ISSN 1064-5624, Doklady Mathematics, 2019, Vol. 100, No. 2, pp. 411–415. © Pleiades Publishing, Ltd., 2019.
Russian Text © The Author(s), 2019, published in Doklady Akademii Nauk, 2019, Vol. 488, No. 1.

MATHEMATICS

Neural Network Construction for Recognition Problems with Standard Information on the Basis of a Model of Algorithms with Piecewise Linear Surfaces and Parameters

Academician **Yu. I. Zhuravlev**[a,*] and **A. E. Dyusembaev**[b,**]

Received May 21, 2019

Abstract—For recognition problems with standard information, a neural network reproducing computations performed by a correct algorithm is constructed on the basis of the operator approach and a model of algorithms with parameters and piecewise linear surfaces.

DOI: 10.1134/S1064562419050041

In this paper, the operator approach [1–3] is used as in [4–6] to improve the accuracy of solutions produced by some class of neural networks. Let $C'_1, C'_2, \ldots, C'_l$ be classes that entirely cover the space of initial objects $X = \{x | x = (x_1, x_2, \ldots, x_n),\ x \in R^n\}$, where x_i is a feature of the object x and $i = 1, 2, \ldots, n$. Let $Q_1, Q_2, \ldots, Q_l$ be a system of unary two-valued predicates over X such that $Q_j(x) \equiv \langle\langle x \in C'_j \rangle\rangle$, $x \in X, j = 1, 2, \ldots, l$. The recognition problem $u \in U$ is an ordered pair $u = (I_0, X^q)$, where $I_0 = \langle X^m, \omega\rangle$ is initial information for the problem u, $X^m = \{x^1, x^2, \ldots, x^m\}$ is a training sample, and $\omega = \|\omega_{ij}\|_{m\times l}$ is the classification matrix of the sample X^m ($\omega_{ij} = Q_j(x^i)$, $i = 1, 2, \ldots, m; j = 1, 2, \ldots, l.$). The sample $X^q = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^q\}$ is a sample of test objects, and the classification matrix $f = \|f_{ij}\|_{q\times l}$ of the sample X^q (as in the problem u) has to be computed. Here, $f_{ij} = Q_j(\boldsymbol{x}^i)$; $i = 1, 2, \ldots, q; j = 1, 2, \ldots, l$. A model $\mathfrak{M}(H, \tilde{\mathrm{x}}, \gamma^m, \theta_1, \theta_2)$ of recognition algorithms was proposed in [1, 2]. Here, H is a piecewise linear surface in R^n; $\tilde{\mathrm{x}} = (\mathrm{x}_{00}, \mathrm{x}_{11}, \mathrm{x}_{10}, \mathrm{x}_{01})$ is a binary set of parameters; $\tilde{\gamma}^m$ is a set of parameters of the weight type; and $\theta_1 = \min \theta_{1j}$ and $\theta_2 = \max \theta_{2j}$ are parameters of the decision rule R^* such that $0 < \theta_{1j} \le \theta_{2j}, j = 1, 2, \ldots, l$. Correctness conditions for linear and algebraic closures of algorithms of the model $\mathfrak{M}(H, \tilde{\mathrm{x}}, \tilde{\gamma}^m, \theta_1, \theta_2)$ over sets of problems with standard information were found in [2]. For regular problems, an analytical form of a correct algorithm was found in [3], namely,

$$\mathcal{A}^* = \left((\theta_1 + \theta_2)\sum_{i=1}^{q}\sum_{j=1}^{l} f_{ij} \cdot B^k(i, j)\right) \circ R^*, \tag{1}$$

$$k = [(\ln q + \ln l + \ln(\theta_1 + \theta_2) - \ln\theta_1)/|\ln a_0|] + 1, \tag{2}$$

where $a_0 = \max\limits_{i,j} \max\limits_{(r,h)\ne(i,j)} |\Gamma_{rh}(i, j)|$. Here, $\|\Gamma_{rh}(i, j)\|_{q\times l}$ is the matrix of the quasi-basis operator $B(i, j)$ [3], $i = 1, 2, \ldots, q; j = 1, 2, \ldots, l$. As an initial model, the model of estimate calculation algorithms was used [3]. In what follows, let each recognition algorithm $\mathcal{A}$ be such that $\mathcal{A}\colon \mathcal{A} = A \circ R^*$ [3], where A is a recognition operator and R^* is a threshold decision rule. Given the regular problem u [3], an operator A computes the matrix $\varphi = \|\varphi_{ij}\|_{q\times l}$, where φ_{ij} is an estimate determining the membership of the object $\boldsymbol{x}^i \in X^q$ in the class C_j. In (1), given φ and k computed according to (2), the decision rule $R^*(\theta_1, \theta_2)$ yields a matrix δ coinciding with the matrix f of the problem u. In a more general case, given φ, the operator R^* computes a matrix that may not coincide with the matrix $f = \|f_{ij}\|_{q\times l}$. Let U be the class of problems with standard information [2]. Our goal is, relying on [1–3] and the neural network paradigm and using algorithms of the model $\mathfrak{M}(H, \tilde{\mathrm{x}}, \gamma^m, \theta_1, \theta_2)$, to show the possibility of con-

[a] *Dorodnicyn Computing Center, Federal Research Center "Computer Science and Control", Russian Academy of Sciences, Moscow, 119333 Russia*
[b] *Al-Farabi Kazakh National University, Almaty, 050040 Kazakhstan*
**e-mail: zhur@ccas.ru*
***e-mail: anuardu@yahoo.com*

structing a neural network which, given an arbitrary problem $u \in U$, outputs its matrix f. There are a variety of families of recognition algorithms based on the partition principle [2, 7, 8], and the research of them both earlier and today forms the content of the classical field.

In what follows, let $A', A'', A \in \mathfrak{M}$ be operators. The algebra ϑ over $\mathfrak{M}$ is constructed using the following operations [2, 3]: (a) const Au, (b) $(A' + A'')u = A'u + A''u$, and (c) $(A' \cdot A'')u = A'u \cdot A''u$, where, by (c), we mean the power operation A^k. Here, neither the operands of the operations nor the resulting matrices in (a–c) must have elements with large moduli. Suppose that these values are specified on the interval $(-1, 1]$. Let $C_j = X^m \cap C_j', j = 1, 2, \ldots, l$.

Furthermore, relying on the model of algorithms $\mathfrak{M}(H, \tilde{x}, \gamma^m, \theta_1, \theta_2)$ [2] and the neural network paradigm, we also take into account the weights $p_1, p_2, \ldots, p_n$ of features of objects from R^n. This model is denoted by $\widetilde{\mathfrak{M}}(H, \tilde{p}^n, \tilde{x}, \gamma^m, \theta_1, \theta_2)$, where $\tilde{p}^n = (p_1, p_2, \ldots, p_n), p_i \geq 0$. Let α, β be characteristics of an object x associated with its membership in a fixed class and its position relative to H. The two-valued predicate $H(x)$ defined on X determines which of the half-spaces relative to H contains the object x. We deal with the values $\alpha = Q_j(x)$ and $\beta = H(x)$. For $\beta : \beta = 1$ if the object x belongs to the positive half-space determined by H in R^n and $\beta = 0$ otherwise [2]. The values of β for objects from X^m are denoted by β_t, $t = 1, 2, \ldots, m$, while β^i is used for objects from X^q. The values β^i and $\overline{\beta}^i$ will be used later as synoptic weights of fourth-layer neurons in the network j-block (see Fig. 1). The training objects given as input to a neural network are arranged in the form of the list #, $x^m(\alpha_t, \beta_t), \ldots, x^t(\alpha_t, \beta_t), \ldots, x^2(\alpha_2, \beta_2), x^1(\alpha_1, \beta_1)$, which is finished with a special object—the value #. The notation $x^t(\alpha_t, \beta_t)$ is formal and means only that, given C_j and H, the object x^t is assigned with α_t, β_t. Moreover, neither object x^t nor its weight γ depends on α_t, β_t. For objects from X^q, the characteristics are $(\beta, \tilde{x})$. The following question arises: is it possible to construct a neural network that reproduces the computations executed by a correct algorithm on the basis of the model $\widetilde{\mathfrak{M}}(H, \tilde{p}^n, \tilde{x}, \gamma^m, \theta_1, \theta_2)$?

Assuming that the network is multilayered, initially the activation function of first-layer neurons of the network is defined as

$$F^j_{\mu\eta}(x^t(\alpha_t, \beta_t), p) = \begin{cases} \dfrac{(x^t(\alpha_t, \beta_t), p)}{x_1^t(\alpha_t, \beta_t) + x_2^t(\alpha_t, \beta_t) + \ldots + x_n^t(\alpha_t, \beta_t)}, & \text{if} \quad x^t(\alpha_t, \beta_t) \neq \overline{0} \quad \text{and} \quad \alpha_t = \mu, \quad \beta_t = \eta \\ p_0, \quad \text{if} \quad x^t(\alpha_t, \beta_t) = \overline{0} \quad \text{and} \quad \alpha_t = \mu, \quad \beta_t = \eta & \\ 0, \quad \text{if} \quad \alpha_t \neq \mu \quad \text{or} \quad \beta_t \neq \eta, & \end{cases}$$

where $\mu, \eta \in \{0, 1\}$, $p = (p_1, p_2, \ldots, p_n)$ are synoptic weights of first-layer neurons of the network, and p_0 is a parameter. For $\alpha_\#, \beta_\#$ of the object #, we assume that $\alpha_\# = 0$, $\alpha_\# = 1$ and $\beta_\# = 0$, $\beta_\# = 1$ hold simultaneously and, additionally, $0 \cdot \# = \# \cdot 0 = 0$. The weight is $\gamma(\#) = 0$. Ignoring the computations on the adder, we set $F_{\mu\eta}(\#(\alpha_\#, \beta_\#), p) = \#$. For brevity, along with the notation $x^t(\alpha_t, \beta_t)$, we also use the notation $x(\alpha, \beta)$ or, as in Fig. 1, the notation $x^t(\alpha, \beta)$. Define

$$D^j_{\mu\eta} = \sum_{x: \alpha = \mu, \beta = \eta, x \in X^m} \gamma(x(\alpha, \beta)) \cdot F^j_{\mu\eta}(x(\alpha, \beta), p),$$

where $\gamma(x(\alpha, \beta))$ is the weight of the training object $x(\alpha, \beta)$. Relying on the method for computing the estimate Γ_{ij} by an operator of the model $\mathfrak{M}$ and taking into account the neural network paradigm, we assume that, for the given problem u, an operator A of the model $\widetilde{\mathfrak{M}}$ calculates an estimate determining the membership of the object $\boldsymbol{x}^i$ in the class C_j, i.e., the element Γ_{ij} of the matrix $\|\Gamma_{ij}\|_{q \times l}$ defined as

$$\Gamma_{ij} = \frac{\mathrm{x}_{00}(\boldsymbol{x}^i) \cdot D^j_{00} + \mathrm{x}_{11}(\boldsymbol{x}^i) \cdot D^j_{11}}{\mathrm{x}_{10}(\boldsymbol{x}^i) \cdot D^j_{10} + \mathrm{x}_{01}(\boldsymbol{x}^i) \cdot D^j_{01} + 1} \quad \text{if} \quad \mathrm{H}(\boldsymbol{x}^i) > 0, \tag{3}$$

$$\Gamma_{ij} = \frac{\mathrm{x}_{10}(\boldsymbol{x}^i) \cdot D^j_{10} + \mathrm{x}_{01}(\boldsymbol{x}^i) \cdot D^j_{01}}{\mathrm{x}_{00}(\boldsymbol{x}^i) \cdot D^j_{00} + \mathrm{x}_{11}(\boldsymbol{x}^i) \cdot D^j_{11} + 1} \quad \text{if} \quad \mathrm{H}(\boldsymbol{x}^i) < 0, \tag{4}$$

where Γ_{ij} is an intermediate estimate to be processed at internal network layers. The main block of the network is represented in the form of a four-layer plane neural network with the nonstandard part consisting of the second to fourth layers (Fig. 1); this part of the network is called a j-block.

Inspecting the first-layer neurons (Fig. 1), we see that output in each of them corresponds to the jth class and is directed toward the input of a second-layer neuron of the j-block; this input has the synoptic weight $\gamma(x^t(\alpha_t, \beta_t))$. Output of the other first-layer neuron with the same value computed by the adder is directed toward its own class; moreover, for each class, the object x^t has its own characteristic α_t. A second-layer neuron accumulates the estimate $D^j_{\mu\eta}$. Partially defined on $X^m \cup \{\#\}$, the activation function of second-layer neurons is $f^j_{\mu\eta}(x^t, \tilde{x}) = x_{\mu\eta} \cdot D^j_{\mu\eta}$ if $x^t = \#$, and $f^j_{\mu\eta}(x^t, \tilde{x}) = \Delta$ (not defined) means only that x^t is updated if $x^t \neq \#$, i.e., the value $f^j_{\mu\eta}(x^t, \tilde{x})$ is not defined if the inner loop is not completed ($x^t \neq \#$). For each $\boldsymbol{x} \in X^q$, the objects x^t from X^m are sequentially given to the input of the network. The inner loop for the given $\boldsymbol{x}^i \in X^q$ is completed at $x^t = \#$ with computing the ith row of the matrix $\delta = \|\delta_{ij}\|_{q\times l}$ at the output of the neural network. The outer loop is completed with the exhaustion of X^q. Each neuron in the second-layer stratification has one more input from its own output in order to accumulate the estimate $D^j_{\mu\eta}$, and its synoptic weight is $d_0 = 1$ as long as $x^t \neq \#$. For $x^t = \#$, the synoptic weight d_0 becomes "0" and recovers the value of "1" when a new $\boldsymbol{x}^i \in X^q$ is given to the input of the network. The event $x^t = \#$ (exhaustion of X^m) takes place for all $j = 1, 2, \ldots, l$ simultaneously, which leads to a change of the object from X^q. Note that each neuron in the network preserves the adder-computed value, say s, only for a short time until s is transmitted as an output value to the neuron's activation function or until s is transmitted as an intermediate value, as in the case of a second-layer neuron, according to the feedback principle, sending s to its own input for the next iteration with updated x^t. The activation function R of third-layer neurons of the j-block is a diagonal one such that the value at the adder is the output of R. The fourth layer of the j-block consists of a nonstandard neuron with two synchronously functioning adders: Σ^1, Σ^2 and with the two-argument activation function $F_c(a,b) = c\dfrac{a}{b+1}$ with a parameter c. Depending on the value of the synoptic weight β^i (see Fig. 1), the value of this activation function is one computed by formula (3) or (4) multiplied by c. Let U^0 be the class of problems with standard information that do not contain isolated classes [2], and let $f = \|f_{ij}\|_{q\times l}$ be the classification matrix of the problem u. Relying on what was described above and the model $\tilde{\mathfrak{M}}(H, \tilde{x}, \tilde{p}^n, \gamma^m, \theta_1, \theta_2)$, in view of Theorems 3 and 4 from [2], we formulate the following result.

Theorem 1. *Let $u = (I_0, X^q)$ be an arbitrary problem from U^0. Then, for the problem u, a six-layer plane neural network with l blocks (j-blocks) can be constructed such that, given the classification matrix $f = \|f_{ij}\|_{q\times l}$ of the problem u and the parameter k, the matrix $\delta = \|\delta_{ij}\|_{q\times l}$ output by the network coincides with the matrix f of the problem u.*

To expand the class of problems for which a neural network can be constructed that outputs the classification matrix of a problem, the neural network is supplemented with an additional intermediate layer consisting of a nonstandard neuron with an activation function depending on two variables and a parameter c. This function is denoted by $F^0_c(a', b')$. Note that $F^0_c(a', b') = ca'b'$. Recall that a neuron of this kind was used previously as a fourth-layer neuron in the j-block.

The shaded rectangular areas in Fig. 2 correspond to the first layer of neurons of the network shown in Fig. 1. The constructed network fragment, which consists of two j-blocks j', j'', is supplemented with a fifth-layer neuron (Fig. 2). By analogy with the above-mentioned structure, the resulting one is called a J-block and the network itself is called a J-network (Fig. 3.). Assume that $J = 1, 2, \ldots, l$ and each J-block computes its own estimate. In fact, for each $r = 1, 2, \ldots, q$, the rows of the estimate matrix $\|\Gamma_{rh}(i,j)\|_{q\times l}$ of the quasi-basis operator $B(i,j)$ are sequentially calculated in the fifth layer. Eventually, we obtain a seven-layer plane neural network (Fig. 3), where the fifth layer contains l blocks (J-blocks).

Figure 3 shows an adder of the fifth layer as Σ^J_5, which is already known to represent the pair $\sum^J_5 = \left(\sum^1_5, \sum^2_5\right)$ (see Fig. 2). The sixth layer of the network has to implement the power operation for each quasi-basis operator: $B^k(i,j)$, $1 \leq i \leq q; 1 \leq j \leq l$, for which $f_{ij} \neq 0$. The neuron of this operation has two inputs, and the synoptic weight of one of them is d_1. Initially, $d_1 = 1$ to send the previously computed estimate Γ_{rh} to the adder. The synoptic weight d_2 of the second input is equal to 0. Next, the value of d_1 is replaced by 0 and Γ_{rh} becomes the synoptic weight of the second input of this neuron. This input receives the feedback from the neuron output and, by means of the activation function, given k and a_0 (see (2)) the neuron computes an element of the matrix of the operator $B^k(i,j)$ [9]. The output of this neuron is "strengthened" be the value $\theta = \theta_1 + \theta_2$ for pairs $f_{i_v, j_v} = 1$ such that (Fig. 3). Here, the activation func-

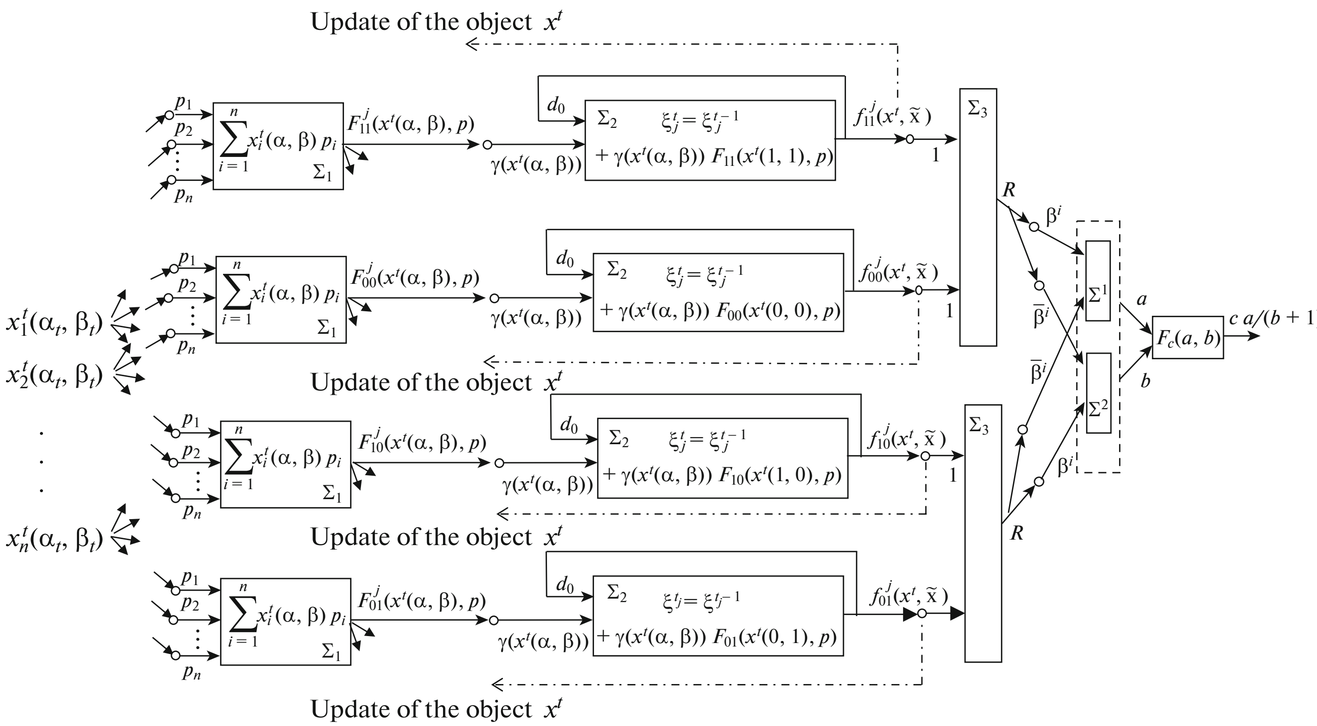

Fig. 1.

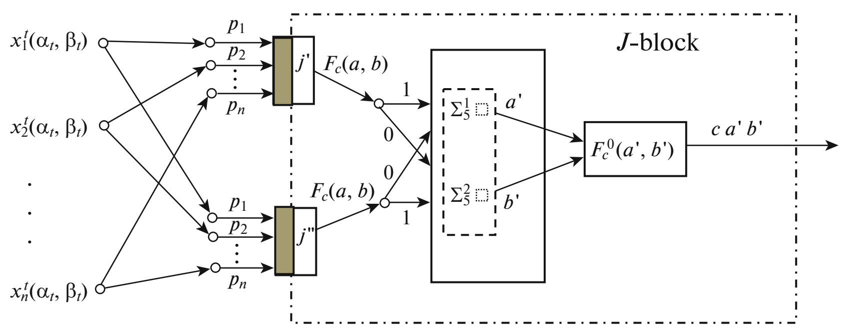

Fig. 2. J-block.

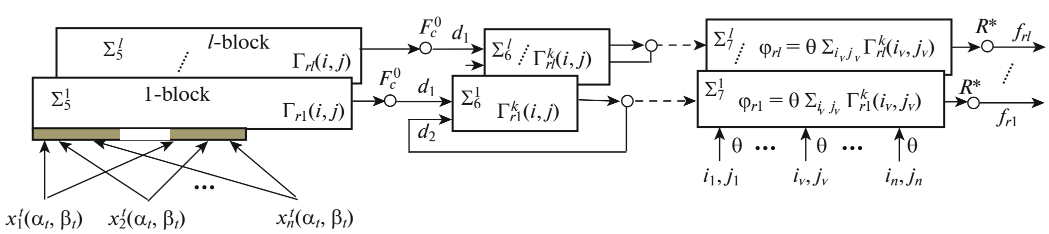

Fig. 3. Seven-layer J-network.

tion is a partially defined diagonal activation function. The adders of seventh-layer neurons sequentially compute the rows of the matrix $\varphi = \|\varphi_{ij}\|_{q \times l}$, which are used by the activation function to compute the corresponding rows of the matrix $\delta = \|\delta_{ij}\|_{q \times l}$. Let U be the class of problems [2] that satisfy the following conditions: (c) $X^q \cap X^m = \varnothing$ and (d) $C_{j'} \neq C_{j''}$ for $j' \neq j'': 1 \leq j', j'' \leq l$.

Theorem 2. *Let* $u = (I_0, X^q)$ *be an arbitrary problem from U. Then, for the problem u, a seven-layer plane neural network with l blocks (J-blocks) can be constructed such that, given the classification matrix* $f = \|f_{ij}\|_{q\times l}$ *of the problem u and the parameter k, the matrix* $\delta = \|\delta_{ij}\|_{q\times l}$ *output by the network coincides with the matrix f of the problem u.*

Note that the construction of j-blocks of the network does not require the construction of a piecewise linear surface H; instead, it is sufficient to specify the characteristics of the objects from $X^q \cup X^m$ for all cases from [2]. The parameters $\gamma_1, \gamma_2, \ldots, \gamma_m$ and $\tilde{x} = (x_{00}, x_{11}, x_{10}, x_{01})$ are determined according to the cases of [2], the parameter c is specified according to [2, 3]; the feature weights are $p_1 = p_2 = \ldots = p_n = p_0 = 1$; and the parameters θ_1, θ_2 are such that $0 < \theta_1 \leq \theta_2 < 1$ and $\theta_1 \leq (1 - \theta_2)/2$. Overall, all these parameters ensure the construction of a neural J-network for the problem $u \in U$.

REFERENCES

1. Yu. I. Zhuravlev, Dokl. Akad. Nauk SSSR **231** (3), 532–535 (1976).
2. Yu. I. Zhuravlev, Kibernetika, No. 4, 5–17 (1977).
3. Yu. I. Zhuravlev, Kibernetika, No. 2, 35–43 (1978).
4. A. E. Dyusembaev and D. R. Kaliazhdarov, Dokl. Math. **91** (2), 236–239 (2015).
5. A. E. Dyusembaev, Dokl. Math. **95** (2), 125–128 (2017).
6. A. E. Dyusembaev and M. V. Grishko, J. Pattern Recogn. Image Anal. **27** (2), 166–174 (2017).
7. K. V. Rudakov, Dokl. Akad. Nauk SSSR **231** (6), 1296–1299 (1976).
8. V. N. Vapnik, *Statistical Learning Theory* (Wiley, New York, 1998).
9. A. E. Dyusembaev and M. V. Grishko, Comput. Math. Math. Phys. **58** (10), 1673–1686 (2018).