Self-organization of Neural Networks with Active Neurons

Published in Pattern Recognition and Image Analysis, v.4, no.2, pp.185-196

A.G. Ivakhnenko[*], G.A. Ivakhnenko[*] and J.-A. Müller[**]

[*]*Glushkov Institute of Cybernetics, National Academy of Sciences, Kiev, Ukraina*
[**]*Hochscule für Techik und Wirtschaft, Dresden, Germany*

Abstract. The paper discusses neural networks intended to handle the tasks of relation approximation (identification) and random process extrapolation (forecasting). Their salient feature is that their ensembles of elements are specified a priori as perceptrons or perceptron-like GMDH algorithms briefly outlined herein. Complex neuron ensembles may be called active neurons in contrast to simple binary neurons. In the course of learning or self-organization, active neurons select their input stimuli according to a specified criterion and determine weight coefficients for connections. In this way, they organize the structure of the entire neural network. The objective sought in constructing a neural network is to enhance the efficiency in handling the task set for each neuron. The number of active neurons is determined by the amount of information coming in for processing as a sample of observations over a test or controlled object. In each layer of a neural network, the active neurons differ from one another in sets of input and output variables. The more important difference between active neurons lies in the structure of their algorithms, which in fact runts an active-neuron neural network into a multilayered decision collective rule [I,2,3].

Introduction

According to physiologists, connections between neurons in the brain are not permanent. They form and break up in response to external stimuli, and this happens at an especially high rate during the first days after the birth of the organism. Supposedly, this process depends solely on the adaptability of a neuron itself to external stimuli. A fundamental fact is that every neuron selects its input connections all by itself according to a specified criterion; the connections are not imposed on it by the environment. The environment only sets limits for the choice of stimuli according to the criterion defined for neurons. Neurons for which connections are assigned from without by the person in charge of modeling or by other neurons should be called *passive neurons*. In contrast, neurons that select their input variables (through learning or self-organization) may be called *active neurons*. The connections selected by active neurons uniquely define the structure of connections for the entire neural network. So far now neural networks have been mainly developed for passive neurons, such as the simple neurons of McCulloch and Pitts, with specified binary input variables [3,4].

The aim of this paper is to extend the theory of self-organization for isolated models [6,7,8,9] to active-neuron networks neurons. In fact, there is a good deal of similarity between the self-organization of models and the self-organization of neural networks. The key procedures of self-organization for models, such as the selection of layer number by exhaustive search according to

an external criterion, optimization of sets of input and output variables, and exhaustive-search termination rules, are valid for the self-organization of both an isolated model and a neural network.

The objective sought in combining active neurons into a neural network is to enhance the accuracy in achieving the task facing both every neuron and the entire neural network. Neurons find themselves in different conditions. They may differ in both output variables and in the set of input variables. In the example, that follows a neural network is used to forecast random processes in a complex economic object with fuzzy characteristics.

1. The spectrum of GMDH algorithms used as active neurons

The GMDH algorithms mainly differ in how the candidate models to be tested by an exhaustive search for compliance with an external criterion are generated. The criteria used most often are of the precision, differential or informative type.

The principal GMDH algorithms listed in Table 1 have been developed for continuous variables. Among the parametric algorithms (with the model coefficients evaluated according to the criterion minimum), those most known are the *Combinatorial algorithm* [7], the *Multilayered Iterative algorithm* [8], and the *Objective Systems Analysis algorithm* [9]. The basic one is the Combinatorial algorithm, which applies an exhaustive search to all polynomial models that can be derived from a complete linear polynomial by eliminating some of its terms. In some cases, the Multilayered algorithm may miss some models, thus giving rise to what is known as multilayer error. For example, it may occur when the true relation contains terms that cannot be formed with partial linear descriptions.

Those less known are the parametric algorithms, which apply an exhaustive search to harmonic and harmonic-exponential functions, and the *Multiplicative-Additive* GMDH algorithm, in which the polynomial models to be tested by exhaustion are obtained by taking the logarithm of the product of input variables [10,11].

Variables	GMDH algorithms	
	Parametric	**Non-parametric**
Continuous	- Combinatorial (COMBI) - Multilayered Iterational (MIA) - Objective System Analysis (OSA) - Harmonical - Two-level (ARIMAD) - Multiplicative-Additive	- Objective Computer Clusterization (OCC); - "Pointing Finger" (PF) clusterization algorithm; - Analogues Complexing (AC)
Discrete and binary	- Harmonical Rediscretization	- Algorithm on the base on Multilayered Theory of Statistical Decisions (MTSD)

Table 1. Spectrum of GMDH algorithms.

The parametric GMDH algorithms have proved highly efficient in cases where one is to model objects with nonfuzzy characteristics, such as, engineering objects. In cases, where modeling involves objects with fuzzy characteristics, it is more efficient to use the non-parametric GMDH algorithms, in which polynomial models are replaced by a data sample divided into intervals or clusters. They are exemplified by the *Objective Computer Clusterization* (OCC) algorithm that operates with pairs of closely spaced sample points (called dipoles [12]), implemented by resting two hierarchical clustering trees for compliance with the balance criterion [13], and the *Analog Complexing algorithm* [14]. The nonparametric class also includes the GMDH algorithm based on the *Statistical Decision Theory* [15]. It is recommended in cases where one is to test the truth of the input data sample and to remove from it the error caused by the sample-defining expert.

As already noted, GMDH algorithms have been developed for continuous variables. In practice, however, the sample will often include variables discretized into a small number of levels or even binary values. To extend GMDH algorithms to discretized or binary variables, the harmonic dediscretization algorithm has been developed [15].

The existence of a broad gamut of GMDH algorithms is traceable to the fact that it is impossible to define the characteristics of the rest or controlled objects exactly in advance. Indeed, one type of objects may require one algorithm and another type, a different algorithm. Therefore, it will be good practice to try several GMDH algorithms one after another and to decide which one suits a given type of objects best. The lower the minimum of the discriminating criterion, the better the rested algorithm. In this way, the type of algorithm is chosen objectively, according to the value of the rest criterion.

1.1. The Combinatorial GMDH algorithm (COMBI)

The flowchart of the algorithm is shown in Fig. 1. The input data sample is a matrix containing N levels (points) of observations over a set of M variables. The sample is divided into two parts. Two-thirds of points having a high variance make up the learning subsample N_A, and the remaining one-third of points form the check subsample N_B. The learning sample is used to derive estimates for the coefficients of the polynomial, and the check subsample is used to choose the structure of the optimal model, that is, one for which the regularity criterion *AR(s)* takes on a minimal value:

$$AR(s) = \frac{1}{N_B}\sum_{i=1}^{N}(y_i - \hat{y}_i(B))^2 \to \min \qquad (1)$$

or better to use the cross-validation criterion *PRR(s)* (it take into account all information in data sample and it can be computed without recalculating of system for each checking point):

$$PRR(s) = \frac{1}{N}\sum_{1}^{N}\left[y_i - y_i(B)\right]^2 \to \min, \qquad N_A = N - 1; \quad N_B = 1.$$

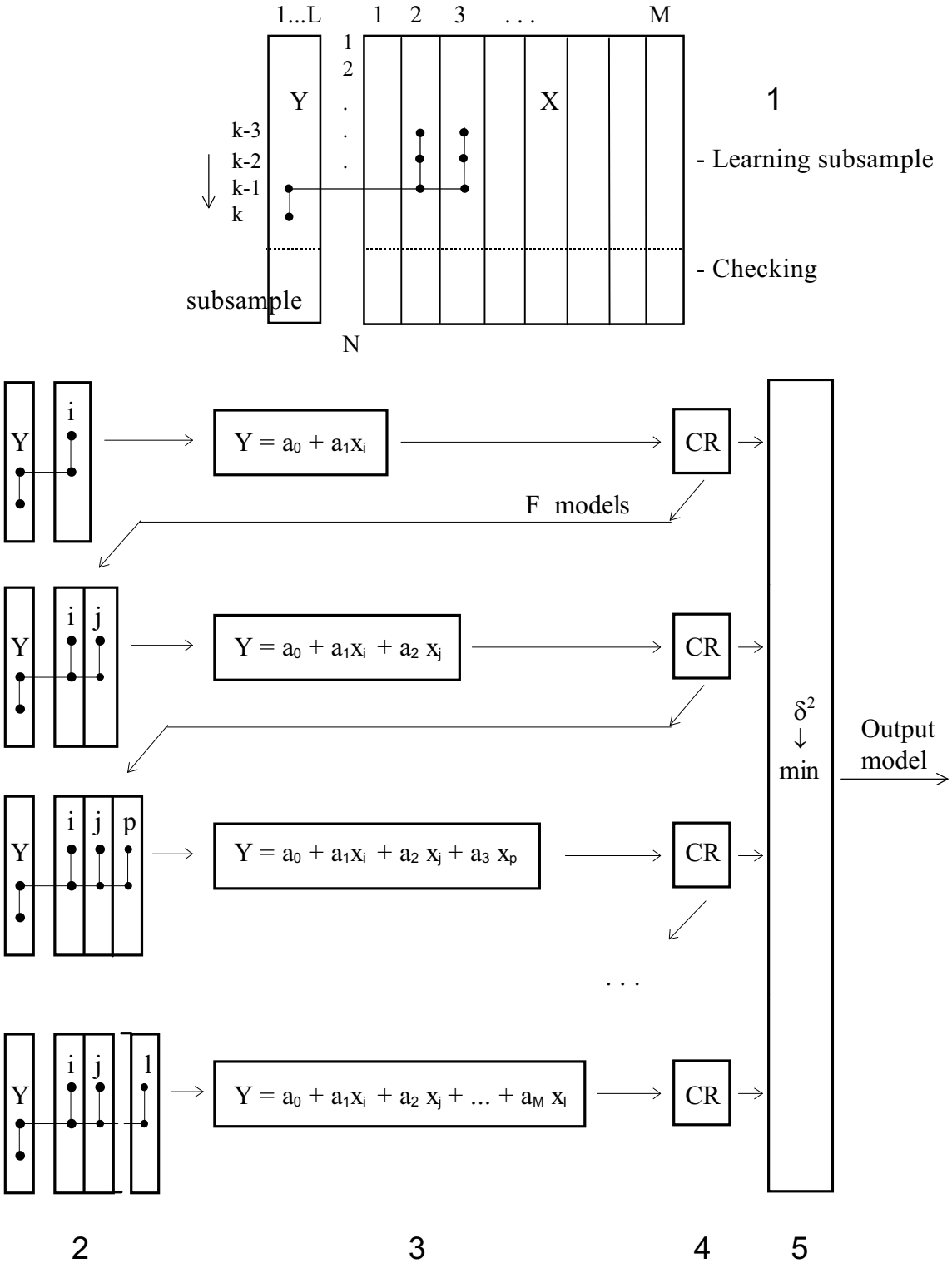


Fig. 1. Combinatorial GMDH algorithm.

1 - data sampling;
2 - layers of partial descriptions complexing;
3 - form of partial descriptions;
4 - choice of optimal models;
5 - additional model definition by discriminating criterion.

To obtain a smooth exhaustive-search curve (Fig. 2), which would permit one to formulate the exhaustive-search termination rule, the exhaustive search is performed on models classed into groups of an equal complexity. The first layer uses the information contained in every column of the sample, that is, the search is applied to models of the form:

$$y = a_0 + a_1 x_i, \qquad i = 1,2,...,M . \tag{2}$$

The output variable must be specified in advance by the experimenter. Only a small number of variables (usually, F = 5), showing the best results in the first layer, are allowed to form second-layer candidate models of the form

$$y = a_0 + a_1 x_i + a_2 x_j, \qquad j = 1,2,...,M . \tag{3}$$

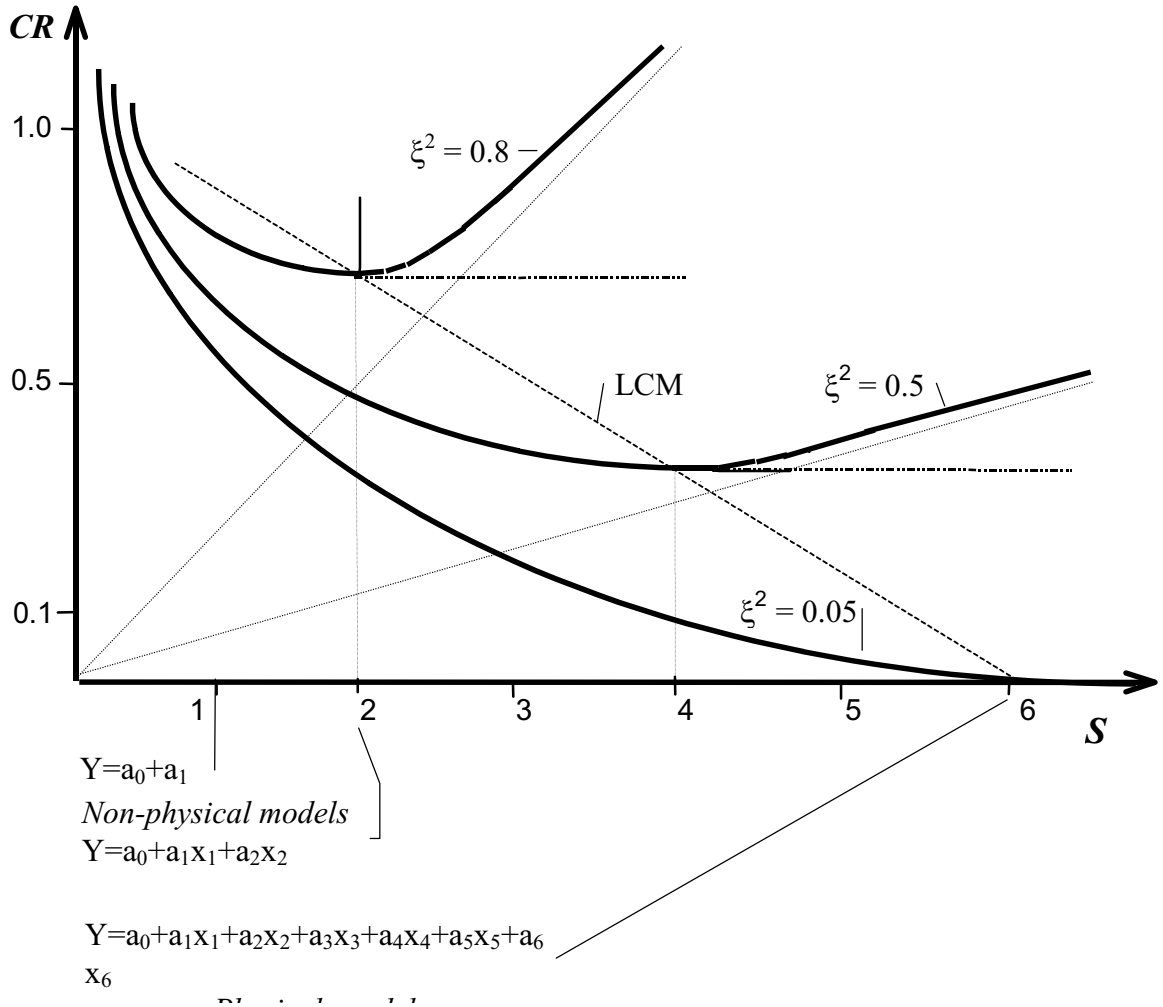


Fig. 2. External accuracy criterion minima values plotted against complexity of model structure S for different noise variance ξ^2.

LCM - locus of criterion minima line;
--- - model choice by criterion minimum;
-·-· - model choice by "left corner rule".

The second-layer models are evaluated for compliance with the criterion, and F best variables are allowed to form third-layer candidates. The procedure is carried on as long as the criterion decreases in value.

The above procedure achieves the following goals. (I) Only the number F of variables yielding the best result are allowed to form the more complicated models of the subsequent layers. (2) Instead of one layer, two last layers are used to determine the best result of the exhaustive search, so that the value of the criterion on the exhaustive-search curve can only decrease (compare the values represented by the dashed and solid lines in Fig. 2). The optimal model has to be chosen not by the criterion, but by what we call the *left-corner rule*. (3) The best variables of each layer are used to continuously expand the input data sample: with each layer, the exhaustive search adds F columns to their number in the sample. Typical curves for the exhaustive search on the basis of the precision criterion and for different noise variances are shown in Fig. 2. It was proved, that calculations are faster when:

a) in all formulae informational array W^TW is used instead of data sampling array $W=(XY)$;

b) models parameters are calculated by recursion method of "framing".

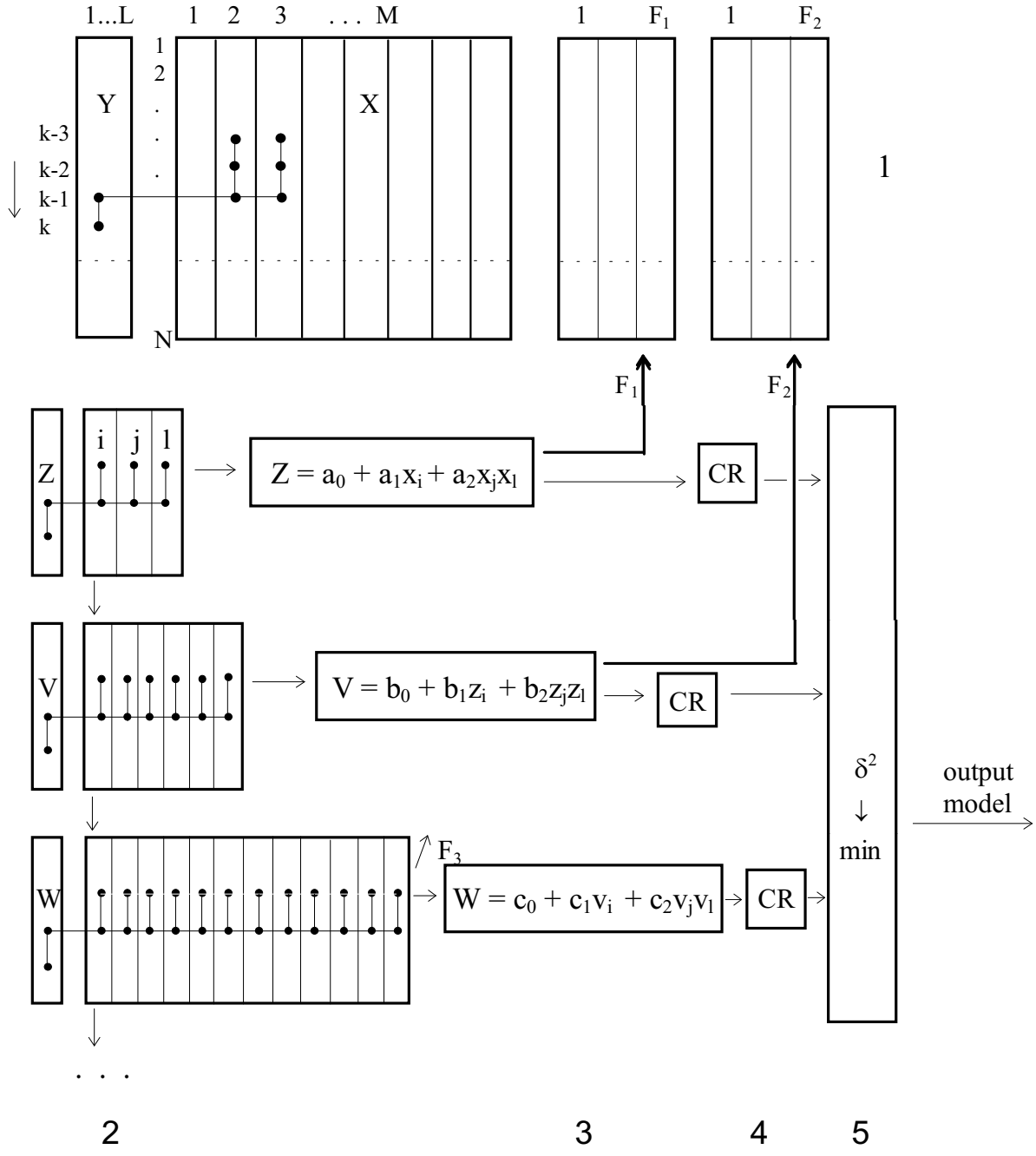
A salient feature of the GMDH algorithms is that, when they are presented continuous or noisy input data, they will yield as optimal *some simplified nonphysical model*. If is only in the case of discrete or exact data that the exhaustive search for compliance with the precision criterion will yield what is called a *physical model*, the simplest of all unbiased models. With noisy and continuous input data, simplified (Shannon) models [6,7,8,9] prove more precise in approximation and for forecasting tasks.

To test a model for compliance with the differential balance criterion, the input sample is divided into two equal parts. The criterion requires to choose a model that would, as far as possible, be the same on both subsamples. The balance criterion will yield the only optimal physical model solely if the input data are noisy. With exact noise-free data, the criterion leads to a multiplicity of optimal models. To resolve the ambiguity in such a case, resort has to be made to a regularizing combination criterion (the algorithm for the extended definition of the only optimal model).

1.2. The Multilayered Iterative GMDH algorithm (MIA)

The Combinatorial algorithm described in the previous subsection may likewise be called a multilayered or iterative one, despite the fact that the iteration rule grows more complicated with every layer. This is possibly the reason why it is customary to call the Multilayered Iterative GMDH algorithm as algorithm in which the iteration rule remains unchanged from one layer to the next. As is shown in Fig.3, the first layer tests the models that can be derived from the information contained in any two columns of the sample. The second layer uses information from four columns; the third, from any eight columns, etc. The exhaustive-search termination rule is the same as for the Combinatorial algorithm: in each layer the optimal models are selected by the

minimum of the criterion or by the left-hand corner rule. In each layer, the F best models are used to successively extend the input data sample (Fig.4).



Output model: $Y_{k+1} = d_0 + d_1x_{1k} + d_2 x_{2k} + ... + d_m x_{M\,k}\, x_{M-1\,k}$

Fig. 3 Multilayered Iteration algorithm

1 - data sampling;
2 - layers of partial descriptions complexing;
3 - form of partial descriptions;
4 - choice of optimal models;
5 - additional model definition by discriminating criterion;
F1 and F2 - number of variables for data sampling extension.

As with the Combinatorial algorithm, the output variable must be specified in advance by the person in charge of modeling, which corresponds to the use of so-called explicit templates [8,21].

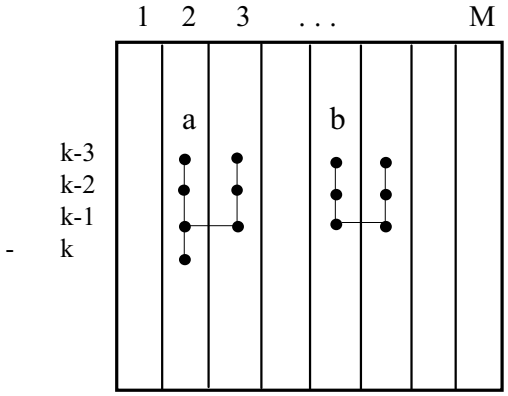


Fig. 4. Derivation of conditional equations on a data sample.

a - explicit templates form; b - implicit templates.

1.3. The Objective System Analysis algorithm (OSA)

In discrete mathematics, the term template refers to a graph indicating which of the delayed arguments are used in setting up conditional and normal Gauss equations. A gradual increase in the structural complexity of candidate models corresponds to an increase in the complexity of templates whose explicit and implicit forms are shown in Fig. 4. When one uses implicit templates, one has, beginning from the second layer of the exhaustive search, to solve a system of equations and to evaluate the model, using a system criterion. The system criterion is a convolution of the criteria calculated by the equations that make up the system

$$CR_{syst} = \frac{1}{S}\sqrt{CR_1^2 + CR_2^2 + ... + CR_S^2} \rightarrow \min, \qquad (4)$$

where s - is the number of equations in the system.

The flowchart of the OSA algorithm is shown in Fig. 5. The key feature of the algorithm is that it uses implicit templates, and an optimal model is therefore found as a system of algebraic or difference equations. An advantage of the algorithm is that the number of regressors is increased and in consequence, the information embedded in the data sample) is utilized better. A disadvantage of the algorithm is that it calls for a large amount of calculations in order to solve the system of equations and a greater number of candidate models have to be searched. The amount of search can be reduced, using a constraint in the form of an auxiliary precision criterion. In setting up the system of equations, one then discards the poorly forecasting equation (using equation only) for which the *variation accuracy criterion* for the forecast is less than unity (narrowing operation):

$$\delta_i^2 = \frac{\sum_1^N \left(y_i - \hat{y}_i\right)^2}{\sum_1^N \left(y_i - \bar{y}\right)^2} \rightarrow \min, \qquad (6)$$

where: y_i - is the variable value in the table; $\hat{y}_i$ - is the value calculated according to the model and $\bar{y}$ is the mean value.

This criterion is recommended in the literature in order to evaluate the success of an approximation or of a forecast [16]. With $\delta^2 < 0.5$, the result of modeling is taken to be good; with $0.5 < \delta^2 < 0.8$ it is taken to be satisfactory; with $\delta^2 > 1.0$, modeling is considered to have failed, and the model yields misinformation.)

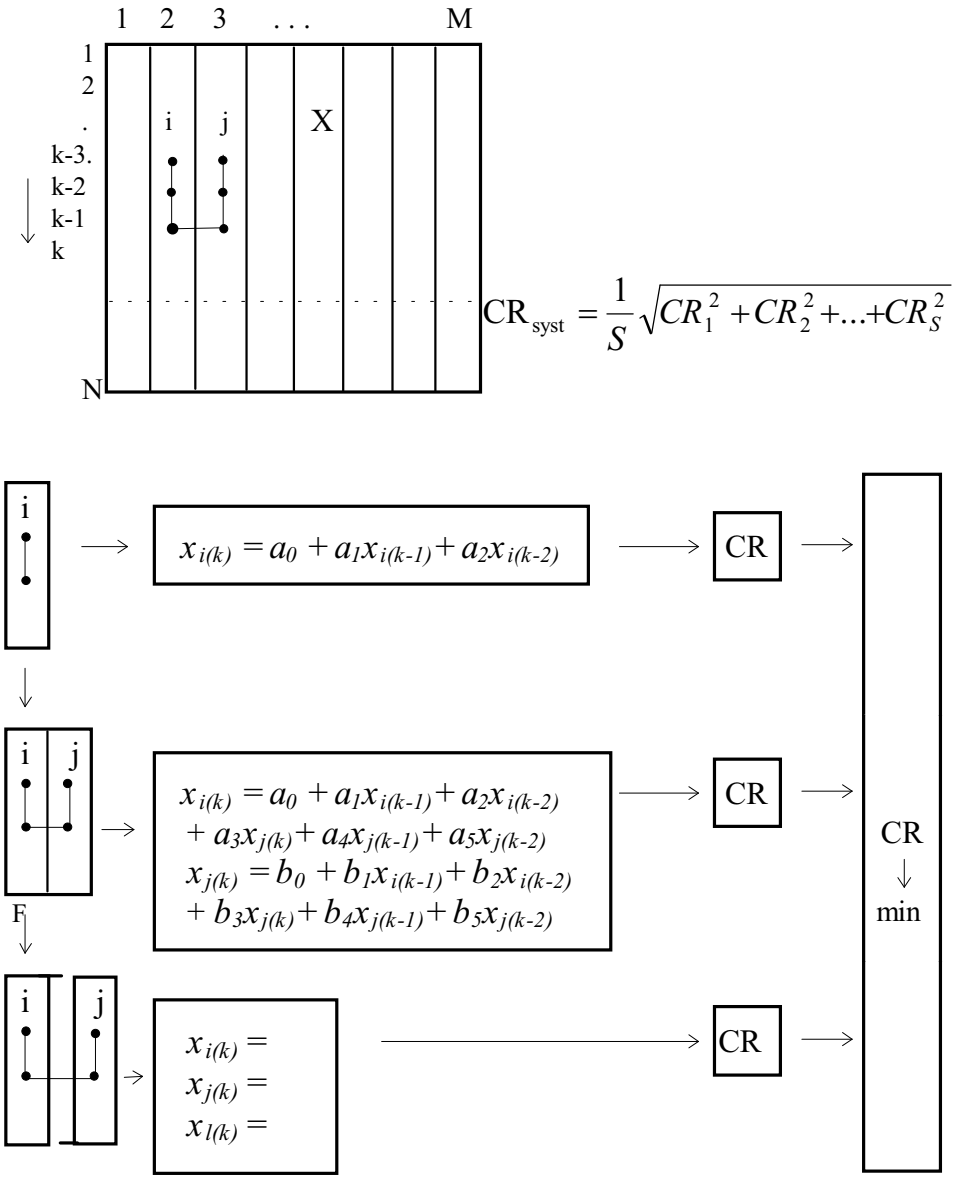


Fig. 5. Objective system analysis (OSA) algorithm

2. Extended definition of the only optimal model by the theory of discriminating criteria

It has been demonstrated theoretically and experimentally that the exhaustive-search curves shown in Fig. 2 are gradual and unimodal for the expected value of the criterion [7,8,9]. The number of candidate models tested in each exhaustive-search layer cannot be infinitely large. In other words, in constructing exhaustive search curves, the expected value of the criterion is in effect replaced by its mean (or least) value. Because of this, the curves take on a slightly wavy shape, and a small error may creep into the optimal model structure choice.

The theory of discriminating criteria has been developed by Fedorov [17] and Yurachkovsky [18] with special reference to experimental design. It has however proved its relevance to the self-organization of models and active-neuron neural networks. The theory proceeds from the following premises: (I) there exists a "true" model represented in the data sample; (2) the assumed few object descriptions fit the model to a different degree; (3) the model that comes closest to the true model can be selected from its compliance with an *auxiliary discriminating criterion*.

With such an approach, every GMDH algorithm consecutively uses two criteria. At first, an exhaustive search is applied to all candidate models for compliance with the main criterion, and a small number of models whose structure is close to optimal is selected. Then only one optimal model is selected that complies with a special discriminating criterion. The theory of optimal discriminating criteria is still in the developmental stage, but successful discriminating criteria are already known.

In cases involving the selection of a structure for optimal polynomial models, the approximation or forecast variation criterion serves well. In the selection of optimal clusterization, good results are obtained with the symmetry criterion for the clusters distance matrix calculated relative to the secondary diagonal [19], etc.

3. Self-organization of an active-neuron neural network

A neural network is designed to handle a particular task. This may involve relation identification (approximation), pattern and situation recognition, or a forecast of random processes and repetitive events from information contained in a sample of observations over a test or control object.

Each neuron is an elementary system that handles the same task. The objective sought in combining many neurons into a network is to enhance the accuracy in achieving the assigned task through a better use of input data. As already noted, the function of active neurons can be performed by various recognition systems, notably, by Rosenblatt's two-layer perceptrons [5]. Such a neural network achieves the task of pattern recognition. Its theory is presented in the algebraic theory due to Zhuravlev [1].

In the case of relation approximation (identification and extrapolation (forecast) of random processes and repetitive events, the active neurons are the GMDH algorithms mentioned earlier. In the self-organization of a neural network, the exhaustive search is first applied to determine the number of neuron layers and the sets of input and output variables for each neuron. The minimum of the discriminating criterion suggests the variables for which it is advantageous to build a neural network and how many neuron layers should be used. Thus, the theory of neural network self-organization is similar in many respects to that of each active neuron.

3.1. The search termination rule

In self-organization, the layers of neurons are extended as long as this improves the accuracy of the solution yielded by the neural network. This will be demonstrated later with reference to a relevant example.

3.2. Group allowance for arguments

We will call as the *exhaustive-search characteristic* of a neural network the graph that relates the main precision criterion for a specified variable to the layer number. This characteristic is similar to that of the GMDH algorithms. To obtain a smooth and unimodal curve, the exhaustive-search characteristic is calculated for many tools in the sample, and the results are averaged. Theoretically, the exhaustive-search characteristic has been investigated for the expected value of the criterion [7]. In practice, the exhaustive-search curve has to be constructed not for the expected value and even not for the mean value of the criterion. Rather, it is constructed for the best results of the exhaustive-search applied to a group for which the criterion takes on the least value. This exhaustive-search termination rule holds only when many approximation or forecast results are average.

3.3. The selection of a discrete template

What type of template to use depends on the task at hand (Fig. 4). In an approximation task, the template does not contain delayed arguments; in a forecast task, two or three delays have the be allowed for. In the former case, one obtains single-moment equations; in the latter, difference equations.

3.4. Extended definition of one optimal model for each neuron in a network

Self-organization of each neuron taken separately uses the differential balance criterion or the regularity precision criterion. As already noted, the exhaustive-search curve approaches its minimum in a gradual manner, and the criteria of models close to the optimal one differ only slightly from one another in value. This explains why one has to use an extended definition algorithm. By this algorithm, instead of one, several of the best models are selected. From them, only one that complies with another variation discriminating criterion chosen.

3.5. Readout of modeling results

Each layer in a neural network contains neurons, whose outputs correspond each to a particular specified variable: the output of the first neuron to the first variable, the output of the second neuron to the second variable, etc.

Each column consists of neurons whose outputs correspond to one of the variables. From each column in turn, one neuron with a minimal variation criterion is selected. More specifically, one neuron having the best result is selected from the first column of neurons for which the output is the first variable; similarly, one neuron is selected from the second column of neurons for which the output is the second variable, etc. This selection procedure uniquely defines the number of layers for each variable and, thus, the structure of the neural network.

3.6. The exhaustive search of methods for data-sample extension and narrowing

The principal method of data-sample extension is by including the output variables from the previous layer that have complied with the criterion best of all. It will also be a good plan to test against the criterion the advisability of sample extension by simple nonlinear transformations of input variables. In the example that follows, three variables are involved. They are x_1 , x_2, and x_3.

(a) The extension using the covariance of the variables

$$v_1 = x_1 x_2, \quad v_2 = x_1 x_3, \quad v_3 = x_2 x_3.$$

(b) The extension using the reciprocals of the variables

$$v_1 = \frac{1}{x_1}, \quad v_2 = \frac{1}{x_2}, \quad v_3 = \frac{1}{x_3}.$$

The reciprocals should above all be proposed for the variables that take a minus sign in the equation; that is, they reduce the value of the output.

3.7. Sample extension by consecutive elimination of the most efficient variables.

The diversity of the variables that come in for the exhaustive search (performed by each neuron) can further be increased by eliminating the most efficient variables, thus producing partial subsets. This can be best illustrated by an example.

Let the input of a neural network accept a data sample containing just M=25 variables. Suppose further that we have used the OSA algorithm and found in the first neuron an optimal system of forecasting difference equations in the variables x_2 x_{12} x_{13} x_{18} x_{22} . These variables are least "fuzzy" and lend themselves to forecasting by this system of equation. We eliminate from the sample the variables thus found and apply the OSA algorithm to a second neuron. This yields a second optimal system of equations in the variables x_3 x_9 x_{14} x_{32}. As a result, the minimum of the criterion increases (because the second set contains other than the best variables) and shifts to the left. Now we eliminate from the sample the nine variables thus found, and apply the OSA

algorithm to a third neuron. This yields an optimal system of equations in only three variables x_5 x_6 x_{11}. The minimum of the criterion goes up still more and again shifts to the left etc.

This shift in the minimum of the system criterion bears out the adequacy law, which states that for more fuzzy systems the optimal description (model) must be likewise more fuzzy and simple; that is, it must have a smaller number of equations [6]. Computer experiments confirm the above form of exhaustive-search curve. In the above example, the number of variables used for decision-making is increased from 5 (in the first neuron) to 5 + 4 + 3 + 2 + I =15 (in five neurons). Ten features are discarded as inefficient. So, we shall have 5 x 15 = 75 neurons in each layer.

3.8 Simultaneous and successive algorithms for neural networks

In a computer program, neurons can be implemented simultaneously or successively, using memory devices.

4. Description of an algorithm for a neural network using forecasting GMDH algorithms as active neurons

An example of the algorithm structure is given in Fig. 6. Sample extension is effected solely by including the output variables from each previous layer of neurons. The samples show the form of the discrete template used to teach the first neurons of a layer by the Combinatorial GMDH algorithm. In particular, when two time delays are allowed for (τ=2), the first template corresponds (to the following complete difference equation:

$$x_{1(k)} = a_0 + a_1 x_{1(k-1)} + a_2 x_{1(k-2)} + \ldots + a_8 x_{4(k-1)} + a_9 x_{4(k-2)}$$

The GMDH algorithm will suggest which of the proposed arguments should be taken into consideration and will help to estimate the connectivity coefficients.

4.1. Self-organization by the Combinatorial GMDH algorithm of an active neuron network intended to forecast the activity index of a stock exchange

The objective of the forecast is this. Given a sample of observations over changes in a small number of *indirect indexes* for which the mathematical relation is not known, one is to form a short-term forecast for the activity of a stock exchange. This should involve all variables, but only those of them for which the forecast yields a variation of less than unity will be taken as the output variables of the forecast. In the example at hand, the input sample (so Table 2) contains generally accessible information. It lists the values of four variables related to sessions at the New York Stock Exchange from May 1 to August 17, 1991 (70 days). They are the minimal activity index x_1 the maximal activity index x_2, the closing activity index x_3, and the total volume of operations over a day x_4.

In schematic form, the neural network used for forecasting is shown in Fig. 6. To begin with, we construct the first layer of neurons in the network. Then, by applying the polynomial Combinatorial GMDH algorithm, we will able to determine how accurate the forecast will be for

all variables. For this purpose, we use a discrete template that allows a delay of one or two days for all variables. Then we add a second, a third, etc. layer to the neural network, as shown in Fig. 6, and go on doing so as long as this improves the forecast. This will demonstrate the efficiency of the neural network.

N	x_1	x_2	x_3	x_4	N	x_1	x_2	x_3	x_4
1	375	375.5	376.8	130	*36*	371	378	378	170
2	371	377	372	220	*37*	377	378	378	160
3	366	373	368	190	*38*	372	377	373	140
4	369	373	373	155	*39*	374	376	375	159
5	369	374	373	175	*40*	372	375	374	170
6	372	374	373	110	*41*	371	378	378	140
7	373	377	376	185	*42*	376	378	376.5	155
8	375	377	377	160	*43*	376	380	376	180
9	374	378	375	177	*44*	376	378	377.5	160
10	375	378	377	128	*45*	376	381	380	177
11	378	380	379	130	*46*	380	383	383	162
12	377	383	383	165	*47*	381	383	382	183
13	382	384	383	187	*48*	381	383	381	195
14	383	388	387	235	*49*	381	386	386	200
15	385	390	390	230	*50*	384	386	385	190
16	387	390	388	175	*51*	382	385	383	150
17	385	388	388	180	*52*	379	385	379.5	160
18	384	388	385	185	*53*	378	380.5	380.3	160
19	383	386	384	170	*54*	380	381	381	145
20	379	384	380	170	*55*	380	382	381	130
21	387	380	379	130	*56*	380	384	384	136
22	379	382	381	162	*57*	384	387	387	170
23	374	381	376	170	*58*	387	388	388	168
24	375	377	377	145	*59*	387	388	387.6	170
25	377	383	383	174	*60*	387	389	387.5	162
26	380	382	380	135	*61*	385	387	385.5	127
27	378	382	378	159	*62*	384	391	391	175
28	375	378	376	160	*63*	390	391.5	390	170
29	374	376.5	375	165	*64*	388	392	389	165
30	375	378	378	195	*65*	387	390	387.5	145
31	371	378	371	140	*66*	386	388	388	146
32	360	373	371	159	*67*	388	392	390	215
33	368	373	372	185	*68*	389	392	390	195
34	372	374	374	159	*69*	389	392	389	176
35	368	374	371	165	*70*	383	390	395	190

We have used the Combinatorial GMDH algorithm to provide for the self-organization of the neuron models shown in Fig.6. For each neuron, we have applied the extended definition procedure to one model (out of the three closest to the optimal one). For the optimal models, we have calculated the forecast variation criterion; the relevant exhaustive-search characteristics are in Table 3.

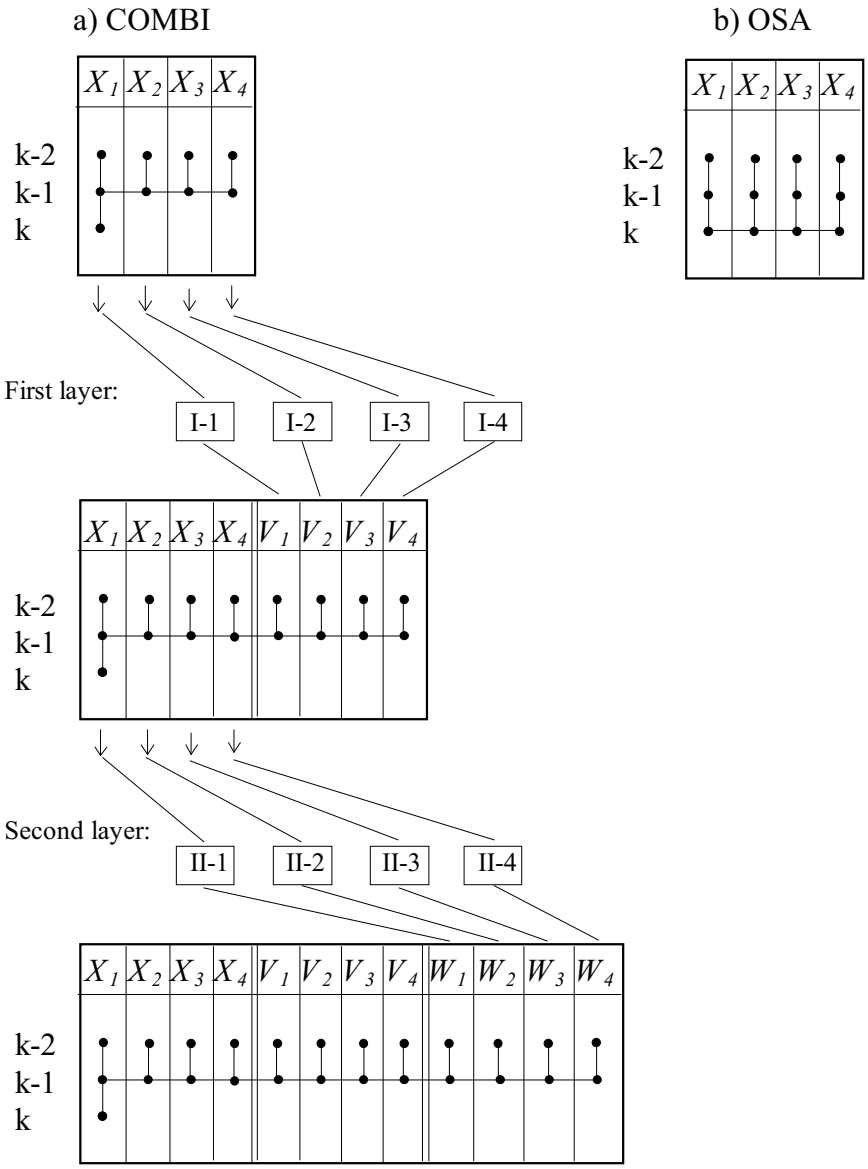


Fig 6. Schematic arrangement of the first two rows of a neural network.

Neuron row	$\delta^2(x_1)$	$\delta^2(x_2)$	$\delta^2(x_3)$	$\delta^2(x_4)$
1	I-1	I-2	I-3	I-4
	0.131	**0.085**	0.200	0.858
2	II-1	II-2	II-3	II-4
	0.143	0.089	0.193	**0.816**
3	III-1	III-2	III-3	III-4
	0.153	0.091	**0.178**	0.828
4	IIII-1	IIII-2	IIII-3	IIII-4
	0.158	0.093	0.183	0.832

Table 3. Minimal values of the variation criterion over neural network layers

It may be inferred that there is no need to construct a neural network in order to form a forecast for the variables x_1, and x_2 because the variation criterion takes on the least value in the first layer. It is advisable to use a neural network to form a forecast for the variable x_4, and especially x_3 for which the variation criterion takes on the least value in the third layer of neurons.

We give the difference equations for the neurons of the network that are needed in order to calculate the output variable x_3 of the last layer of neurons. The third layer of neurons:

Neuron III-3:

$$x_{3(k)} = 181{,}723 - 0.226x_{1(k-1)} + 0.669w_{1(k-2)} - 6.567w_{2(k-1)} - 0.097w_{4(k-1)} +$$
$$+ 0.747x_{3(k-1)} + 5.526x_{3(k-2)}; \qquad \delta^2 = 0.1778$$

The second layer of neurons:

Neuron II-1:

$$w_{1(k)} = 18{,}238 + 0.044x_{4(k-1)} + 0.041x_{4(k-2)} - 0.395v_{3(k-2)} - 0.054v_{4(k-2)} +$$
$$+ 0.735x_{1(k-1)} + 0.116x_{1(k-2)}; \qquad \delta^2 = 0.1430$$

Neuron II-2:

$$w_{2(k)} = 24.120 + 0.862x_{3(k-1)} + 0.032v_{1(k-2)} - 0.046v_{2(k-2)}; \qquad \delta^2 = 0.0898$$

Neuron II-4:

$$w_{4(k)} = -91{,}599 - 1.735x_{2(k-1)} + 1{,}959x_{2(k-2)} + 0.193x_{3(k-1)} - 12.441v_{2(k-1)} +$$
$$+ 10.895x_{3(k-1)} + 0.225x_{4(k-1)}; \qquad \delta^2 = 0.8157$$

The first layer of neurons (only the neurons required to calculate the variables in the succeeding neurons are given):

Neuron I-I:

$$v_{1(k)} = 0.141 + 0.763x_{2(k-1)} + 0.070x_{2(k-2)} - 0.021x_{4(k-2)} - 0.309x_{1(k-1)}; \qquad \delta^2 = 0.1317$$

Neuron I-2:

$$v_{2(k)} = 33.254 + 0.103x_{1(k-2)} + 0.909x_{3(k-1)} - 0.001x_{4(k-2)} - 0.094x_{2(k-1)}; \qquad \delta^2 = 0.0852$$

Neuron I-4:

$$v_{4(k)} = -73.738 + 0.292x_{1(k-1)} + 1.400x_{2(k-1)} - 1.633x_{3(k-1)} + 0.272x_{4(k-1)} -$$
$$- 0.015x_{4(k-2)}; \qquad \delta^2 = 0.8580$$

The neurons selected in self-organization are shown in Fig.6. The equations define the connections that must be implemented in the neural network; in this way, they help achieve the task of structural self-organization of the neural network. For brevity, the data sample in the above example is extended in only one way: tile output variables of the first layer are passed on as additional variables to the second, third, etc. layer of neurons.

We draw the reader's attention to the high accuracy of forecasts for the securities quoted at the Stock Exchange. Undoubtedly, computer self-organization algorithms can have a strong impact on the operation of exchanges. There is one more vital application for neural networks.

Simplified (Shannon) GMDH models, when used as active neurons, can forecast the effects of nuclear tests, thus making unnecessary their actual implementation.

4.2. Neural network self-organization and algorithms for optimization of complex control systems

The principal roadblock in the use of linear and non-linear programming algorithms for complex system optimization is that it is often impossible to specify either the goal function or the applicable constraints with sufficient accuracy. Meanwhile, even minute inaccuracies in their specification may have a strong impact on the outcome of optimization. Active-neuron networks can be readily combined with linear and non-linear programming algorithms.

One of the output functions is taken as the objective function, the equations of the other output variables can serve as equality-type constraints. This removes the subjective factor from the specification of the goal function and constraints. The human operator defines criteria for their choice, and not the objective function and constraints themselves [19,20].

References

1. Zhuravlev, Yu.I. and Gurevich, I.B., Pattern Recognition and Image Recognition*, Pattern Recognition and Image Analysis*, 1991, vol.1, no.2, pp. 149-181.
2. Brusilovskij, P.M. *Kollectivy Prediktorov v Ekologicheskom Prognozirovanii* (Predictor Collectives in Environmental Forecasting), Saratov: Saratov University, 1987.
3. Rastrigin, L.A. and Erenshtein, R.A., *Metod Kollectivnogo Raspoznavaniya* (The Collective Recognition Method), Moscow:Energoizdat, 1981.
4. McCulloch W.S., Pitts W. A Logical Calculus Of Ideas Immanent In Neurons Activity, *Bull. of Math. Biophysics*, 1943, no.5, p.115-135.
5. Herz J.A. Krogh A.S., Palmer R.G. *Introduction to The Theory Of Neural Computation*. Addison Wesley, 1991, p.320.
6. Ivakhnenko, A.G. *Samoorganizirujushiesya Sistemy Raspoznavaniya i Avtomaticheskogo Upravlenija* (Self-Organizing Recognition and Automatic Control Systems), Kiev: Technika, 1969.
7. Ivakhnenko, A.G. and Stepashko, V.S. *Pomekhoustojchivost' Modelirovanija* (Noise lmmunity of Modeling), Kiev: Naukova Dumka, 1986.
8. Ivakhnenko, A.G. and Yurachkovskij, Yu.P. *Modelirovanije Slozhnykh System po Experimentalnym Dannym* (Modeling of Complex Systems from Experimental Data), Moscow: Radio i Svyaz', 1987.
9. Ivakhnenko, A.G. *Nepreryvnost' i Diskretnost'* (Continuity and Discreteness), Kiev: Naukova Dumka, 1990, 222p.
10. Ivakhnenko A.G., Krotov G.I., Cheberkus V.I. Harmonic And Exponential-Harmonic Gmdh Algorithms For Long-Term Predicton Of Oscillating Processes. Part 1. *Sov. J. of Automation and Information Sciences*, vol.14, no.1, 1981, p.3-17.
11. Ivakhnenko A.G., Krotov G.I. Multiplicative And Additive Nonlinear Gmdh Algorithm With Factor Degree Optimization. *Sov. J. of Automation and Information Sciences*, vol.17, no.3, 1984, p.13-18.

12.Ivakhnenko A.G., Sarychev A.P., Zalevsky P.I. Experience Of Solving Solar Activity Forecasting Problems With Precise And Robust Approaches. *Sov. J. of Automation and Information Sciences*, vol.21, no.3, 1988, p.31-43.

13.Ivakhnenko A.G., Petukhova S.A., Ivakhnenko G.A., Yudin V.M., Kovbasyuk S.A. Computer Choice of The Optimal Clusterisation Of The Data Sampling Under Compensation Of Fuzzy Characteristics Of The Object. *Sov. J. of Automation and Information Sciences*, v.26, no.3, 1993, p.46-58.

14.Ivakhnenko,A.G. An Inductive Sorting Method for the Forecasting of Multidimensional Random Processes and Events with the Help of Analogs Forecast Complexing. *Pattern Recognition and Image Analysis*, vol.1, no.1, 1991, pp.99-108.

15.Ivakhnenko A.G., Zholnarsky A.A., Muller J.A. Algorithm of Harmonic Rebinarization of the Data Sampling. Sov. J. of Automation and Information Sciences, vol.25, no.6, 1992, p.34-39.

16.Belogurov V.P. A Criterion Of Model Suitability For Forecasting Quantitative Processes. *Sov. J. of Automation and Information Sciences*, v.24, no.3, 1990, p.21-25.

17.Fedorov V.V. Teorija Optimalnogo Experimenta. *Planirovanie Regressionnykh Eksperimentov* (Theory of Optimal Experiment, Regression Experimental Design), Moscow: Nauka, 1971.

18.Yurachkovsky Yu.P. Analitical Construction Of Least-Variance Discriminating Criterion. *Sov. J. of Automation and Information Sciences*, vol.23, no.2, 1990, p.7-14.

19.Triseyev Yu.P. Approaches to The Solution Of Mathematical Programming Problems On The Basis Of Heuristic Self-Organization. *Sov. J. of Automation and Information Sciences*, vol.19, no.2, 1987, p.30-37.

20.Ivakhnenko A.G. and Ivakhnenko G.A. Simplified Linear Programming Algorithm as Basic Tool for Open-Loop Control. *System Analysis Modeling Simulation (SAMS),* 1995, vol.18-19, pp.315-319.

21.Sawaragi Y.,Soeda T.,Tamura H.,Yoshimura T., Ohe S., Chujo Y., Ishihara H. Statistical Prediction of Air Pollution Levels Using Non-Physical Models. *Automatica (IFAC),* vol.15, no.4, 1979, p.441-452.

22.Ivakhnenko A.G., Muller J.-A. Self-Organization of Nets of Active Neurons *System Analysis Modeling Simulation (SAMS),* 1995, vol.20, pp.93-106.

23.Ivakhnenko G.A. Self-Organization of Neuronet with Active Neurons for Effects of Nuclear Test Explosions Forecastings. *System Analysis Modeling Simulation (SAMS),* 1995, vol.20, pp.107-116.