Model's stratification of self-similar multi-layer neural network and fast transformations

Alexandr Dorogov[a]

[a] *Saint Petersburg State Electrotechnical University "LETI", St. Petersburg, 197376, Russian Federation*

Abstract

A stratified model is proposed for self-similar modular neural networks includes morphological, structural, topological and parametric levels. Without loss of functionality, simplification of modular neural network models is achieved using transitive connections. It is shown that the morphogenesis of the structural model is determined on the population of graded spaces of the terminal layers of the network. Structurally regular neural networks are considered. It is shown that fast transformation algorithms (including FFT) can be described by a topological model of a structurally regular self-similar network. A linguistic model for describing the topologies of regular self-similar networks is presented. An algorithm for constructing topological models of fast algorithms is proposed. The sufficiency of the topological model for describing the complete set of fast algorithms is shown. Examples are given.

Keywords 1

Stratification, neural network, modularity, morphogenesis, self-similar graph, FFT, structural model, topological model, parametric model, transitive connections

1. Stratification of neural networks model representations

Mathematical models serve as a tool for studying and designing neural networks. The model is required to be simple, but functionally sufficient. The main problem of mathematical modeling is finding an acceptable compromise between detail and simplicity of description. One way to solve this problem is to form hierarchically nested model families, where each level of the hierarchy corresponds to a level of reasonable abstraction of system properties, which leads to simplification of each particular model. This multi-level model representation is called as stratification [1], and each level of the model representation is called as strata. Morphological level models were constructed for self-similar networks in [2].

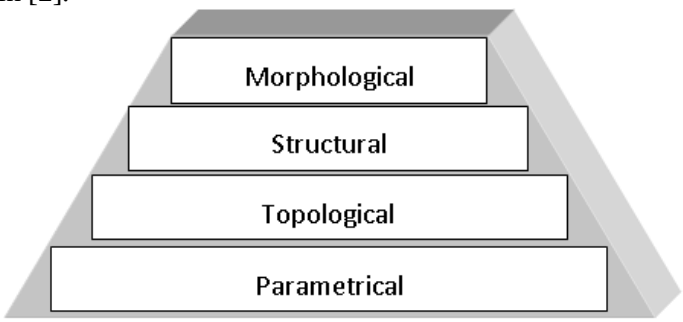

Figure 1: Levels of stratified model

Russian Advances in Fuzzy Systems and Soft Computing: selected contributions to the 8-th International Conference on Fuzzy Systems, Soft Computing and Intelligent Technologies (FSSCIT-2020), June 29 – July 1, 2020, Smolensk, Russia
EMAIL: vaksa2006@yandex.ru
ORCID: 0000-0002-7596-6761

In this article, models of the structural, topological, and parametric levels will be considered. Strata of model representations are ordered by degree of abstraction (see Figure 1). The highest level of abstraction corresponds to a morphological representation, and the lowest level corresponds to a parametric representation. In the mathematical formulation, stratification is associated with the extraction of equivalent relations at each level of the hierarchy and the transition to factor models that describe the next level.

2. Structural model of a modular neural network

We introduce structural characteristics for modules and intermodule connections of multi-layer neural network. In further the neural module of the layer m with the number z^m is denoted $A_{z^m}^m$. For the module $A_{z^m}^m$, we denote by p_{z^m} dimensionality of its receptor field, and by g_{z^m} dimensionality of its axon field. The module performs data processing and is generally described by a non-linear operator. The operator is characterized by an operator rank. When the module's data processing capabilities are fully used (i.e. the module does not have "hanging" receptors or axons that are not connected to other modules), its maximum rank is determined by the expression.

$$rank\left(A_{z^m}^m\right) = \min(p_{z^m}, g_{z^m}).$$

We will call such module as *complete module*. We assume that for the neural network, the connection between neural modules A_z^m and A_z^{m+1} of adjacent layers is a linear operator that connects the axon and receptor fields of two neighboring modules. It is advisable to simplify the neural network model, assuming that all processing is concentrated in neural modules, and connections only transmit data without internal processing and distortion. This leads to the need to reject branching connections in the neural network and assume that all intermodule connections are injective and identity. The latter simplification does not reduce the network's data processing capabilities, since data branching and scaling can be organized inside neural modules. Under the assumptions made all matrices of the connection operators are unit:

$$P_{zq} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

and the operator rank in this case is equal to the dimensionality of the unit matrix. Injective non-distorting connections will be further called as *transitive*. An example of a structural model of a modular self-similar neural network with one rank transitive connections is shown in Figure 2. The modules in the input layer have the dimensionalities $(3,2)$ and other layers, $(2,2)$. The structural model differs from the morphological one by the presence of weights of vertices and arcs on the model graph.

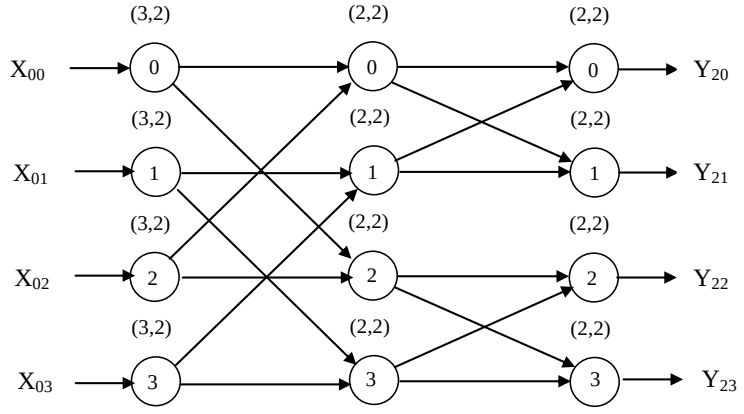

Figure 2: Structural model of a self-similar modular neural network with one rank transitive connections

3. Morphogenesis of the structural level

The morphogenesis conditions [2] of a self-similar multilayer network at the morphological level were determined through vertex projections on the terminal fields of the network. According to the proven theorem "on the morphology of weakly connected networks", the graph of a self-similar network can be expressed by a linguistic sentence in which each word represents a bit-by-bit representation of the layer vertex number z^m in an alphabet with two types of indexes i and j.

$$z^m = \langle j_0 j_1 \dots j_{m-1} j_{m+1} j_{m+2} \dots j_{n-1} \rangle.$$

We will show how these morphogenesis conditions are transformed to the structural level. We associate the linear vector spaces $P(z^m)$ and $G(z^m)$ with each module$A_{z^m}^m$, which we call associative spaces to the input and output of the module. The dimensionalities of the associative spaces are defined by the dimensionalities of the terminal fields of the neural module so that $p_{z^m} = \dim\big(P(z^m)\big), g_{z^m} = \dim\big(G(z^m)\big)$. Since the intermodule connections for a self-similar network are injective [2], the same-named associative spaces of the layer do not intersect with each other. Therefore, the vector space equal to the direct sum of the input associative vector spaces of the zero layer will be associated with the afferent of the network:

$$P(Aff) = \bigoplus_{z^0} P(z^0).$$

and the vector space equal to the direct sum of the output associative vector spaces of the final layer of the network will be associated with an efferent of the network:

$$G(Eff) = \bigoplus_{z^{n-1}} G(z^{n-1}).$$

Recall [2] that afferent of a multilayer neural network is a set of vertices of the input (zero) layer network, and efferent is the set of vertices of the last layer. The input and output layers of the network are called terminal layers. Assuming that the dimensionality of the network in the input is equal to*N*, and in the output, is equal to *M* we can write:

$$N = \sum_{z^0} p_{z^0}, M = \sum_{z^{n-1}} p_{z^{n-1}}.$$

The module associative spaces from the layer m are projected onto the terminal spaces of the network via chains of transitive connections. Modules of the layer that have coinciding projections form *domains*. A domain's own space is the direct sum of the associative vector spaces of its representatives. Projections of own domain spaces in terminal layers do not intersect and are equal to the direct sum of associative input or output module spaces of terminal layers.

If a vector space has a fixed expansion into a direct sum of subspaces, then it is said that the vector space is graded [3]. Thus, morphogenesis of the morphology level induces morphogenesis on a population of graded subspaces of terminal layers. For a structural model, you must set the ranks of intermodule relationships. Since the connections are transitive, the following conditions are met for each module:

$$p_{z^m} = \sum_{z^{m-1}} r(z^{m-1}, z^m), g_{z^m} = \sum_{z^{m+1}} r(z^m, z^{m+1}),$$

where $r(z^{m-1}, z^m)$ and $r(z^m, z^{m+1})$ are ranks of connections between adjacent modules. We call a network structurally regular if all vertices within a layer have the same dimensionalities and the same operator ranks for input and output connections. If the structural regularity conditions are met, the dimensionalities of modules for a layer network can be set as following tables:

$$\begin{pmatrix} 0 & 1 & \dots & n-1 \\ p_0 & p_1 & \dots & p_{n-1} \end{pmatrix}, \begin{pmatrix} 0 & 1 & \dots & n-1 \\ g_0 & g_1 & \dots & g_{n-1} \end{pmatrix}.$$

The left table defines the dimensionalities of the receptor fields of the neural modules by layers, and the right table defines the dimensionalities, of axon fields. For structurally regular networks, the index tuple can be considered as a positional representation of a number in a mixed-radix number

system with radixes $\{p_m, g_m\}$, for example, assuming that the left digit of the tuple of the vertex number is the highest one can write:

$$z^m = \langle j_0 j_1 \dots j_{m-1} i_{m+1} i_{m+2} \dots i_{n-1} \rangle = j_0 g_1 g_2 \dots g_{m-1} p_{m+1} p_{m+2} \dots p_{n-1} + \cdots$$
$$+ j_1 g_2 g_3 \dots g_{m-1} p_{m+1} p_{m+2} \dots p_{n-1} + \cdots + i_{n-2} p_{n-1} + i_{n-1}.$$

Figure 2 shows a self-similar neural network in which all connections have unity rank and all modules have equal dimensionalities. As already noted in [2], this is a fairly narrow class of multilayer self-similar networks. It is advisable to expand it to include structured regular networks. The basis for this is the auto-simulation of the linguistic description of graphs of this class of networks to the graph of trivial self-similar network, so we will also call the structurally regular network as self-similar. A great advantage of regular self-similar networks is the ability to obtain analytical expressions for algorithms for structural synthesis and training of neural networks. Structural regularity is correlated with the concept of graph regularity. For one rank self-similar networks, both concepts are the same. The structural synthesis of irregular self-similar networks is considered by the author in the paper [4]

4. Topological model of a modular neural network

Inputs and outputs of the structural model in Figure 2 are shown as three-and two-coordinate vectors without reference to the coordinate numbers. To describe the algorithm, you must enter a topological model. In the topological model, the elements of consideration are the physical contacts of the neural modules; either input receptors or output axons of a module.

Let's consider a modular network where all modules are complete and all connections are transitive. Let's denote $u_{z^m} = [0,1, \dots, (p_{z^m} - 1)]$ the local number of the receptor for the module $z^m$of the layer m, and $v_{z^m} = [0,1, \dots, (g_{z^m} - 1)]$, the local number of the axon of the module. The positional number of a receptor within the neural layer is denoted by U^m, and the positional number of a axon is denoted by V^m. Set of mappings:

$$\{u_{z^m}\} = \bigcup_{z^m} u_{z^m} \to U^m, \{v_{z^m}\} = \bigcup_{z^m} v_{z^m} \to V^m$$

are projections, since all modules in the model under consideration are complete. Therefore, for the networks of this type, the union mappings are one-to-one.

Let's now consider a structurally regular self-similar network (in which all modules within each layer have the same structural characteristics $[p_m, g_m]$). In this case, you can simplify the notation for the receptor number and axon number of the neural module by writing:

$$u_m = [0, 1, \dots, (p_m - 1)], \quad v_m = [0, 1, \dots, (g_m - 1)].$$

We will also assume that all connections in the network are one rank. Then the topological mappings of the layer can be expressed as tuples:

$$U^m = \langle \langle z^m \rangle \oplus u_m \rangle, \quad V^m = \langle \langle z^m \rangle \oplus v_m \rangle.$$

In these expressions the symbol $\oplus$ emphasizes that the placement of additional digits $u_m$and v_m in the tuple $z^m = \langle j_0 j_1 \dots j_{m-1} j_{m+1} j_{m+2} \dots j_{n-1} \rangle$ can be arbitrary. For unity rank connections, each arc in the structural model graph one-to-one corresponds to the arc of the topological model, for example, you can choose the following one-to-one concordances: $i_m \leftrightarrow u_m, j_m \leftrightarrow v_m$, then there is the following variant of topological mappings:

$$\begin{aligned} U^m &= \langle v_0 v_1 \dots v_{m-1} u_m u_{m+1} \dots u_{n-1} \rangle, \\ V^m &= \langle v_0 v_1 \dots v_{m-1} v_m u_{m+1} \dots u_{n-1} \rangle, \\ z^m &= \langle v_0 v_1 \dots v_{m-1} u_{m+1} \dots u_{n-1} \rangle \end{aligned} \qquad (1)$$

The graph of the topological model is constructed according to the same rules [2] as the graph of the morphological model i.e. arcs connect vertices that have the same values of bit digits in adjacent layers. The graph of the topological model for this example is shown in Figure 3.

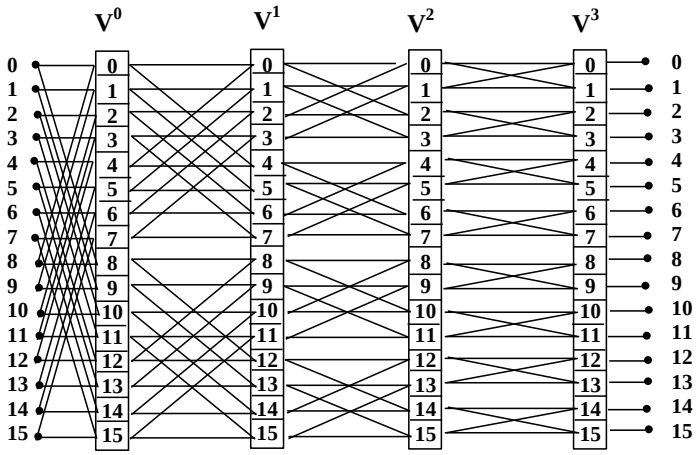

Figure 3: Topological model of the decimation-in-frequency FFT

It is not difficult to make sure that this model corresponds to the fast Fourier transform (FFT) graph in the Cooley-Tukey topology with "decimation-in-frequency" [5]. Another variant of topological mappings can be set as:

$$U^m = \langle u_{n-1}u_{n-2} \dots u_{m+1}u_m v_{m-1}v_{m-2} \dots v_0 v_1 \rangle,$$
$$V^m = \langle u_{n-1}u_{n-2} \dots u_{m+1}v_m v_{m-1}v_{m-2} \dots v_0 v_1 \rangle, \quad (2)$$
$$z^m = \langle u_{n-1}u_{n-2} \dots u_{m+1}v_{m-1}v_{m-2} \dots v_0 v_1 \rangle$$

This model corresponds to the FFT graph in the "decimation-in-time" Cooley-Tukey topology. The graph of this topological model is shown in Figure 4.

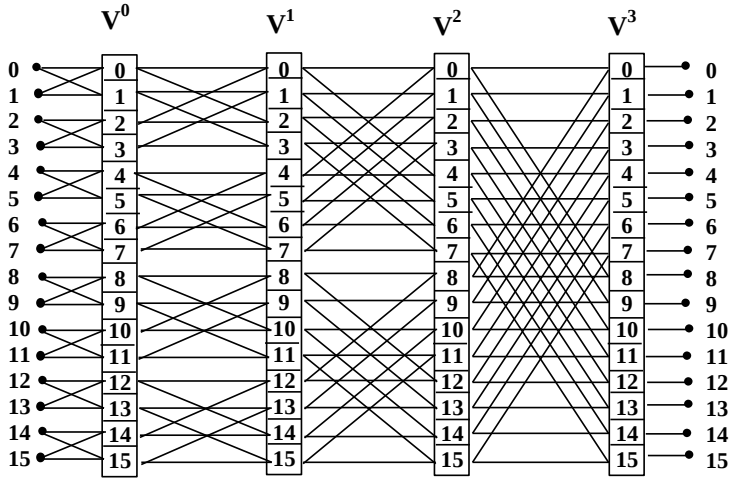

Figure 4. Graph of the topological model of the FFT algorithm with " decimation-in-time"

From the obtained results, we can conclude that the fast Fourier transform algorithm is a topological implementation of a self-similar modular network, where the modules are basic operations of the *'butterfly'* type. For fast neural networks, the term *'neural kernel'* is used instead of the basic operation. From the topological model (1) for terminal layers, we get:

$$U^0 = \langle u_{n-1}u_{n-2} \dots u_1 u_0 \rangle, \;\; V^{n-1} = \langle v_{n-1}v_{n-2} \dots v_1 v_0 \rangle.$$

If *N* is the dimension of the receptor field, and *M* is the dimension of the axon field of the network, then from the latter expressions directly follow:

$$N = p_{n-1} \dots p_1 p_0, \quad M = g_{n-1} \dots g_1 g_0.$$

Thus, the dimensionalities of the terminal fields of the network are determined by the product dimensionalities of neural modules by layers. The examples of self-similar networks discussed above were constructed for dimensionalities $p_i = g_i = 2$. Figure 5 shows the graph of the *'decimation-in-frequency'* topology for dimensionalities:

$$\begin{pmatrix} p_0 & p_1 & p_2 \\ 3 & 2 & 2 \end{pmatrix}, \begin{pmatrix} g_0 & g_1 & g_2 \\ 2 & 2 & 2 \end{pmatrix}.$$

Topological sentences for this example have the form:

$$\begin{aligned} Rp &= \{U^m\} = [\langle u_2u_1u_0\rangle\langle u_2u_1v_0\rangle\langle u_2v_1v_0\rangle], \\ Ax &= \{V^m\} = [\langle u_2u_1v_0\rangle\langle u_2v_1v_0\rangle\langle v_2v_1v_0\rangle], \\ \{z^m\} &= [\langle u_2u_1\rangle\langle u_2v_0\rangle\langle v_1v_0\rangle]. \end{aligned} \quad (3)$$

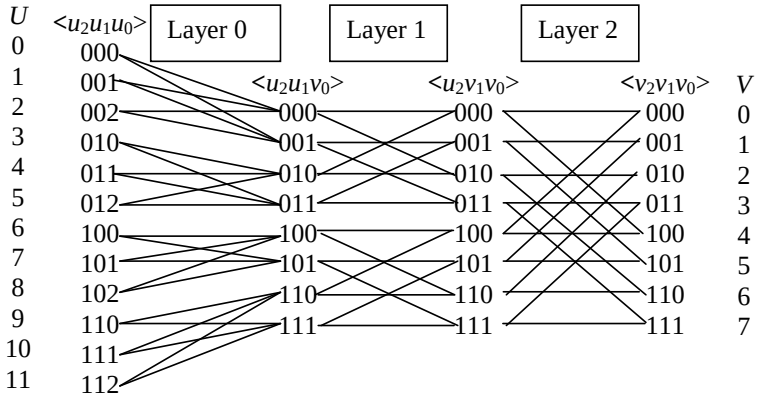

Figure 5: Topological graph of fast transformation of dimensionalities 12×8

This graph is an example of a topological model of a fast neural network [6]. The structural model for this network is shown in Figure 2. For this structural model, there are many other variants for constructing topological models, any one-to-one mappings $\{u_{z^m}\} \to U^m, \{v_{z^m}\} \to V^m$ are acceptable, but not all of them are represented as digit tuples. Topological models that can be described by digit tuples are naturally to consider as topologically regular.

5. Parametric models of fast transformations

In the graph of the fast algorithm, the modules are basic operations (neural kernels), represented as matrices of small dimension. For the fast transformation algorithm, the basic operation z^m in the layer m performs linear processing for the components of the input layer vector, following the rule:

$$y^m_{z^m}(v_m) = \sum_{u_m} x^m_{z^m}(u_m) w^m_{z^m}(u_m, v_m),$$

where $x^m_{z^m}$ and $y^m_{z^m}$ are coordinates of the input and output vectors of the base operation; $w^m_{z^m}$ is matrix of weights of the base operation. To build the transformation algorithm, you need to switch from local kernel variables to external bit variables of the layer. This transition is implemented based on a topological model. The parametric description of basic operations together with the topological model forms a parametric model of fast transformation. Figure 6 shows the graph of the topological model (3) with the selected basic operations.

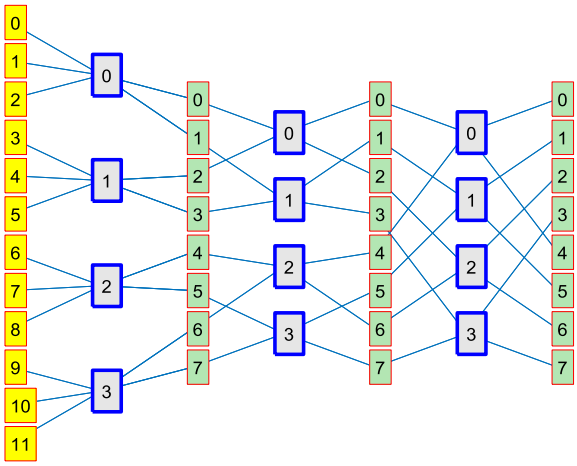

Figure 6: Graph of a topological model with selected basic operations

6. Conclusion

A set of algorithms called fast Fourier transform algorithms (FFT) has been used in spectral analysis since the work of Cooley-Tukey in the 60s of the last century. The theoretical basis of fast algorithms has long been based on various factorization theorems, which proved the possibility of decomposing the spectral transformation matrix into a product of weakly filled matrices [5,8]. At the time, this generated a surge of work on factorization theorems. However, along the way, the researchers encountered the fact that there are many different decompositions for the same spectral transformation. When the number of theorems of all possible factorization theorems exceeded dozens, it became clear that this path of development of the theoretical foundations of fast algorithms is a dead end.

The misunderstanding was purely methodological and consisted in mixing the concepts of structure and topology of the fast transformation algorithm. The structure is stable system invariant characteristic of the entire class of fast algorithms, and the topology is no more than a valid implementation of the system invariant in the relationships between the base operations. Each factorization theorem corresponds to one of the possible forms of topological implementation, and the number of them increases rapidly with the growth of the dimensionality of the transformation, so the flow of factorization theorems can be almost inexhaustible. Moreover, factorization theorems exist only for specific transformations with an analytically defined type of functions but there are no such theorems for tunable fast transformations and fast neural networks.

Fast neural networks are variant of multilayer neural networks that have a fast algorithm for processing input data. In order to develop a new understanding of the principles of building fast algorithms, a system analysis was required [9] which led to the ideology of stratified models of weakly connected networks. The FFT algorithm turned out to be a special case of a weakly connected network. Later, the author proved that for the condition of weak connectivity of a regular network and the condition of self-similarity of the morphological structure of the network are equivalent.

Strata of model representations are ordered by the degree of abstraction: the highest level corresponds to the morphological representation, and the lowest corresponds to the parametric one. Stratification allows us to explore the system at different knowledge stages and describe each level by adequate means. The structural model presents the dimensionalities of neural modules and operator ranks of intermodule connections, but there is no information about the data vector coordinate indexes. This model is intended for evaluating qualitative indicators of fast tunable transformation, such as performance and plasticity.

Binding of input and output vectors to the structural model might have a lot of variants and generates a set of topological implementations of fast transformation. The topological model allows us to choose the form of the algorithm that is convenient for practical implementation. The topological model, supplemented with the values of the coefficients of the basic operations, forms a

model of the parametric level. At the parametric level, methods for training and tuning neural networks based on specified quality indicators are implemented.

The main conclusion is that stratification of model representations allows us to perform a hierarchical decomposition of models of self-similar neural networks and fast algorithms into relatively independent levels and use a specific method of research and design for each level. The self-similar structure of a fast neural regular network allows us to offer training methods that converge absolutely in a finite number of steps [7]. In addition, there is a variant of analytical extension of the topology of a self-similar network that leads to architectures of deep neural networks with fast absolute convergence learning algorithms [10]

7. References

[1] V. N. Volkova, A. A. Denisov, Osnovy teorii sistem i sistemnogo analiza: Ucheb. dlja studentov vuzov. SPb.: Izd-vo SPbGTU, 1999.

[2] A. Yu. Dorogov Morphological model of self-similar multi-layer neural networks. In this conference Proceedings.

[3] A. I. Kostrikin, Ju. I. Manin, Linejnaja algebra i geometrija: Ucheb. posob. dlja vuzov.- 2-e izd. pererab.- M.: Nauka, 1986.

[4] A. Yu. Dorogov, Structural Synthesis of Modular Weakly Connected Neural Networks. II. Nuclear Neural Networks, Cybernetics and Systems Analysis July - August , 37 (4) (2001) 470-477.

[5] L. R. Rabiner, B. Gold, Theory and Application of Digital Signal Processing. – Englewood Cliffs, NJ: Prentice-Hall, 1975.

[6] A. Yu. Dorogov, Teorija i proektirovanie bystryh perestraivaemyh preobrazovanij i slabosvjazannyh nejronnyh setej. SPb.: “Politehnika”, 2014.

[7] A. Yu. Dorogov, Teoreticheskie osnovy obuchaemyh algoritmov bystryh preobrazovanij, Tezisy dokladov Nauchno-tehnicheskoj konferencii, 2011, 107-109.

[8] Je. E. Dagman, G. A. Kuharev, Bystrye diskretnye ortogonal'nye preobrazovanija, 1983.

[9] A. Ju. Dorogov, Sistemnye invarianty bystryh preobrazovanij. Trudy Chetvertoj mezhregional'noj shkoly seminara, 14 -16 sentjabrja 2004 goda, g. Krasnojarsk. (in Russian).

[10] A. Yu. Dorogov, Fast Deep Learning Neural Networks, Proceedings of 3rd International Conference on Control in Technical Systems, CTS 2019, conference-paper. doi: 10.1109/CTS48763.2019.8973297.