Creating distributed artificial neural networks based on orthogonal transformations

Nikolay Vershkov
North-Caucasus Federal University
Stavropol, Russia
https://orcid.org/0000-0001-5756-7612

Mikhail Babenko
North-Caucasus Federal University
Stavropol, Russia
https://orcid.org/0000-0001-7066-0061

Vladislav Lutsenko
North-Caucasus Federal University
Stavropol, Russia
https://orcid.org/0000-0003-4648-8286

Nataliya Kuchukova
North-Caucasus Federal University
Stavropol, Russia
https://orcid.org/0000-0002-8070-0829

***Abstract*—The article addresses the issue of separating input information of artificial neural networks into modules using orthogonal transformations. This separation enables modular organization of neural networks with layer separation, facilitating the use of the proposed approach for distributed computing. Such an approach is required for organizing the operation of neural networks in fog and edge computing environments, as well as for high-performance computing across multiple low-performance computational nodes. The possibility of cross-layer separation of artificial neural networks using orthogonal transformations is theoretically substantiated, and practical examples of such an approach are provided. A comparison of the characteristics of modular neural networks using various types of orthogonal transformations, including the Haar wavelet transform, is conducted.**

***Index Terms*—orthogonal transformations, modular artificial neural networks, neural network optimization, wavelet transformations**

I. Introduction

The practical need to represent functions of n variables as a superposition of functions from a smaller number of variables arose due to the development of the theory and practice of neural networks. The basis of Artificial Neural Networks (ANN) is the Kolmogorov-Arnold theorem [1], [2], which showed the possibility of representing a continuous real function of n variables $f(x_1, x_2, \ldots, x_n)$ as a superposition of functions of a smaller number of variables.

A.N. Gorban [3] concludes that while the Kolmogorov-Arnold theorem guarantees the exact representation of functions of many variables in the class of continuous functions, the practical computation of most functions is only approximate even when exact formulas are available. The solution lies in approximating the function $f(x_1, x_2, \ldots, x_n)$ on a compact set Q using a sequence of polynomials (theorems of Weierstrass, Stone). Furthermore, functions can be approximated through linear operations and superpositions of one-variable functions [3]. This approach gained popularity after the works of McCulloch and Pitts [4], which predicted the emergence of ANN. The Hecht-Nielsen theorem [5] was a significant advancement in ANN, demonstrating the possibility of approximating a multi-variable function with a single hidden layer ANN in a non-constructive manner. However, the single-layer perceptron based on the Hecht-Nielsen theorem demonstrated low efficiency.

The emergence of multilayer ANNs and the development of methods for their training have made it possible to solve problems of classification, extrapolation, feature extraction, etc. under conditions of high uncertainty, i.e. to obtain satisfactory results with a sufficiently small training sample size. Due to the very rapid growth of the amount of data generated and the need to process it, ANNs also increase in size and require significant expenditure of computational resources. Natural Language Processing (NLP) is of great interest and also requires very large ANNs. For example, the popular GPT-3 ANN created in 2020 uses 175 billion parameters. It is clear that such ANNs require very high computational costs to run, which can only be achieved in cloud data centres.

Simultaneously, there is an increasing demand for data processing in close proximity to the equipment being used. This demand has led to the rise of edge computing and fog computing [6], which are becoming more popular due to their enhanced information security and the limited communication channels used for cloud computing. However, the computational nodes of fog computing typically lack the necessary power to run ANNs. Therefore, ANNs must be optimized by reducing the network size while only slightly degrading performance. Furthermore, modular artificial neural networks (ANNs) have been developed on multiple computational nodes [7], [8], creating distributed computing structures. However, the proposed ideas for modular ANNs are based on the assumption that network layer separation is impossible [9], and therefore rely on sequential separation into modules, one layer at a time. However, this approach does not address the main issue of resource estimation, as the number and performance of available fog computing nodes are typically unknown beforehand. The same problem arises when utilising a swarm of unmanned aerial vehicles (UAV). A single vehicle, equipped

The research was supported by the Russian Science Foundation Grant No. 24-21-00149, https://rscf.ru/en/project/24-21-00149/.

with a low-power computational node, cannot perform ANN operations. By leveraging the computing capabilities of a UAV swarm, it becomes feasible to operate ANNs of significant size. In this case, it is crucial to optimize the amount of information transferred between computational nodes. When separating the ANN layer by layer, the amount of information is significant due to the large number of parameters in each layer.

Ahmed and Rao [10] present their approach to building image recognition systems with optimal architecture. They suggest using orthogonal transformations to optimize image recognition algorithms, which reduces the number of significant features and the size of the classifier, a forward propagation ANN. The authors propose a concept of optimization and partitioning into ANN modules based on this approach.

II. Utilizing Orthogonal Transformations for Optimization of Neural Networks and Modularization

Dimensionality reduction is a transformation of data from a high-dimensional dataset to a lower-dimensional vector by eliminating uninformative features while preserving the structure and information contained within them to the maximum extent possible [12]. This transformation typically involves two steps: feature generation and selection [13]. In the first step, features that most comprehensively describe the research object are identified, while selection involves identifying features with the best classification properties for the given task. Commonly used methods for dimensionality reduction include Principal Component Analysis (PCA) [14], Factor Analysis (FA) [15], Linear Discriminant Analysis (LDA) [15], Singular Value Decomposition (SVD) [16], Kernel PCA [17], Independent Component Analysis (ISA) [18], Matrix Factorization [19], among others. However, they all have a significant drawback: dimensionality reduction requires preprocessing of information, which can sometimes demand considerable time and computational resources.

N. Ahmed and K. R. Rao [10] proposed optimizing the structure of the input signal through orthogonal transformations, rather than the ANN architecture. This approach is of interest because it allows for the realization of ANNs based on existing principles, approaches, and libraries. By reducing the amount of the input signal, it is possible to decrease the size of the ANN. If the input signal is divided into modules, processing can be carried out by several ANN modules. The orthogonal transformations discussed in the works of N. Ahmed and K. R. Rao can be considered as the first step, i.e., feature generation.

The method of orthogonal transformations is a well-known technique associated with the concept of orthogonal functions. A set of functions of a real variable, denoted by $\{d_i(t)\} = \{d_1, d_2, \ldots, d_i\}$, is considered orthogonal on the segment $[0; T]$ if the following condition is satisfied:

$$\int_0^T d_i(t)\, d_j(t)\, dt = \begin{cases} k \ if \ i = j, \\ 0 \ if \ i \neq p_i. \end{cases} \tag{1}$$

where k is the autocorrelation coefficient of the function $d_i(t)$.

If $x(t)$ is a function of a real variable on the interval $[0; T]$, it can be represented as a series

$$x(t) = \sum_{n=1}^{\infty} a_n d_n(t). \tag{2}$$

In (2), a_n represents the expansion coefficients, which can be determined by

$$a_n = \frac{1}{k} \int_0^T x(t)\, d_n(t)\, dt.$$

From the definition of a closed (complete) orthogonal set, it can be inferred that a function $x(t)$ can be represented as a finite set of expansion coefficients $\{a_0, a_1, \ldots, a_n\}$ by decomposing it into orthogonal functions. Even if the set of orthogonal functions is not closed, a finite set of coefficients can still be used. In this case, the representation of the function $x(t)$ is not exact, but rather an interpolation based on a certain criterion. The most common criterion used is the least squares principle, which is defined by the functional.

$$\Phi = \int \left(x(t) - \sum_{n=0}^{l} a_n d_n(t) \right)^2. \tag{3}$$

Once the value of the small ε has been determined, the number of members of the series can be calculated so that the condition $\Phi \leq \varepsilon$ is satisfied.

The Fourier transform is a well-known orthogonal transformation that is widely used in the theory of information processing and transmission. It allows for the transition from the time representation of a signal to the frequency representation and vice versa. In this context, we will consider the application of an orthogonal transformation based on the Fourier transform [10], [11]. In some cases, functions such as Lagger, Lejandre, Hermite, Walsh, Chebyshev, Adamar, etc. may be more appropriate than trigonometric functions as a kernel. As the input and output signals are represented discretely, it is possible to use variants of the discrete Fourier transform, including the Discrete Cosine Transform (DCT) [10]. The DCT is commonly used in the JPEG format for lossy image compression and operates exclusively with real numbers. The DCT utilises a set of basis vectors in the form of [10]

The DCT employs a set of basis vectors in the form of [10], which is explored on the interval $[0, \pi]$:

$$\left\{ \frac{1}{\sqrt{n}}, \sqrt{\frac{2}{n}} cos \frac{(2m+1)k\pi}{2n} \right\}. \tag{4}$$

Here, k represents the harmonic (coefficient) number, and $m = 0, 1, 2, \ldots, n-1$, representing the size of the initial data array. Equation (4) represents a class of discrete Chebyshev polynomials [10]. The DCT has a notable property where the basis vectors approximate the eigenvectors of the Toeplitz matrices, allowing for effective compression of the original signal using DCT. Therefore, applying an orthogonal transform

of the form (1) generates a set of coefficients. In the case of the Fourier transform and its variant DCT, these coefficients represent the amplitudes of frequency harmonics. In this case, the signal can be completely restored or restored with a certain level of accuracy by summing the harmonic components (inverse Fourier transform), depending on the number of components summed.

For the second step (feature selection), N. Ahmed and K. R. Rao [10] utilized the root-mean-square deviation (RMSD) of coefficients in their studies. The higher the RMSD of a coefficient, the more pronounced its classification capability. However, this measure is inconvenient as it requires additional processing, similar to other dimensionality reduction methods. While the issue of orthogonal transformation in the first layer of neural networks was addressed [23], a fundamentally different approach is needed for real-time feature selection.

When transitioning to the investigation of a function presented in both time and frequency domains, it is necessary to delineate the distinction in its reconstruction in each case. The representation of a continuous function by discrete samples in time is defined by the Whittaker-Kotelnikov-Shannon theorem [20], which states that the sampling rate should exceed the maximum frequency of the signal by a factor of two or more. Each sample characterizes the instantaneous value of the function at time t_i. The expansion coefficient a_i represents the projection of the function onto the i-th orthogonal function of the chosen basis over the interval $[0; T]$. Moreover, a higher value of a_i indicates a closer affinity of the investigated function to the i-th component. We shall employ the concept of approximation accuracy for closed systems of orthogonal functions [21]. Based on the Lyapunov-Steklov (Parseval) equality:

$$B = \int_0^T x^2(t)\,dt = \sum_{n=0}^{\infty} a_n^2,$$

Then the relative integral accuracy of the approximation can be estimated as

$$\gamma = \frac{\sum_{n=0}^{k} a_n^2}{B} = \frac{a_0^2}{B} + \frac{a_1^2}{B} + \ldots + \frac{a_n^2}{B}. \quad (5)$$

In (5), each term characterizes the contribution of each projection to the formation of the original function. Based on this, one can speak about the accuracy of approximating a function obtained from a limited number of components (coefficients).

Thus, an approximate function ($\gamma < 1$) can be used for training ANNs, consequently reducing the size of the ANN. From the course of mathematical analysis, there is a known relationship between the smoothness of a function and the rate of decrease of Fourier coefficients: if a function on an interval has piecewise continuous derivatives of the first and higher orders, then it converges to the original function absolutely and uniformly [22]. Hence, it follows that the partial sums of the coefficients decrease as the component harmonic numbers increase. Therefore, dividing the spectrum can be divided into two parts, based on equal number of frequencies, then we obtain the following values of integral approximation accuracy for each half of the spectrum:

$$\gamma_1 = \frac{\sum_{n=0}^{k/2} a_n^2}{B}, \quad (6)$$

$$\gamma_2 = \frac{\sum_{n=k/2+1}^{k} a_n^2}{B}. \quad (7)$$

Since the spectral density in (7) will be smaller than in (6), the accuracy of approximation by the neural network trained on the first part of the spectrum will be higher than that of the second. This will be demonstrated experimentally in Section 3. Depending on the problem being solved, the number of parts into which the spectrum of the input function is divided may vary. Additionally, the parts of the spectrum do not necessarily have to be identical. If there are several computational nodes of higher power, a larger portion of the spectrum can be allocated to them using a larger size ANN.

By using wavelet transforms instead of Fourier-like transforms, we can combine the operations of feature generation and selection and obtain a gain in the number of relevant features without additional research [24]. In general, wavelets are a system of functions of the following form:

$$\varphi_{a,b}(x) = \sqrt{2^a}\varphi(2^a x - b). \quad (8)$$

If V_a is the space spanned by the system of functions (8), then the following inclusions hold [25]: $V_0 \subset V_1 \subset \ldots \subset V_a$. Thus, we obtain a sequence of nested subspaces $V_i \subset L^2(R)$, each equipped with an orthonormal basis $\{\varphi_{i,b}(x)\}$. This sequence of subspaces can be used to approximate a function $f(x)$ from $L^2(R)$ by its projection operator

$$P_a : L^2(R) \to V_a, P_a(f) = \sum_{b \in Z} (f, \varphi_{a,b})\varphi_{a,b}(x).$$

The projections P_i become increasingly accurate approximations of $f(x)$ as i increases. Returning to neural networks, the following analogy can be drawn. A set of input vectors $f_i(x_j)$ in the $L^2(R)$ space can be projected onto a set of subspaces $V_0 \subset V_1 \subset \ldots \subset V_a$ such that each projection P_i is an approximation of the input data. By training a neural network on the projections P_0, we obtain the coarsest approximation of the expected outcome. However, due to decimation, this will be the most "compact" approximation, requiring minimal computational resources for operation.

The widely known Haar wavelet [10], [25], [26] allows for partitioning the $L^2(R)$ space into two subspaces, V_0 and V_1. By using V_0 as the base subspace, a modular architecture of neural networks can be constructed, enabling the use of a basic module for low-power devices [24]. Essentially, the Haar wavelet performs a partitioning of the coefficient space into two equal parts [25], as previously proposed. This allows for solving the problem as described above, where the basic module facilitates the application of neural networks on low-computational-power devices with minimal loss of accuracy and without additional training. In practical applications of the proposed theoretical principles, it is important to consider

that such transformation can be repeated for each half of the coefficients. In this case, the entire space can be partitioned into 4 parts and a 4-module structure can be formed, and so forth.

Wavelet transformation divides the coefficient space into several equal parts. However, if there exist nodes with high computational capabilities in a distributed neural network, multiple modules can be assigned to such a node. Consequently, the computational load can be distributed more evenly.

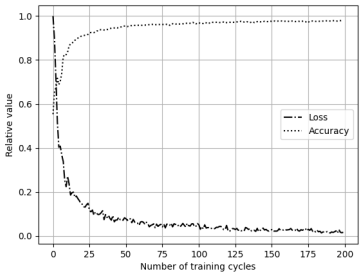
The following conclusion can be drawn from this: orthogonal transformations allow you to divide the input signal space into segments (modules). At the same time, due to (1) modules can be processed independently of each other on different nodes of a distributed computing system. Combining the information processed on different nodes can be realised due to the possibility of inverse orthogonal transformation. However, due to the nonlinearity of ANN, the application of the inverse transformation core is usually impossible and requires training of the layer that combines the results of the ANN modules. In this case, the training of the unifying layer is possible together with the training of the modules.

Therefore, the implementation of distributed (modular) ANN requires a number of computational nodes that corresponds to the number of modules to be organised, as well as two additional nodes: one to perform orthogonal transformation and another to combine the results of the modules. This organization optimizes the amount of information transmitted through communication channels. The amount of information transmitted from the orthogonal transformation module to the ANN modules is not greater than the amount of the input signal. Additionally, the amount of information transmitted from the modules to the unifying layer is l times greater than the amount of output data of ANN, where l is the number of modules in the ANN. All other information is transmitted within the layers of the modules and does not require communication channels.

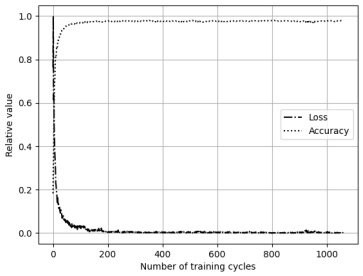
III. Practical Demonstration of Optimization Potential for ANNs and Cross-Layer Network Partitioning for Deployment on Edge and Fog Computing Nodes

Thus, by performing orthogonal transformation and dividing the coefficients into modules, for example, with the help of ideal digital filters, we obtain a group of feature modules, which can be used to train several ANN modules. In this case, each layer in the ANN modules is reduced proportionally to the amount of the coefficient modules and does not require from the computational node such a high performance that is required for the ANN as a whole. The real discrete Fourier transform and DCT were chosen for the experiment. The first step is to evaluate the quality of ANN training in time and frequency domains.

The PyTorch library [28] was used to implement the ANN, with the MNIST database [30] serving as the training data. The experiment was conducted on a personal computer running the



(a) ANN training in the time domain



(b) Training of ANN in the frequency domain

Fig. 1: ANN learning processes in time and frequency domains using the Fourier transform.

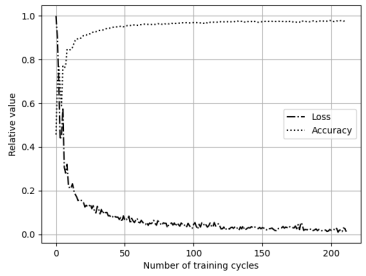
Windows 10 Home operating system, equipped with an Intel Core i7-10510U processor and 16 GB of RAM.

Fig. 1 shows that training of ANN in the frequency domain using Fourier transform takes 5 times more cycles than in the time domain.

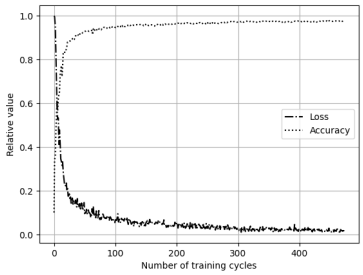
However, the application of DCT showed a slightly different picture (Fig. 2). ANN training in the frequency domain outperformed training in the time domain by only a factor of 2. This can be explained by the compression properties of DCT mentioned earlier.

The problems of optimisation and design of distributed ANNs have a common solution: in optimisation, as many modules are placed on a computing node as the node's computing power allows. And in the design of distributed ANNs, the modules are placed on different nodes connected by communication channels.

Consider the construction of modular ANNs. The first layer performs an orthogonal transformation based on the weights set and fixed in the neurons of the first layer [24]. Furthermore, the transformation coefficients obtained after passing through the first layer are divided into two equal parts and sent for processing to different ANNs with layer sizes reduced by a factor of 2. The outputs of the two ANNs are combined in a layer with 20 inputs and 10 outputs. Fig. 3 shows the training processes of the modular ANNs on MNIST data presented in the frequency domain using Fourier transform (Fig. 3a) and using DCT (Fig. 3b). The learning rate is approximately the same and comparable to the learning rate of the monolithic

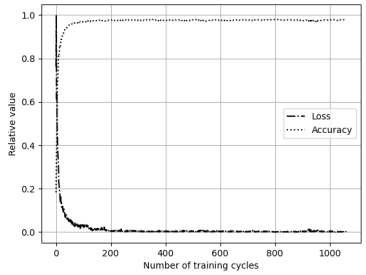


(a) ANN training in the time domain

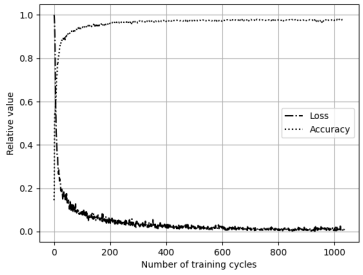


(b) Training of ANN in the frequency domain

Fig. 2: ANN learning processes in time and frequency domains using DCT.



(a) Training of 2-module ANN (Fourier transform)



(b) Training of the 2-module ANN (DCT)

Fig. 3: Learning processes of 2-module ANNs.

ANN in the frequency domain.

The characteristics of the modular ANNs obtained for optimisation and construction of distributed neural networks are considered. Table I presents the characteristics of 2-modular ANN.

TABLE I: Characteristics of the 2-module ANN

	Recognition quality, %	Average time of 1 cycle, sec
Module 1	79.63	0.029
Module 2	69.02	0.024

Table I confirms that the recognition quality of the second module is lower than that of the first module. This is due to the fact that the integral energy spectrum used to train the first module was larger than that used for the second module.

Table II presents the characteristics of the four-module ANN.

TABLE II: Characteristics of a 4-module ANN

	Recognition quality, %	Average time of 1 cycle, sec
Module 1	78.76	0.013
Module 2	37.39	0.012
Module 3	17.49	0.013
Module 4	20.86	0.012

Table II shows that the average execution time per cycle for all modules is approximately the same, while the recognition quality decreases as the module number increases. Module 4 is the exception, with recognition quality exceeding that of module 3. This can be explained by the high-frequency oscillations at the object boundaries, which cause the input signal to the ANN to be not completely smooth.

Table III presents the characteristics of the 4-module ANN when the modules are connected in turn.

TABLE III: Characteristics of ANN when several modules are connected

	Recognition quality, %	Average time of 1 cycle, sec
Module 1	78.76	0.013
Module 2	94.01	0.017
Module 3	97.22	0.022
Module 4	98.05	0.028

Table III shows the improvement in ANN recognition quality with the addition of modules. The average cycle time increases at a slower rate than the number of connected modules. Therefore, the time required for four modules is only twice that of one module. This is due to the parallelism of the modules working simultaneously.

The Haar wavelet transform is used as an orthogonal transform for constructing a modular ANN. The procedure for wavelet transformation involves passing the input signal

through a half-band digital filter with frequency response $h(n)$ (high-pass filter) or $g(n)$ (low-pass filter) [25], [27]:

$$\begin{cases} x(n) * h(n) = \sum_k x(k)\, h(n-k), \\ x(n) * g(n) = \sum_k x(k)\, g(n-k). \end{cases}$$

If the input signal of the ANN is a one-dimensional series of numbers with a length of n, we can obtain wavelet transform coefficients at the output by using a one-dimensional convolution layer with either kernel $h(n)$ or $g(n)$. To reduce the number of ANN layers, we can use one layer with two or more different kernels. To achieve this, we create a convolution layer with one input, two or more outputs, and a step equal to the dimensionality of the wavelet kernel. In this case, a convolution layer will be created with two kernels, where the values of $h(n)$ and $g(n)$ are inputted [25].

TABLE IV: Characteristics of 2-module ANN using wavelet transform

	Recognition quality, %	Average time of 1 cycle, sec
Module 1	97.01	0.019
Module 2	53.12	0.019

The Haar wavelet transform, as shown in Table IV, distributes significant features in the frequency-time matrix more strictly. This allows for the separate use of the first module with $\approx 1\%$ loss in accuracy. Additionally, the module's speed is slightly increased due to the use of the convolutional layer as an orthogonal transformer. The use of wavelets for constructing modular ANNs is discussed in more detail in [27].

IV. Conclusion

The article discusses the use of orthogonal transformations, specifically the Fourier transform, discrete cosine transform, and Haar wavelet transform, for constructing distributed modular ANNs. The approaches outlined in detail in [10] enable the use of these transformations for training ANNs and dividing the input vector into parts for modular network application. The examples provided demonstrate the potential for optimising ANNs for use on low-performance computing devices. They also enable the creation of distributed computing systems with performance equal to that of a monolithic network. The proposed approach is considered advantageous due to its independence from the ANN architecture. This allows for the separation of input information and a reduction in the size of modules, which can be applied to any neural network architecture using standard libraries.

Acknowledgment

The research was supported by the Russian Science Foundation Grant No. 24-21-00149, https://rscf.ru/en/project/24-21-00149/.

References

[1] Kolmogorov A. N. O predstavlenii nepreryvnyh funkcij neskol'kih peremennyh v vide superpozicij nepreryvnyh funkcij odnogo peremennogo i slozheniya [On the representation of continuous functions of several variables as superpositions of continuous functions of one variable and addition]. Doklady Akademii nauk [Reports of the Academy of Sciences], 1957, vol. 5, pp. 953-956 (In Russian).

[2] Arnol'd V. I. O predstavlenii funkcij neskol'kih peremennyh v vide superpozicii funkcij men'shego chisla peremennyh [On the representation of functions of several variables as a superposition of functions of a smaller number of variables]. Matematicheskoe prosveshhenie [Mathematical education], 1958, vol. 3, pp. 41-61 (In Russian).

[3] Gorban' A. N. Obobshchennaya approksimacionnaya teorema i vychislitel'nye vozmozhnosti nejronnyh setej [Generalised approximation theorem and computational capabilities of neural networks]. Sibirskij zhurnal vychislitel'noj matematiki [Siberian Journal of Computational Mathematics], 1998, vol. 1, pp. 11-24 (In Russian).

[4] McCulloch W. S., Pitts W. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 1943, vol. 5, is.4, pp. 115-133.

[5] Hecht-Nielsen R. Neurocomputing. Addison-Wesley, 1989.

[6] Kirsanova A.A., Radchenko G.I., Chernykh A.N. Obzor tekhnologij organizacii tumannyh vychislenij [Review of technologies of fog computing organization]. Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta [Bulletin of the South Ural State University], 2020, vol. 9, no. 3, pp. 35-63, DOI: 10.14529/cmse200303 (In Russian).

[7] Mao, J., Chen, X., Nixon, K.W., Krieger, C., Chen. MoDNN: Local distributed mobile computing system for deep neural network. Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, Lausanne, Switzerland, 2017, pp. 1396-1401, doi: 10.23919/DATE.2017.7927211.

[8] Bakhtin V.V. Algoritm razdeleniya monolitnoj nejronnoj seti dlya realizacii tumannyh vychislenij v ustrojstvah na programmiruemoj logike [Separation algorithm of the monolithic neural network for realization of fog computing in devices on programmable logic], Vestnik PNIPU. Elektrotekhnika, informacionnye tekhnologii, sistemy upravleniya [Bulletin of PNIPU. Series: Electrical engineering, information technologies, control systems, 2022, vol. 41, pp. 123-145, doi: 10.15593/2224-9397/2022.1.06 (In Russian).

[9] Ushakov Y.A., Polezhaev P.N., Shukhman A.E., Ushakova M.V. Raspredelenie nejronnoj seti mezhdu mobil'nym ustrojstvom i servisami oblachnoj infrastruktury [Distribution of the neural network between mobile device and cloud infrastructure services]. Research and development in the field of new IT and their applications, 2018, vol. 14, no.4. pp. 903-910, doi: 10.25559/SITITO.14.201804.903-910 (In Russian).

[10] Ahmed N., Rao K.R. Ortogonal'nye preobrazovanija pri obrabotke cifrovyh signalov [Orthogonal transforms for digital signal processing]. Moscow, Svjaz' publ., 1980. (In Russian).

[11] Solodov A.V. Information theory and its application to the tasks of automatic control and monitoring. Izdvo "Nauka", glav. red. fiziko-matematicheskoj lit-ry [Publishing house "Science"], 1967 (In Russian).

[12] Burges C.J.C. Dimension reduction: A guided tour. Foundetions and Trends in Machine Learning, 2010, vol.2, no. 4, pp. 275-365, DOI: 10.1561/2200000002.

[13] Erohin S.D., Borisenko B.B., Martishin I.D., Fadeev A.S. Analiz sushchestvuyushchih metodov snizheniya razmernosti vhodnyh dannyh [Analysis of existing methods to reduce the dimensionality of input data]. T-Comm: Telekommunikacii i transport [T-Comm: Telecommunications and Transport], 2022, vol. 16, no. 1. pp. 30-37 (In Russian).

[14] Jolliffe I.T. Principal component analisis. Second Edition, Springer, 2007, 487 p.

[15] Mardia K.V., Kent J.T., Bibby J.M. Multivariate analysis (Probability and mathematical statistics). Academic Press Limited, 1995, 521 p.

[16] Stewart G.W. On the early history on the singular value decomposition. SIAM Review, 1993, vol. 35, no. 4, pp. 551-566. DOI: 10.1137//1035134.

[17] Van Der Maaten L., Postm, E. O., Van den Herik H. J. Dimensionality reduction: A comparative review. Journal of Machine Learning Research, 2009, vol. 10, 13 p.

[18] Hyvarinen A., Karhunen J., Oja E. Independent component analysis. John Wiley and Sons, 2001, 504 p.

[19] Snasel V., Horak Z., Kocibova J., Abraham A. Reducing social network dimensions using matrix factorization analysis. Proceedings of the 2009

International Conference on Advances in Social Network Analysis and Mining, 2009, pp. 348-351. DOI: 10.1109/ASONAM.2009.48

[20] Jerry A. J. Teoriya otschetov SHennona, ee razlichnye prilozheniya i obobshcheniya [Shannon's reference theory, its various applications and generalisations]. Obzor TIIER [Review TIIER], 1977, vol. 65, no. 11, pp. 53 - 89 (In Russian).

[21] Dedus F.F., Kulikova L.I., Pankratov A.N., Tetuev R.K. Klassicheskie ortogonal'nye bazisy v zadachah analiticheskogo opisaniya i obrabotki informacionnyh signalov [Classical orthogonal bases in problems of analytical description and processing of information signals]. FVMIK MGU [FCMIC OF MSU], 2004, 168 p. (In Russian).

[22] Zorich V. A. Matematicheskij analiz [Mathematical analysis]. Izdatel'stvo Nauka, Glavnaya redakciya fiziko-matematicheskoj literatury [Nauka Publishing House, Main Editorial Office of Physical and Mathematical Literature], 1984, pp. 637 (In Russian).

[23] Vershkov N.A., Kuchukov, V.A., Kuchukova, N.N., Babenko M., The Wave Model of Artificial Neural Network. Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, EIConRus, 2020, pp. 542-547, DOI: 10.1109/EIConRus49466.2020.9039172

[24] Vershkov, N., Babenko, M., Tchernykh, A., Kuchukov, V., Kucherov, N., Kuchukova, N., Drozdov, A. Y. Optimization of Artificial Neural Networks using Wavelet Transforms. Programming and Computer Software, vol. 48, no. 6, pp. 376-384, https://doi.org/10.1134/S036176882206007X

[25] Smolencev N.K. Osnovy teorii vejvletov. Vejvlety v MATLAB [Fundamentals of wavelet theory. Wavelets in MATLAB]. DMK Press, 2019. (In Russian).

[26] Haar A. Zur theorie der orthogonalen funktionensysteme. Georg-August-Universitat, Gottingen, 1909.

[27] Vershkov N.A., Babenko M.G., Kuchukova N.N., Kuchukov V.A., Kucherov N.N. Transverse-layer partitioning of artificial neural networks for image classification. Computer Optics, 2024, vol. 48, no. 2, pp. 312-320. DOI: 10.18287/2412-6179-CO-1278.

[28] PyTorch. Source: https://pytorch.org/

[29] PyWavelets. Source: https://pypi.org/project/PyWavelets/

[30] Qiao Y. THE MNIST DATABASE of handwritten digits. 2007. Source: http://www.gavo.t.utokyo.ac.jp/qiao/database.html.