УДК 681.513.8

NETWORK STRUCTURES ALGORITHMS OF GROUP METHOD OF DATA HANDLING

Kondrashova N.

International Research and Training Centre for Information Technologies and Systems of the National Academy of Sciences and Ministry of Education and Science of Ukraine

NKondrashova@ukr.net

У статті представлена порівняльна характеристика мережевих структур алгоритмів МГУА, відповідно до запропонованої класифікації алгоритмів. Наведено основні різновиди алгоритмів. Структури алгоритмів пошуку моделей дозволяють візуалізувати подібності та відмінності між основними алгоритмами МГУА, показати їх зв'язок з нейронних мереж і алгоритмів глибокого навчання. Структура фрагментів алгоритмів (функцій частинних описів) дає можливість наочно представити їх спільні та відмінні риси і оцінити обчислювальну складність алгоритмів.

Ключові слова: метод групового урахування аргументів (МГУА), мережеві структури, повного і спрямованого перебору, ітераційні і комбіновані алгоритми МГУА, МГУА-подібні нейронні мережі, алгоритми глибокого навчання.

The article presents the comparative characteristic of network structures of GMDH algorithms, in accordance with the proposed classification of algorithms. The main diversity of algorithms is described. Structures of algorithms search of models allow visualize the similarities and differences between the basic GMDH algorithms show their connection with neural networks and algorithms of deep learning. Fragments' structure of algorithms (or functions of partial descriptions) gives you the ability to visualize their common and distinctive features and assess the computational complexity of algorithms.

Keywords: Group Method of Data Handling (GMDH), network structures, exhaustive and directional searching, iterative and combined GMDH algorithms, GMDH-like neural networks, deep learning algorithms.

В статье представлена сравнительная характеристика сетевых структур алгоритмов МГУА, в соответствии с предложенной классификацией алгоритмов. Приведены основные разновидности алгоритмов. Структуры алгоритмов поиска моделей позволяют визуализировать сходства и различия между основными алгоритмами МГУА, показать их связь с нейронными сетями и алгоритмами глубокого обучения. Структура фрагментов алгоритмов (функций частных описаний) дает возможность наглядно представить их общие и отличительные особенности и оценить вычислительную сложность алгоритмов.

Ключевые слова: метод группового учета аргументов (МГУА), сетевые структуры, полного и направленного перебора, итерационные и комбинированные алгориты МГУА, МГУА-подобные нейронные сети, алгоритмы глубокого обучения.

Introduction

Main part algorithms of the Group method of data handling (GMDH) are the search the model structure on the decision tree. The result of problems optimal solution of structural and parametric identification of an object depends on the tree structure algorithm, degrees of freedom at each level, type of the criterion, and determinant of the correlation values matrix of input variables. A feature of wood is the fact that, exactly, constitutes a place of a branching tree. In the sequel we will distinguish between network structure of the algorithm building the final appearance of the models, and tree search function of partial description, which is an intermediate model (or a fragment of the model structure).

1. Main varieties of algorithms

We give a description of algorithms in order of complexity of their networks, rather than in chronological order of their creation. The main algorithms are divided on the search and iterative algorithms.

The search algorithms in turn are divided into exhaustive and directional iterate through the arguments. Among the search algorithms are allocated to combinatorial algorithm of exhaustive search COMBI [1] - [3] and directional algorithm busting MULTI [4], sometimes called a multi-stage. Searching of arguments may be carried out when you include them in the model, or delete from the full model (containing all possible arguments). The search algorithms of sequential inclusion and exclusion are proposed in [5], [6].

The iterative algorithms include relaxation [7] - [9] and multilayer [10] algorithms. Combined are obtained as a result of combining under one interface relaxational and multilayer iterative [11] and "crossbreeding" relaxation structure with the structure of the multilayer algorithms [12]. The most complex network structure has GMDH-similar neural networks [13] - [15]. They use external addition to determine the optimal number of balls, eras, layers, sites, and have for partial descriptions, in addition to a differential polynomial function more complex functions, called neurons.

In accordance with the above, we will consider network algorithms, whose classification scheme is presented in Fig. 1.

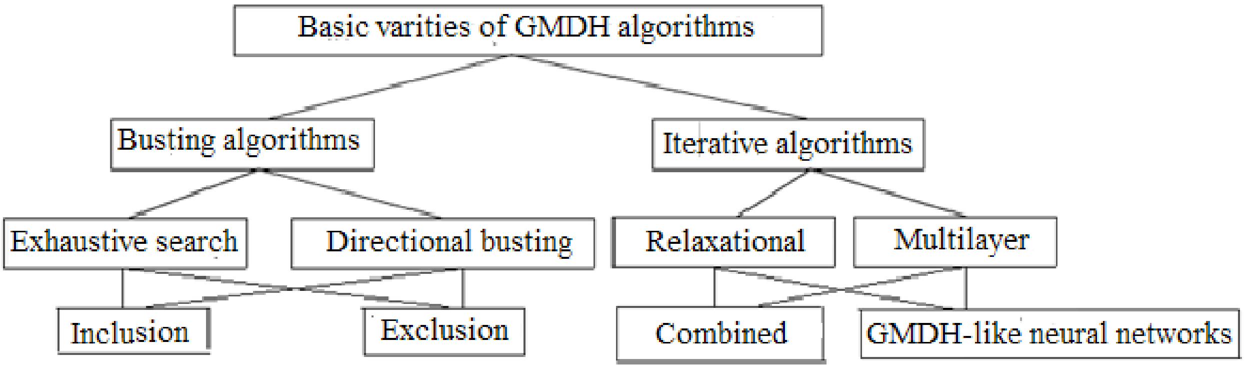

Figure 1. - Classification of the basic varieties of GMDH algorithms

By using GMDH most often are building models, linear in the parameters, but initial variables can be included non-linearly. An example is polynomial of all the variables (the so-called Kalmogorov-Gábor polynomial):

$$y = f(x_1,...,x_m,\Theta) = \theta_0 + \sum_{i=1}^{m}\theta_i x_i + \sum_{i=1}^{m}\sum_{j=i}^{m}\theta_{i,j}x_i x_j + \sum_{i=1}^{m}\sum_{j=i}^{m}\sum_{k=i+j}^{m}\theta_{i,j,k}x_i x_j x_k + ... \quad (1)$$

where $x_1,..,x_m$ are input variables. Estimated parameters vector Θ= $(\theta_0,..,\theta_{i,j},...,\theta_{i,j,k},...)$ and y is an output variable. A special case of a nonlinear variable model (1) is a linear model by parameters and initial variables:

$$y = \sum_{i=1}^{m}\theta_i x_i . \quad (2)$$

All classical algorithms when you are building models of GMDH use three basic principles: 1) the principle of consecutive complication of models in the process of finding the best one of the specified class; 2) principle of not final solution or freedom of choice; 3) principle of external add-on.

Functions $f_k(\mathbf{v},\Omega)$, defined with a precision of parameter vector Ω values, are called functions of partial description (FPD) in algorithm's network nodes. The freedom of choice of the algorithm F_i refers to the number of selected "promising" edges of the graph (network paths) algorithm. Degree of freedom F_i number intermediate functions of partial descriptions that continue a complication from a layer to a layer.

The principle of gradually increasing the complexity of the structure of the model is also based on the idea of self-organization of constructing models of such complexity, which is necessary to adequately describe the object in the context of a minimum specified criterion *CR* quality models. Model structure generation algorithm carries out a number of successive complications. At each stage, or iteration *(r* = 1, 2...) is generated by a model-applicant with a more complex structure than the previous ones.

In various algorithm stop occurs when one or both of the following conditions are satisfied if: 1) finished exhaustive search of all the models specified class structures; 2) value of an external criterion after its fall begins to grow; 3) criterion value after its fall is beginning to change within the specified limits the accuracy of the model; 4) number of iterations reaches beforehand to a specified value.

2. Network structures of the searching algorithms

Classic GMDH algorithm COMBI of exhaustive searching [1] - [3] builds the view model (1). For an analysis of the algorithm model (1) re denotation variables be reducible to (2) with more input variables $l_p > m$, that depends on the number of multiplications variables or total maximum degree p_{max} in terms of a polynomial (1). For example, $l_p = C_{m+p_{\max}}^{p_{\max}} = C_{2+2}^{2} = 6$ the assumption that p_{max} = 2 and m =2, the values of the variables: x_1, x_2.

In the searching algorithm COMBI is carried out the exhaustive busting possible structures, are built all the trees (forest) functions of partial descriptions of the algorithm and the forest in this case coincides with the network tree of the algorithm.

Without loss of generality the further description of the algorithm is given in the case of the search model structure corresponding to the results of exhaustive searching in a specified class of linear models. In Figure 2 shows tree for the model (2) when m = 4, representing the result of exhaustive searching for the set of its structures. The degree of freedom for each i-th level of the tree varies and is equal to $F_i = C_m^i$. The use of the prefix tree [16] does not allow repetition and the skip of monomial when forming a complete set of multinomials for polynomial form (2). In the places of branching (in nodes) of a tree shown the names of the variables. To obtain each of the k structures need to pass the path from the root to a specific node $k = \overline{1,(2^m - 1)}$. Different variants of structures are received by the combinatorial busting of different paths from the root to the vertex of the graph. In each place of branching in the tree to the previous set of variables are added alternately variables in the current node, i.e. the algorithm is working on the inclusion of variables with no repeats.

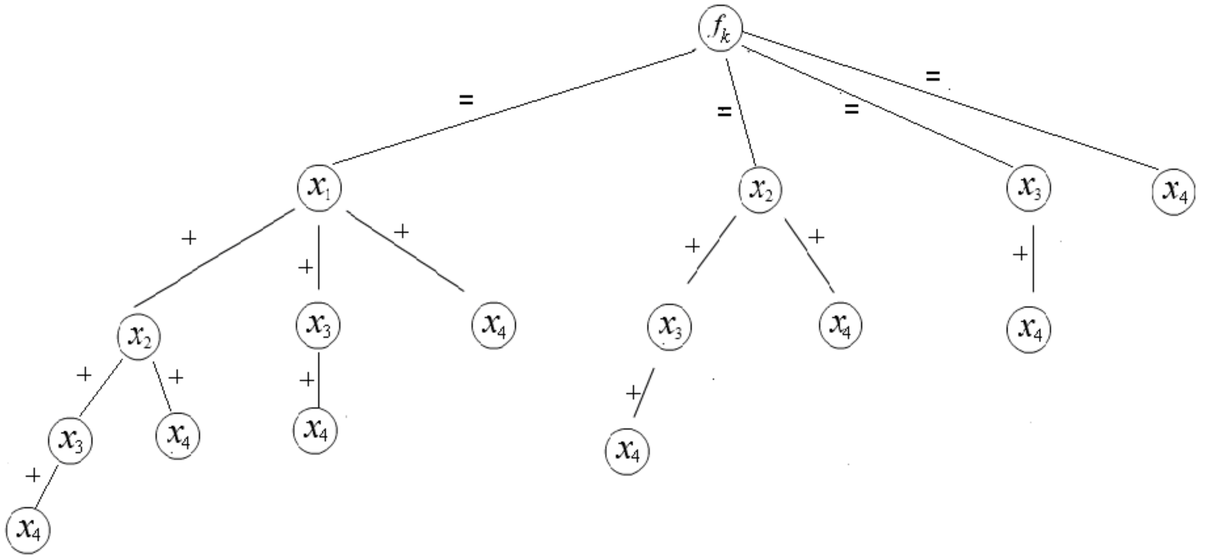

Figure 2. An image on a tree of all possible variants of model structures (2) when m = 4 ("+" means adding, "=" means equal).

Growing tree roots (in other words, "inverted tree") similar to a complicated tree structure of the algorithm of the search model. Depicted in Figure 2 tree illustrates the process of complication of models structure by way nested structures, if you descend from the root along any tree branch.

Regardless of the algorithm (the decision), at the outset, is specified class of model structures. Most often this is a class of polynomial functions of the form (1), finite-difference or polynomial-difference equations of linear by parameters where, where $\mathbf{v} \in \mathbf{X}$ is any vector-row of input variables matrix $\mathbf{X} = (\mathbf{x}_1,..,\mathbf{x}_m)$, $\mathbf{v} \in \Re^{n_w}$ Θ is a vector of parameters.

The purpose of the GMDH algorithm of directional busting find a solution that matches for the result of an exhaustive search of variant models with some not zero probability. The first algorithm busting with the consistent inclusion of arguments is the algorithm MULTI [4], which uses a truncated directed busting of decisions. It works as follows: first built, many models (1), consisting of a single argument, the sequel of the two, and so on up to the model that contains all l_p arguments. At every stage, selected a specified number of $F_s = l_p$ the better models (s is stage number), etc. to each of the best models of the previous step add one of the missing arguments in it. Note that at $F_1 = l_p$ in the first selection stage models for algorithms of inclusion ensures that global minimum will not be lost in the first stage (assuming that all l_p structures is different), while at other intermediate stages-when $F_s = l_p$ there is no such guarantees.

For example, at $m = 5$ the view of the tree for MULTI algorithm when built model (2) will differ from wood COMBI depicted in Fig. 2, by fewer branches. For example, if the at degrees of freedom $F_i = m = 5$ on the second row of branching by external criterion will be selected as the best models $y = x_1x_2,\ y = x_1x_3,\ y = x_1x_4,\ y = x_1x_5,\ y = x_2,$ then from busting will be permanently excluded models $y = x_2x_4x_5,\ y = x_3x_4x_5$.

Described GMDH algorithms busting (COMBI and MULTI), as has been said, are the algorithms of inclusion. In [5] describes an algorithm for directional busting, named BSS (Backward Successive Selection) that implements the idea of sequential exclusions spurious arguments. He uses as constituent the exhaustive search with exception of arguments (algorithm COMBIS). The algorithm BSS in comparison with the algorithm directional busting FSS (Forward Successive Selection) consistent inclusion argument that is used as a part exhaustive search of arguments with their inclusion, showed the best search results of sets true arguments in the task of identifying "the true" (unbiased) models with the frequency criterion of selection informative arguments.

To imagine a tree of algorithm exception, we may use wood of subtraction, if from model structures, containing in each stage and at each node are all m arguments deduct corresponding structure tree busting inclusion arguments. For $m = 4$ will get "a lot of bushes’’, depicted in Figure 3.

The image of this set of a graph has a view of an inverted at 180° the tree shown in Figure 2. In it paths of length (m-1),.., 0, which going upstairs and starting from the last level (where the vertices are labelled as x_i) will display a set of structures of models from zeroth to (m-1) stage in the algorithm of the exception. In Figure 3 it shown. It should be noted that the algorithms FSS and BSS have implement ideas of algorithms exhaustive and directional busting inclusion and exclusion. They have more complicated structure of inclusion and exclusion algorithm arguments using an interim exhaustive search in specified limits of variation of arguments number and frequency selection criterion when you exclude and include.

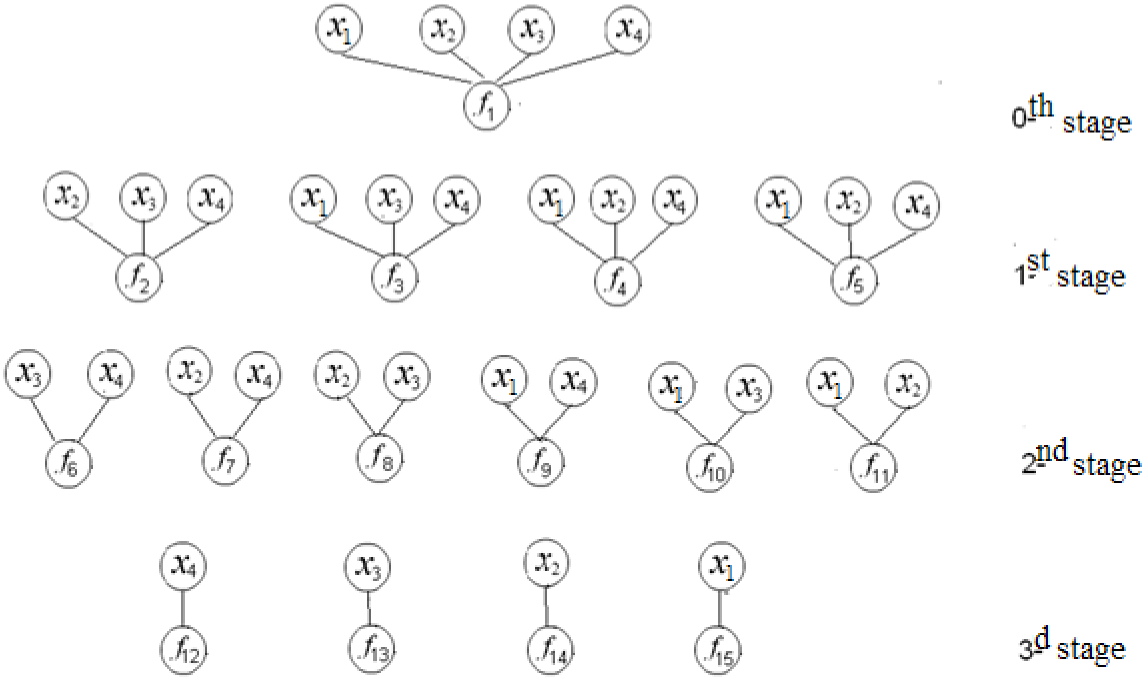

Figure 3. Trees of algorithm exception COMBIS, with exhaustive search of arguments (*m*= 4).

Algorithms classification scheme in Fig. 1 do not contain an obvious hybrid of search algorithm for inclusion-exclusion (similar to the regression method of inclusion-exclusion). It is named the algorithm of combined successive selection (CSS) [6].

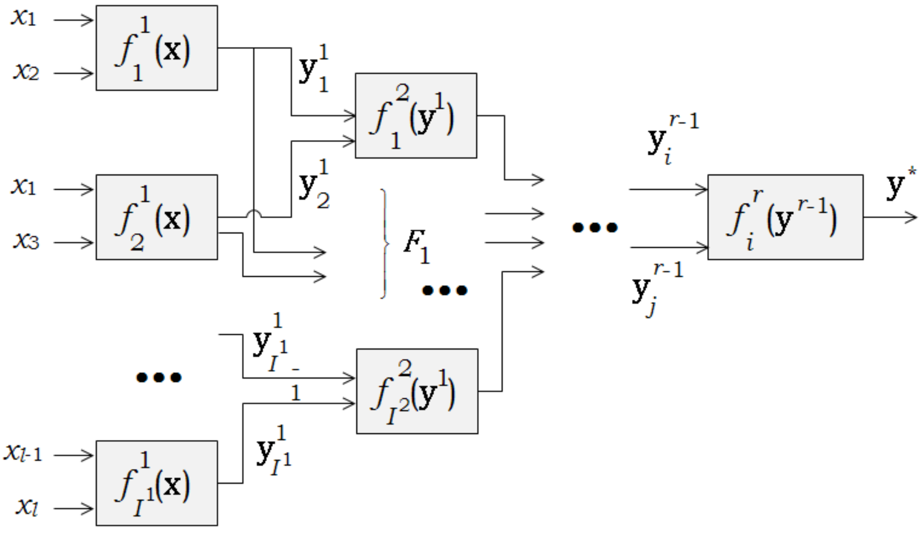

Figure 4. Scheme of successive complicated algorithm of model structure by MIA

Treelike network of the first multi layered iterative algorithm (MIA) [10] displays the build process of function of many variables as the superposition of a sum of functions of two variables. Complication of the model structure by MIA has the configuration tree. Schematic representation of the MIA GMDH construction tree network is presented in Figure 4

In MIA the model (1) as a superposition of functions of two variables (x_i, x_j) or (y_i^r, y_j^r) is presented in a treelike network, where in places branching of tree are neural-type elements. The main features of these elements are the summation with the weights of the input signals and presentation of conversion result according to some function. An important step is the interim selection decisions between iterations. For the selection of variables and their number in the first algorithm GMDH [18] applied thresholds, i.e. comparison of specified values with the values of the coefficients of correlations of deviations output variable and *i*-th input variable with a shift on a finite number of steps calculated over the whole sample.

3. Network structures of relaxation iterative algorithms

The first relaxation iterative GMDH algorithm is a multilayer simplified algorithm (MSA) [7], which builds the FPD like:

$$\mathbf{y}^0 = f_0(q(\mathbf{X},\omega) = \omega_0 q(\mathbf{X}), \qquad (3)$$

$$\mathbf{y}^{r+1} = f(\hat{\mathbf{y}}^r, \mathbf{X}, \Omega) = \omega_1 \hat{\mathbf{y}}^r + \omega_2 q(\mathbf{X}), \quad r \geq 0, \qquad (4)$$

where r is number of additive iterations, and the matrix of the initial variables x_i can be extended by reciprocal variables $1/x_i$. Denote zeroth iteration the additively-multiplicative monomial as $q(\mathbf{X}) \overset{\Delta}{=} \mathbf{x}_i$ and for extended matrix may be $q(\mathbf{X}) \overset{\Delta}{=} 1/x_{ij}, \quad \forall\, j = \overline{1, n_W}$, and multiplicative monomial of a higher order, as $q(\mathbf{X}) \overset{\Delta}{=} \mathbf{z}_k$, where the elements of the vector $\mathbf{z}_k$ calculated as the product

$$z_{jk} = \prod_{i=i_k}^{p} x_{ij}, \; j = 1, n_W, \quad f(\mathbf{z}_k, \hat{\mathbf{y}}^r, \Omega), \; r > 0,$$

$$i_k \in \{1,2,\ldots,m\}, \; k \in \{1,2,\ldots,m\}, \; p \in \{1,\ldots,p_{\max}\}.$$

The Figure 5 shows the MSA. The zeroth iteration model consists of busting through all models containing a single argument. The first stage of the build multiplicative monomial of the second order shows a fragmentary for some models of zero multiplication iteration. Other iterations of the algorithm are given without detail. Freedom of choice on the zero iteration $F_0 = m$ and for extended matrix it is $2m$. In MSA the definition of argument inclusion in the additive-multiplicative model realized by using of direct combinatorial busting options constructed multiplicative monomials.

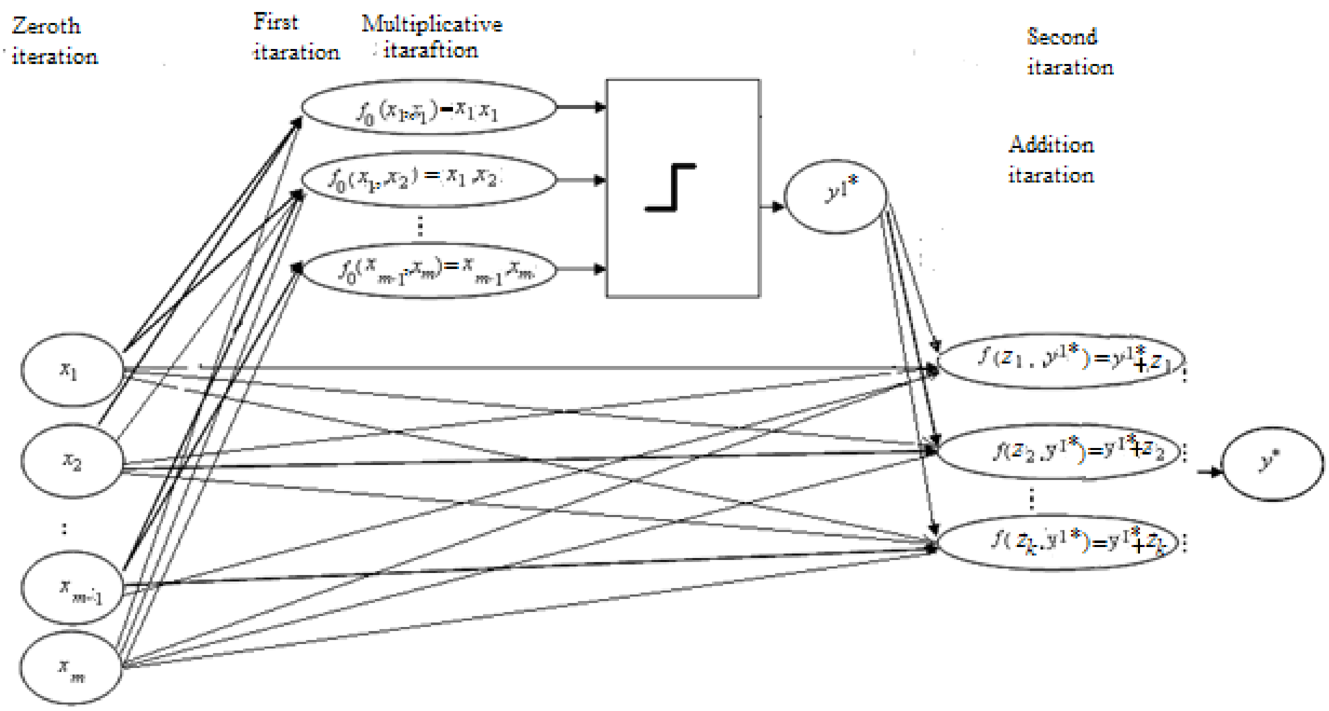

Figure 5. Network structure multilayered simplified algorithm GMDH.
_|‾ indicates the threshold selection of the best model, F_1 = 1.

Diagram of the tree, illustrating the multiplicative monomial searching, is presented in Figure 6. In the algorithm implemented directional busting at the current iteration multiplication. Choosing F_1 = 3 best models implemented through breeding one best model- child from each parent-model, the number is equal to m =3. On the latest iteration of the multiplication is selected best model, $q(\mathbf{X}) = x_1^2 x_2$, F= 1.

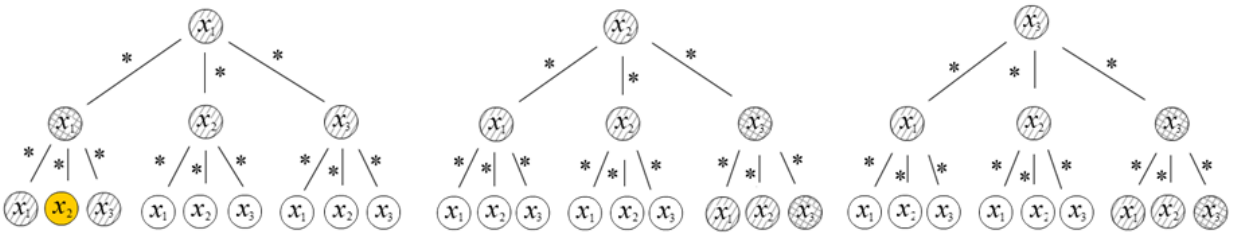

Figure 6. Illustration of multiplicative monomial search in trees
"*" is multiplication sign; x_1 denotes the calculated monomial; x_1 is not calculated monomial; x_1 is the best the monomial in a multiplicative iteration; x_2 is the best monomial on the second iteration of multiplication, with which need to enter term in the multiplicative model (r + 1)-*th* additive iteration. (Scheme from [9]).

The main advantages of MSA algorithm are that it does not lose the initial basis of variables when truncated busting and each layer has a description of the model outputs, represented in the initial input variables. Given these advantages, it was chosen as a basic algorithm for further improvements in generalized relaxation iterative algorithm (GRIA) [9].

Zero GRIA iteration coincides with the MSA. Illustration of multiplicative monomial searching by GRIA with the help of the trie shown in Fig. 7. Images of trees are minimal (has no repeats). Due to this performance calculations number in GRIA compared to MSA decrease from 40% to 20% approximately.

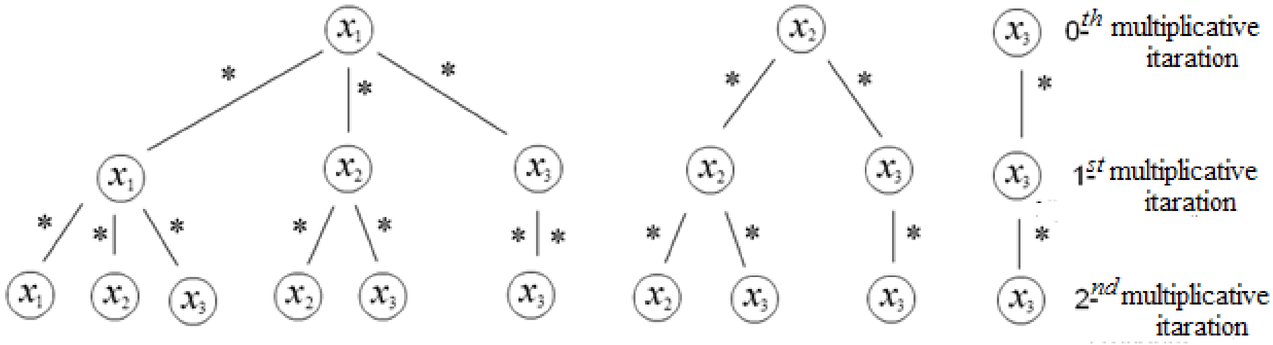
x_1
x_2
x_3
*
0^{th} multiplicative itaration
1^{st} multiplicative itaration
2^{nd} multiplicative itaration

Figure 7. Illustration of search structure monomials of multiplicative model using trie. (Scheme from [9])

Pay attention to the shape of the minimum tree of exhaustive search result multiplicative monomials. It has an expanding crown, "clipped" at the end on one level, in contrast to the minimal tree additive monomials shown in Fig. 2. When searching for set of variants exhaustive search of its elements defined by enumeration and united under the signs summation (+) or (∪), search tree has the shape ranked by quantity, "hanged out" on one level of "bunch of grapes". Advanced method of monomial search using the proposed algorithm bypass trie, in contrast to the MSA, it is more flexible. Since it allows to vary, as the number of best models *F*, passing to the next additive iteration of the algorithm, and the number of the best models of F_1, which passing to the next multiplicative iteration. If to the next iteration pass all generated models, then it carried out an exhaustive search. The degree of freedom for the multiplicative iteration will depend on the number *i*:

$$F_0 = C_m^1 = m, \quad p = 1 \quad ,$$

$$F_i = C_{m+i}^{i+1} = C_{m+i}^{m-1}, \quad 0 < i \le p_{\max} - 1,$$

If you set $F = F_{\mathrm{i}} = 1$, where $0 < i \le p_{\max} - 1$, and $F_0 = C_m^1 = m$ the tree traversal algorithm in GRIA of directed search algorithm is similar to the construction of the model structure in the MSA, and for $p_{\max} = 2$ the set of busting models structures of these algorithms are congruent. The advantage of the network structure in the speed and accuracy of the model GRIA compared with MSA appears at $p_{\max} > 2$.

4. Network structures of partial descriptions for varieties of multilayered iterative algorithm

In [18] in MIA "with covariations and quadratic description" is offered not only to calculate the functional dependence of the form

$$f_0(x_i,x_j,\Theta)=y_k^0=\theta_0+\theta_1x_i+\theta_2x_j+\theta_3x_ix_j+\theta_4x_i^2+\theta_5x_j^2, \qquad (5)$$

$$f\,(y_i^{r-1},y_j^{r-1},\Theta)=y_k^r=\theta_0+\theta_1y_i^{r-1}+\theta_2y_j^{r-1}+\theta_3y_i^{r-1}y_j^{r-1}+\theta_4(y_i^{r-1})^2+\theta_5(y_j^{r-1})^2, \quad r\ge1, \qquad (6)$$

and apply an exhaustive search of functions of partial descriptions of the general form:

$$f_i(v_1,v_2)=\omega_0+\omega_1v_1+\omega_2v_2+\omega_3v_1v_2+\omega_4v_1^2+\omega_5v_2^2 \;. \qquad (7)$$

Imagine that tree on Fig.8 a) is a binary. In his nodes are two states 0 or 1, where the unit corresponds to the presence of variable vi, and zero its absence. Then the sum of all the paths on this binary tree equals the number of structures FPDs in the result of an exhaustive search in a node of the treelike network of the algorithm, $K=2^{l-1}-1=$ 31. Redenote (l-1) monomials in (7) ($\omega_0=0$) as x_i. On the tree of structure search shown in Figure 8 b), convenient to show distinction of the algorithms described in [18] and [19]. In [10] used a shortened search

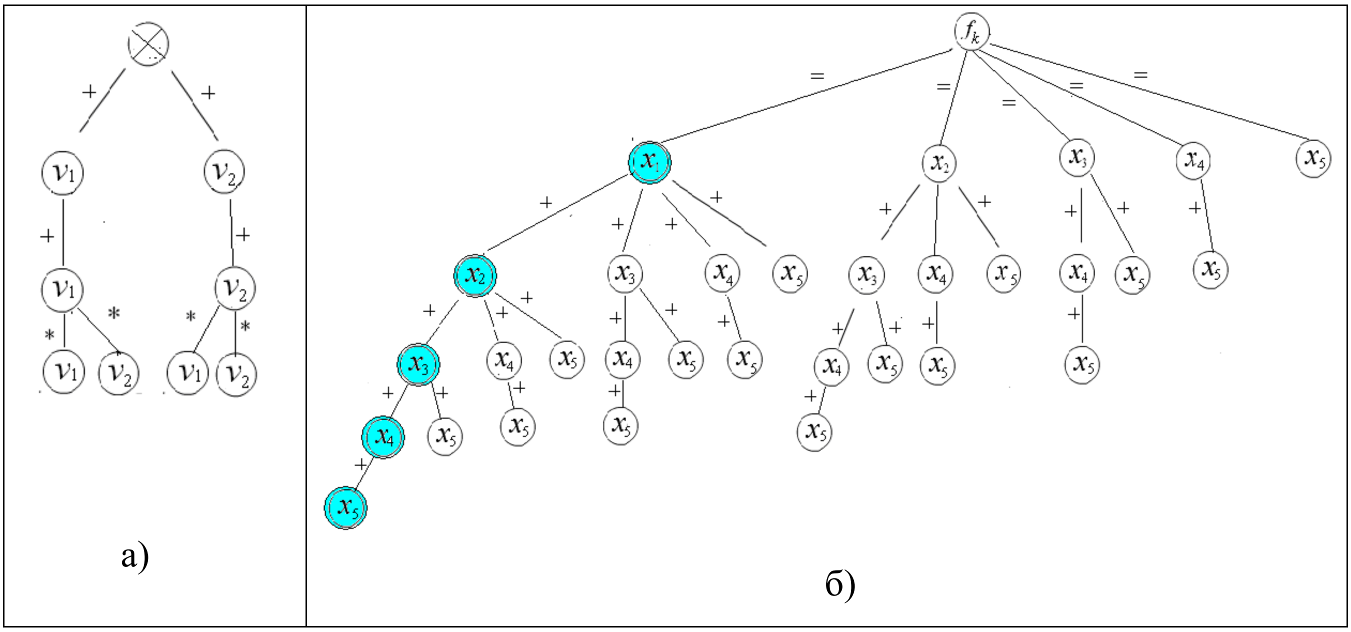

Figure 8. Variants of trees for FPDs of MIA "with covariations and quadratic descriptions": a) on the binary search tree; b) search for a conventional tree (shortened search allocated darkened circles). The "+" and "*" denote addition and multiplication.

. Lots of structure equal to five obtained by the method of nested structures. The shortened search marked by shaded circles shown in the tree in Fig. 8 b). Exhaustive search of structures models on possible paths tree [18] where m = 5, is reduced in [19] to a single branch, which is the longest path.

Is necessary to note that it known algorithms that use similar trees, where instead of the summation sign "+" and multiplying "*" used other operators, such as "$\vee, \wedge$" [20].

The capabilities of GMDH algorithms can be expanded by using additional operators "$\cup, \cap, \otimes, \oplus$."

5. Network structure of the combined algorithms

In [12] a polynomial algorithm GN (with Gödel numbering) in which the functions of partial descriptions in general case depend on three variables:

$$f(v_1, v_2, v_3, \Omega) = \omega_1 v_1 + \omega_2 v_2 v_3, \qquad (8)$$

where v_i is the element value in a row-vector **v**. Input variables $v_i \in \mathbf{z}$ can take the following values:

$$v_i \in \{x, \hat{y}, 0, 1\}, i = \overline{1,3},$$

where $x \in \mathbf{x}_j$ and $\mathbf{x}_j$ is column-vector of the matrix **X**; $\hat{y} \in \hat{\mathbf{y}}_k^r$ and $\hat{\mathbf{y}}_k^r$ is column-vector of the matrix $\hat{\mathbf{Y}}^r = (\hat{\mathbf{y}}_1^r, ..., \hat{\mathbf{y}}_F^r)$; $k = \overline{1,F}$, $r = 1, \ldots, R$.

In [8], [12] containing the description of the GN-algorithm neither the network structure of the algorithm, nor wood FPDs not shown. We represent the look of the search tree FPDs, according to the description of the algorithm in [8].

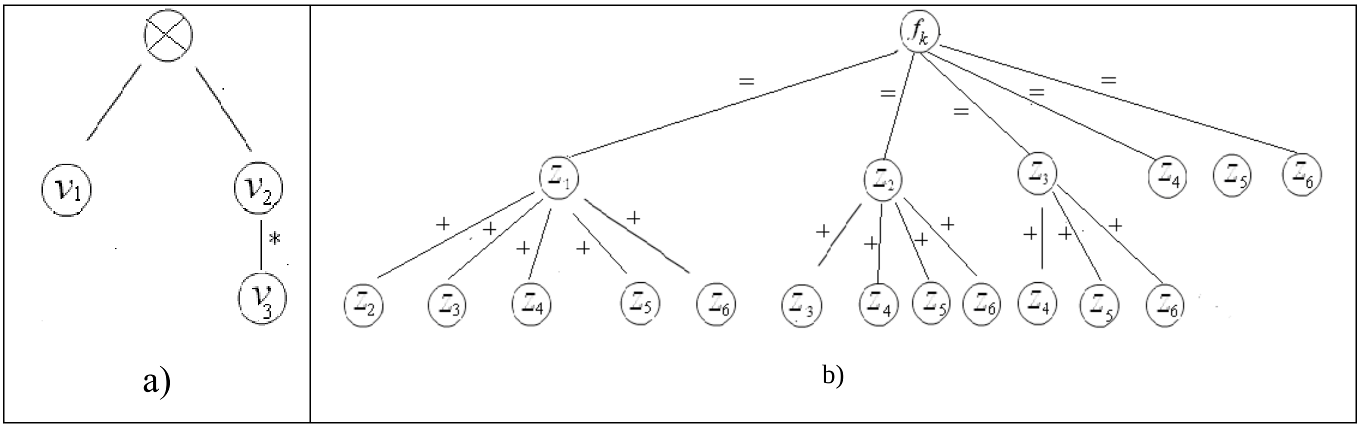

Figure 9. Functions partial description of the type (8) a) binary tree; b) construction of the tree of all variants FPDs

Let the freedom of choice is 1, the set of inputs and the set of outputs to r-th iteration consists of one output and one input x_i, i.e. l = 2. Indices at $\hat{\mathbf{y}}$ is omitted. The number of no more p_{max}=2 degree monomials in model (7) is calculated by a known formula $l_p = C_{2+2}^2 = 6$ provided that the variables can take two values (y or x_i). We receive six variants of the polynomials with two monomials from (7) in the form (8) like:

$$\left.\begin{array}{l}\omega_1 x_i+\omega_2 x_i^2\\ \omega_1 x_i+\omega_2 x_i\hat{y}\\ \omega_1 x_i+\omega_2 \hat{y}^2\\ \omega_1\hat{y}+\omega_2 x_i^2\\ \omega_1\hat{y}+\omega_2 x_i\hat{y}\\ \omega_1\hat{y}+\omega_2\hat{y}^2\end{array}\right\} \qquad (9)$$

If we consider that the input variables $v_i \in \mathbf{z}$ can take a single value, it is necessary to add to the set (9) model of view $\omega_1+\omega_2 f_p(x_i)$, $\omega_1+\omega_2 f_p(\hat{y})$, $\omega_1+\omega_2 x_i\hat{y}$, where f_p is power function is not higher than the second order ($0<p\le 2$). Given the zero values $v_i \in \mathbf{z}$ need to add a lot of models that include only one argument. It is easy to see that the number of possible structures the models of obtained set (9) with all of the specified additions coincides with the sum of the number of combinations of the two, and one each from the following polynomial

$$f_i(x_i,\hat{y})=\alpha_0+\alpha_1 x_i+\alpha_2\hat{y}+\alpha_3 x_i\hat{y}+\alpha_4 x_i^2+\alpha_5\hat{y}^2, \qquad (10)$$

except of several of combinations. From (10) it is clear that the algorithm GN is the first combined iterative algorithm, as it gets FPDs not only with covariances carried out not only combinatorial busting $f_0(x_i)$ analogous to (5), and $f(\hat{y})$ similarly to (6), some functions $f_i(x_i,\hat{y})$ (4) of relaxation iterative algorithm, but were built variants of structures FPDs with products $x_i\hat{y}$. Since FPD has form (8), it is excluded in the calculation of exhaustive search three models for which the sum of degrees exceeds 3:

$$f_i(x_i,\hat{y})=\alpha_3 x_i\hat{y}+\alpha_4 x_i^2,\ f_i(x_i,\hat{y})=\alpha_3 x_i\hat{y}+\alpha_5\hat{y}^2\ ,\ f_i(x_i,\hat{y})=\alpha_4 x_i^2+\alpha_5\hat{y}^2. \qquad (11)$$

Views of FPDs are shown in Fig. 9 a). Algorithm with linear partial description, which is called converged multilayered (CML) algorithm is a special case of GN algorithm, if the column vector corresponding to the variable v2 or v3, can be equated to the vector-column of units, **1**, respectively, then $f(v_1,v_3,\Omega) = \omega_1 v_1 + \omega_2 v_3$ or $f(v_1,v_2,\Omega) = \omega_1 v_1 + \omega_2 v_2$. Algorithm CML also is the special case of algorithm MDR, if put in (3) $q(\mathbf{X})=\mathbf{x}$, $\mathbf{x}\in\mathbf{X}$. The "internal convergence" is proved for CML [8].

Advantages of the GN-algorithm are: 1) the construction of nonlinear polynomial models; 2) control of complexity at each iteration; 3) a simple representation of the model in expanded form; 4) the presence of the initial basis (variables of the matrix **X**) at each iteration; 5) the complexity of the algorithm according to the number of variables and the degrees of freedom less than five.

Consider the structure of generalized iterative algorithm GMDH [11].

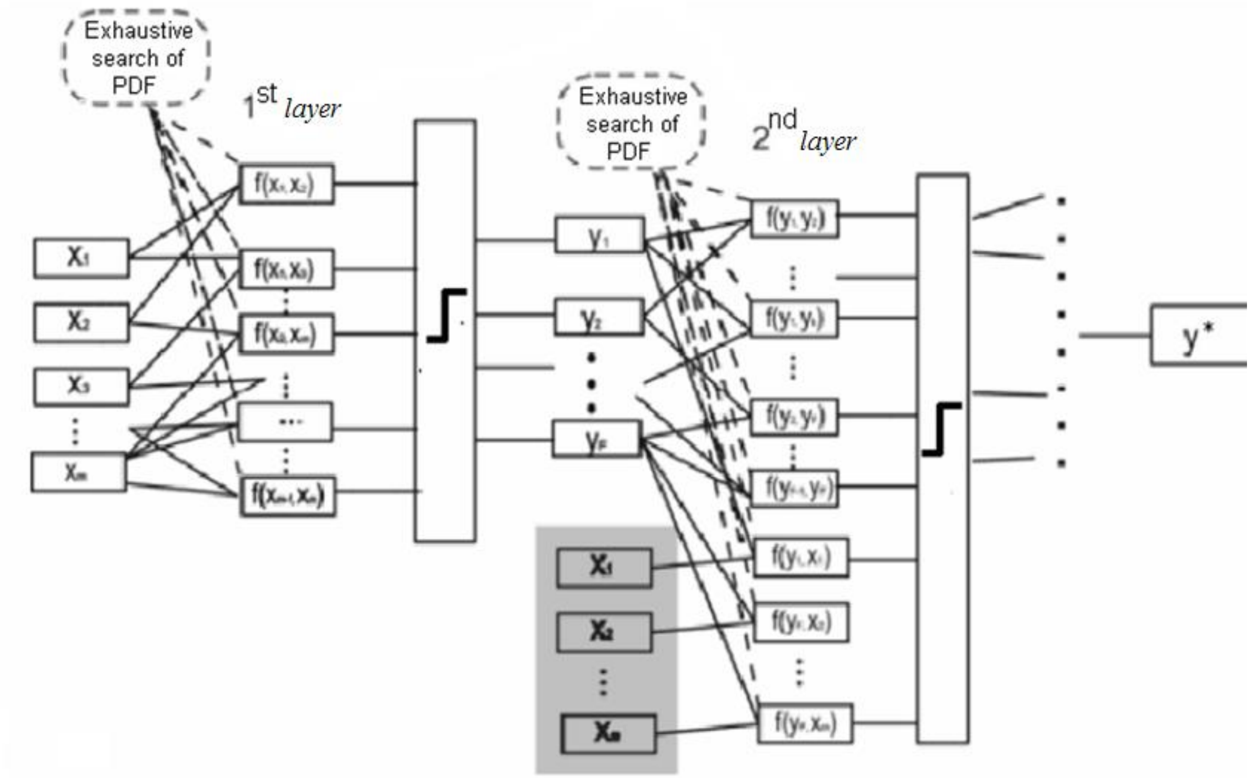

Figure 10. Network structure of the generalized iterative algorithm.
(ʃ is a sign of the selection threshold of *F* better models). (Scheme from [11])

This algorithm has the most branched network and largest volume of calculations in an exhaustive search of FPDs, combines basic types of networks of the above described algorithms: MSA, MIA, and MIA with combinatorial an exhaustive search of FPDs. The substructures of individual algorithms (COMBI, MIA with covariances and quadratic descriptions, the MSA with exhaustive search of covariances and quadratic descriptions) may be extracted from total network structure. Its main disadvantage: it has a low speed.

6. Structures of GMDH-similar neural networks

Structure of *fully automated knowledge extraction using a group of adaptive model evolution* (FAKE GAME) was implemented in [14] (see Fig. 11). The algorithm has the structure of a multilinked network (graph) (there are connections not only between successive layers, but also through layers). Also at the branch points of a pyramidal network GAME unlike MIA are busting not only functions of (5), (6) and

(4), but may be: Gaussian, logistic function and other functions inherent to the nodes of the neural network, therefore, the algorithm refers to GMDH-like neural networks. At the input of nodes may more than two variables, besides search algorithm is implemented, characteristic for the genetic algorithms. Nodes are active neurons (exhaustive search and the genetic selection of the arguments composition).

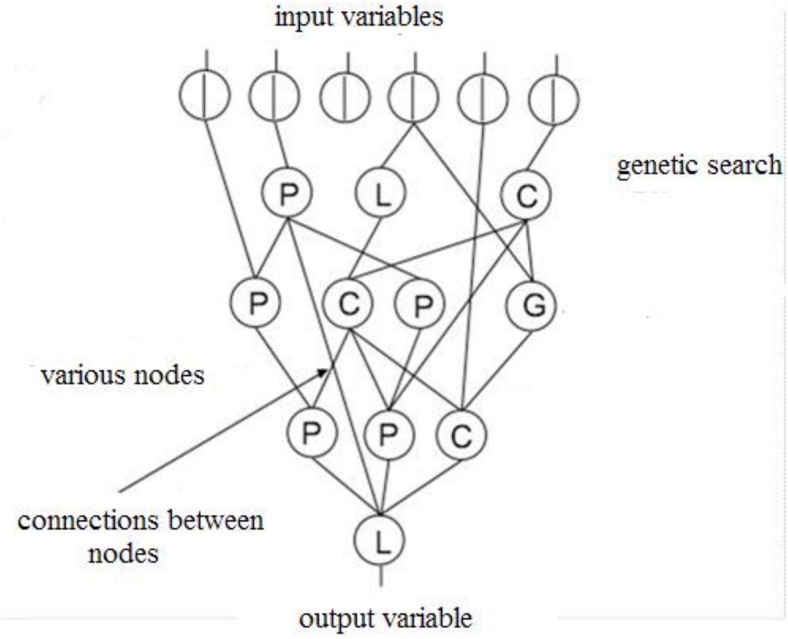

Figure 11. Structure of FAKE GAME: Ⓟ is polynomial FPD; Ⓖ is Gaussian FPD; - combi-neuron FPD; Ⓛ is linear FPD. (Scheme from [14]).

In [15] is proposed dynamic GMDH-similar wavelet Neuro-fuzzy system a data mining with the partial description functions, having the form shown in Fig. 12 A) and B).

Partial description function A) in the general case has the form:

$$\varphi = (1-\alpha_v(\tau)\tau_y^{s_{ij}}(x_i(k)))\exp(\tau_{ij}^2(x_i(k)/2)\,,$$

where

$$\tau_{ij} = (x_i(k)-c_i(k))\sigma_{ij}^{-1}(k)\,.$$

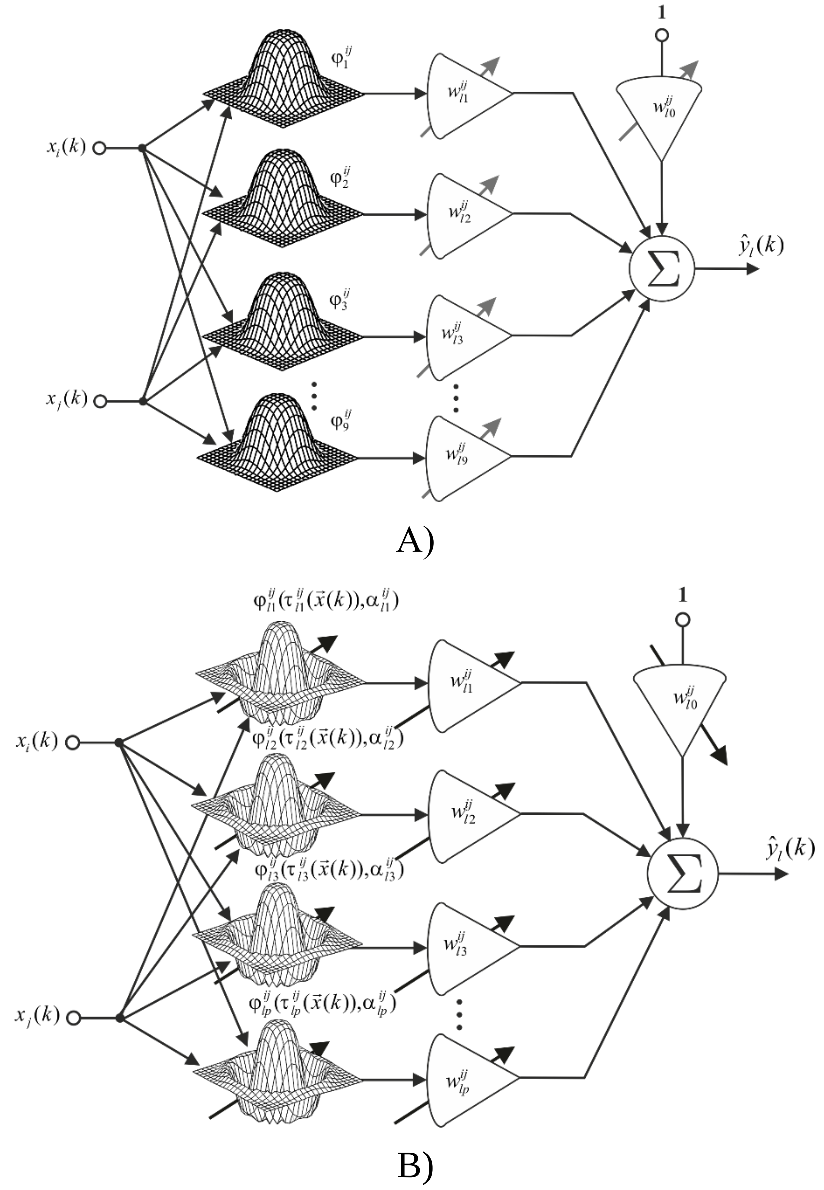

Figure 12. Varieties partial descriptions functions: A) a radial basis functions of FPD; B) FPD having the form of a Mexican hat. (Scheme from [15])

The parameters, which nonlinearly included in the functions of partial descriptions of the network, are adjusted iteratively using a gradient descent algorithm, and linearly included are adjusted by the method of least squares. In Figure 13 is represented a GMDH-similar adaptive neural network built by image and likeness a MIA GMDH (see. Fig 4), wherein as partial descriptions one of the varieties FPDs on two inputs shown in Fig. 12, are used. The number of network elements, namely of FPDs and their tuning, as well as the number of layers are determined by external criterion by adding new observations.

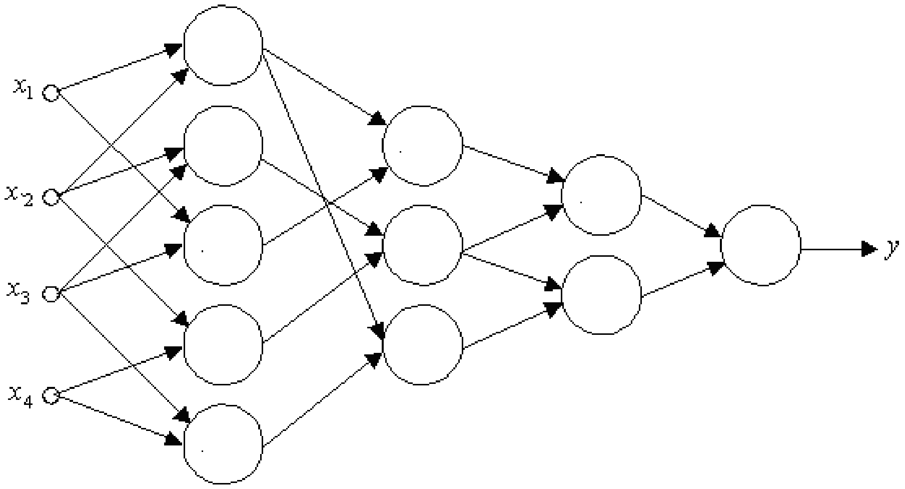

Figure 13. Scheme of GMDH-similar neural network.

Circles denote FPDs on the base of the radial basis network (RBN) or more complex functions. (Figure from [15]).

Last years, rapidly developing the area of deep learning, briefly deep neural networks (DNN) [20], [21]. Scheme of these type algorithms for classifier building is presented in Fig.14. Firstly learns features, use encoder to produce features and train another layer, etc. Overfitting excluded since features are learned in an unsupervised way. Number of layers is given and is limited by computer capacity. After that the identification of weights in every layer RBN of the deep learning is realized by method of gradient descent.

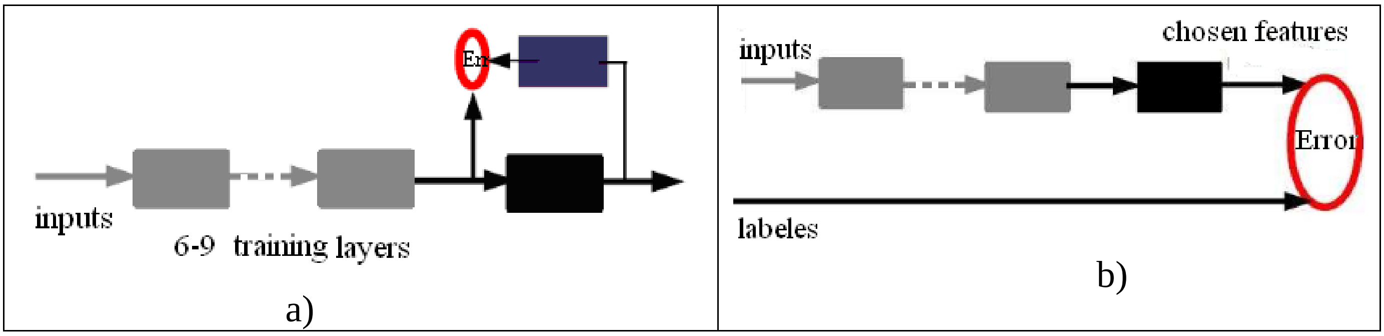

Figure 14. Scheme of deep neural network for classification of images a) stage of encoding features; b) unsupervised way of overfitting exclusion. (Figure from [22])

It’s easy to see these algorithms have a deep historical connection with GMDH [23], [24].

7. Conclusions

Presented structures of algorithms and search trees FPDs of the models allow visualize the similarities and differences between the basic GMDH algorithms, show their connection with neural networks and networks of deep learning, perform a further analysis of their computational complexity, and fuller to identify additional reserves to increase the speed and accuracy of the results.

References

1. Ivakhnenko N.O., Kvasko M.Z. Combinatorial GMDH algorithms, wherein are tested on the regularity of as "symmetrical" and "asymmetrical" polynomials. *Avtomatika*. Kiev, USSR. – 1972. – № 5. – p. 42-48. (In Ukrainian).
2. Stepashko V. S. A combinatorial GMDH algorithm with an optimal scheme busting of models // *Ibid*. – 1981. – № 3. – p. 31-36. (In Ukrainian).
3. Stepashko V.S. Potential noise immunity modeling combinatorial algorithms GMDH without using information about noise. *Ibid*. – 1983. – № 3. – p. 18-27. (In Russian).
4. Ivakhnenko A. G., Stepashko V.S. Noise-immunity of modeling. *Naukova dumka*, Kiev, USSR. – 1985. – 214p. (In Russian).
5. Samoilenko A. A., Stepashko V. S. Analysis effectiveness of using frequency criterion for consistent screening of algorithms' noninformative arguments. *Inductive modeling of complex systems. Col. of Sc. works. IRTC IT&S NASU*, – Issue 4. – Kiev, Ukraine. – 2012. – p. 198-205. (In Ukrainian).
6. Samoilenko A.A. Weight criterion for determining the informativity of the arguments in the methods of constructing models with a consistent selection of variables. *Upravlyayushchiye sistemy i mashiny (CS&C)*. NASU. IRTC IT&S. Institute of Cybernetics. 2013. – N 2. – p. 33–39, 46. (In Russian).
7. Shelud'ko O.I. Self-organization of mathematical models for solving some problems of reliability and control: Dis. ... Cand. Tehn. sc.: 05.13.01 – *Institute of cybernetics*. Academy of Sciences of Ukraine. Kiev, USSR. – 1975. – 166 p. (In Russian).
8. Ivakhnenko A.G., Yurachkovsky Ju.P. Modeling of complex systems on experimental data. *Radio and Communications*. Moscow, USSR. – 1987. – 120 p. (In Russian).
9. Pavlov A.V. The generalized relaxation iterative GMDH algorithm. *Inductive modeling of complex systems. Col. of Sc. works*. NASU. IRTC IT&S, – Issue 3. – Kiev, Ukraine, 2011. – p. 95-108. (In Russian).
10. Ivakhnenko A.G. Systems of heuristic self-organization in technical cybernetics. – *Tekhnika*, Kiev, USSR. –1971. – 372 p. (In Russian).
11. Bulgakova A.S. Generalized iterative algorithm inductive modeling with application of network technologies: Ph.D. thesis. IRTC IT&S NASU, – Kiev, Ukraine. – 2011. – 171 p. (In Russian).
12. Yurachkovsky Yu.P. Restoring polynomial dependency on the basis of self-organization. *Avtomatika*. Kiev, USSR. – 1981. – № 4. – p. 15-20. (In Russian).

13. Kondo Tadashi, Ueno Junji. Feedback GMDH-Type Neural Network Self Selecting Optimum Neural Network Architecture and Its Application for 3-Dimensional Medical Image Recognition of the Lungs. *Proc. International Workshop on Inductive Modeling* (IWIM'2007). – Prague: CTU in Prague, 2007. – p. 63-70.

14. Kordik P. Fully automated knowledge extraction using group of adaptive model evolution: Ph.D. thesis. : *Electrical Engineering and Information Technology*. – Prague, 2006. – 150 p.

15. Bodyanskiy Ye., Pliss I., Vynokurova O. Hybrid GMDH-neural network of computational intelligence. In: *Proc. 3rd Int. Workshop on Inductive Modelling*. Poland, Krynica, 2009. – p. 33–41.

16. Ivakhnenko A.G. The group method of data handling is a competitor of method stochastic approximation. *Avtomatika*. Kiev, USSR. – 1968. – №3. – p. 58–72. (In Ukrainian).

17. Ivakhnenko N.A., Marchev A.A. Self-organization of the mathematical model for forward planning of construction and installation works. *Ibid*. – 1978. – № 3, – p. 12–18. (In Ukrainian).

18. Ivakhnenko A. G., Ivakhnenko G. A., Muller J.-A. Self-organization of Neural Networks with Active Neurons. *Pattern Recognition and Image Analysis*. – v.4. – N. 2. – Moscow, Russia. – 1994. – p. 185–196. (In Russian).

19. Lbov G.S. Methods of processing different types of experimental data. *Nauka*, Novosibirsk, USSR. – 1981. (In Russian).

20. Hinton G. E., Salakhutdinov R. R. Reducing the Dimensionality of Data with Neural Networks // Science 313 (2006), p. 504 – 507.

21. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature* 521, 436–444 (28 May 2015), doi: 10.1038/nature14539
http://www.nature.com/nature/journal/v521/n7553/full/nature14539.html

22. http://irtc.org.ua/image/app/webroot/Files/presentations/IPMMS/%D0%A0%D0%B5%D0%B7%D0%BD%D0%B8%D0%BA_Chernodub_Deep_v03_10_11_15.pdf

23. http://people.idsia.ch/~juergen/deep-learning-overview.html

24. http://people.idsia.ch/~juergen/deep-learning-conspiracy.html