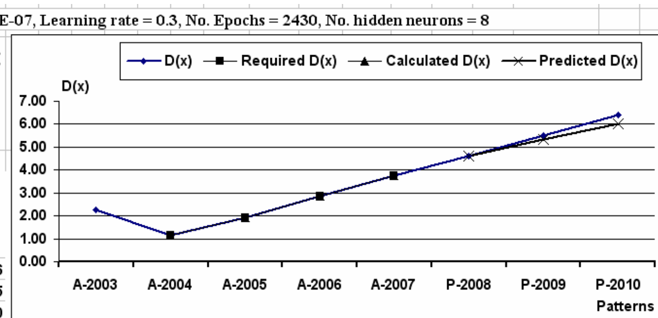


Figure 12

The results of network retraining and prediction - phase 3

Avg.error:, 1.09E-04, Learning rate = 0.1, No. Epochs = 5000, No. hidden neurons = 8

ID_Pattern	D(x)	Required D(x)	Calculated D(x)	Predicted D(x)
A-2003	2.26			
A-2004	1.14	1.14	1.14	
A-2005	1.93	1.93	1.93	
A-2006	2.87	2.87	2.89	
A-2007	3.75	3.75	3.72	
A-2008	4.60	4.60	4.61	
P-2009	5.51			5.50
P-2010	6.41			6.28

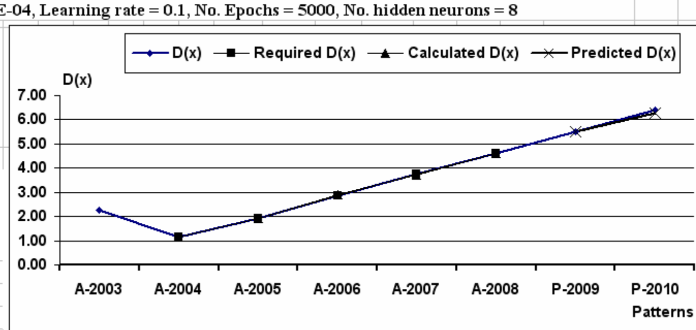
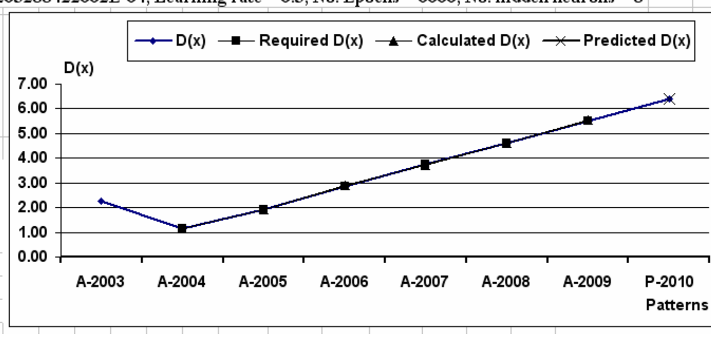


Figure 13

The results of network retraining and prediction - phase 4

Avg.error:, 1.89263288422602E-04, Learning rate = 0.3, No. Epochs = 6000, No. hidden neurons = 8

ID_Pattern	D(x)	Required D(x)	Calculated D(x)	Predicted D(x)
A-2003	2.26			
A-2004	1.14	1.14	1.14	
A-2005	1.93	1.93	1.93	
A-2006	2.87	2.87	2.89	
A-2007	3.75	3.75	3.72	
A-2008	4.60	4.60	4.61	
A-2009	5.51	5.51	5.54	
P-2010	6.41			6.39



Conclusions

The paper describes a method that quantifies and represents the evolution of an activity. Starting from a set of indicators that characterizes an activity, are defined a scalar D(x) that allows to measure an activity described by the x patterns and an order relation “<” over the set of classes resulted using unsupervised classification techniques for the graphical representation of the activity evolution using classes.

One of the methods used for solving diagnostication and prediction problems is the regression analysis. In the case of linear systems the regression approach has practical relevance. Because of their ability to detect nonlinear dependences in the input data set, neural networks represent an efficient alternative to the existing methods.

The prediction of the evolution of the nonlinear system is a difficult problem and sometimes impossible to solve. The obtained results using the model presented in this paper, confirm that prediction for a longer time period (4 years) is not accurate (Figure 10). In order to achieve accurate long-period prediction, a multiple phase prediction (4 years, 3 years, 2 year, 1 year) is carried out. In each phase the neural network is retrained using the results of the prediction obtained in the previous phase.

In order to test the model, we have used a narrower set of real data available at this time.

The researches in the field take into consideration other approaches that are using concepts specific to chaos theory such as: the phase space, attractors, fractals and Lyapunov coefficients measuring the sensitivity to the initial conditions, one of the main characteristics of the chaotic systems.

References

Abdi Herve, Les Reseaux de Neurones, Universite de Bourgogne,1993.

Nikola K. Kasabov, Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering, A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, 1998, pp 393-397

Menahem Friedman, Abraham Kandel Introduction to pattern recognition statistical, structural, neural and fuzzy logic approaches, World Scientific, 2000

Nicolae Morariu, “REFORME – A software product for pattern classification and recognition by joint use of pattern recognition techniques and multi-layer perceptron”, “The Proceedings of the Central and East European Conference in Business Information Systems”, Cluj-Napoca, may 2004, Ed. Risoprint, ISBN 973-656-648-X, pag.100-105.

Sorin Vlad, “Neural Networks Applications-Chaotic Time Series Prediction”, Distributed Processing, Suceava University Publishing House, Romania, 2005.

Ştefan-Gheorghe Pentiuc, “Aplicaţii ale recunoaşterii formelor în diagnosticul automat”, 158 p. ISBN 973-31-1096-5, Editura Tehnică, Bucureşti, 1997.

Romul Vancea, Ştefan Holban, Dan Ciubotariu, “Recunoaşterea Formelor Aplicaţii”, Ed. Academiei R.S.R. 1989.

Iulian Năstac, Emilian Dobrescu, Elena Pelinescu, ”Neuro-Adaptive Model for Financial Forecasting”, Romanian Journal of Economic Forecasting – 3/2007

www.biblioteca.ase.ro, Proiecţia principalilor indicatori macroeconomici în perioada 2006-2010, Comisia Naţională de Prognoză, 3 mai 2006.

Daniel Graupe, “Principles of Artificial Neural Networks, 2nd edition”, World Scientific 2007

Nicolas Carnot, Vincent Koen, Bruno Tissot, “Economic Forecasting”, Palgrave Macmillan 2005.

Alexander I. Galushkin, “Neural Networks Theory”, Springer 2007.