Vol. 08, No. 2 (2026) 887-894, doi: 10.24874/PES08.02A.004

Proceedings on Engineering Sciences

www.pesjournal.net

INTELLIGENT DATA MINING BASED ON MULTIDIMENSIONAL NEURAL NETWORK INTERPOLATION IN ARTIFICIAL INTELLIGENCE SYSTEMS

Evgeny Andreevich Nekrasov [1]
Alexey Vladimirovich Degtyar
Tatyana Anatolyevna Markina
Nikolai Nikolaevich Zhukov

Received 20.01.2026.
Revised 12.05.2026.
Accepted 29.05.2026.

Keywords:

Artificial intelligence, Generalization, Similarity, Training samples, Multivariate nonlinear interpolation, Extrapolation

A B S T R A C T

In practice, engineering and scientific decisions are rarely made from scratch. We rely on what we already know on analogies, on similarity, on interpolation. That's especially true when we're dealing with new systems, unknown operating conditions, or noisy data. Modern AI, particularly deep learning, often tackles this by simply feeding the network more and more examples. But that approach hits a wall eventually data is always finite, and real-world scenarios are infinitely variable.

This paper looks at the link between mathematical interpolation and how neural networks generalize. Our argument is that if we treat generalization as a form of multidimensional nonlinear interpolation, we can build networks that do more than just classify they can actually synthesize reasonable responses to inputs they've never seen before. That's a big deal for control systems, for engineering design, and for any field where you can't afford to retrain a model every time conditions change.

Beyond the theory, the approach has clear practical value. In manufacturing, it enables smarter adaptive control and process optimization on the factory floor, reducing downtime and improving efficiency. In management information systems, it supports better forecasting, resource planning, and data-driven decision making under uncertainty. In materials science, it offers a pathway to cut down on expensive physical testing by reliably predicting how materials behave in new environments, thereby accelerating innovation and reducing development costs.

© 2026 Published by Faculty of Engineering

1. INTRODUCTION

The role of many cybernetic systems isn't just to mimic the structure of biological neural networks in order to simulate intelligent behavior what's often called the bionic approach. More importantly, these systems are expected to perform specific functions that are characteristic of what we might call "strong" artificial intelligence (Luger, 2003). One of the most critical of these functions is the ability to make reliable, deductive inferences by automatically generalizing from input data

[1] Corresponding author: Evgeny Andreevich Nekrasov
Email: evgeniy@nekrasov.pw

(Khaikin, 2006; Sadyraliev et al., 2025). This ability is what gives AI systems much of their practical power.

A quick look back shows how things have evolved. For a long time, statistical methods, analytical techniques, and expert systems were the go-to tools. But they were gradually replaced by machine learning, which turned out to be much better at discovering patterns and rules directly from data without requiring humans to predefine possible solutions. This shift has led to AI systems that are not only more productive but also more adaptive and user-oriented.

Still, there's a fundamental challenge: a truly strong AI must be capable of generalization. And generalization, at its core, is deeply tied to the mathematical idea of interpolation. In practice, many real-world problems—whether in engineering, control, or design—require making decisions by analogy. That often means relying on similarity theory and interpolation methods, especially when we're dealing with new equipment designs or control scenarios where the behavior of the system isn't fully known.

Take dynamic systems, for example. When we need to control such a system under new conditions, we often find that we don't have a reliable model for it especially when different combinations of disturbances and control inputs come into play. To make things worse, the data we do have is usually noisy, and the quality of derivative estimates (particularly higher-order ones) degrades quickly as noise increases.

This is where neural networks come in. Over time, they've become a popular tool for tackling these kinds of problems, particularly in the bionic branch of AI. In neuroinformatics, solving a problem "by analogy" is essentially a form of generalization. And in many AI systems, generalization is achieved by throwing more data at the network training it on larger and larger datasets. But datasets are always finite. No matter how much data we collect, there will always be new cases the network hasn't seen. So the reliability of these systems is inherently limited.

Our goal in this paper is to take a closer look at the relationship between interpolation and generalization in neural networks. We argue that if we frame generalization as a form of multidimensional nonlinear interpolation, we can build networks that reliably synthesize responses to new, unseen inputs not just classify them. This is different from conventional classification, where we assign inputs to predefined categories. Here, we're talking about a more flexible, synthesis-driven kind of reasoning.

We also explore how this approach can be applied in practice particularly in managing complex systems and designing new types of equipment. Through examples like PID controller tuning and precedent-based control, we show that neural interpolation offers a practical and reliable alternative to traditional methods.

Finally, we hope our findings will help guide future research. Specifically, we point to the need for better, more thoughtful neural network architectures that are designed with generalization in mind—not just trained on larger datasets, but built to handle the unknown more gracefully.

2. RESEARCH METHODS AND RESEARCH RESULTS

The research methods we use sit at the intersection of several mathematical and computational fields. Linear algebra and matrix analysis form the backbone of our work, giving us the language to describe how data flows through a neural network, how weights interact with inputs, and how outputs are produced. These are not abstract tools for us. They are practical instruments that let us see inside the network and understand what it is actually doing.

Alongside that, we rely heavily on pattern recognition theory. This helps us think about how a network learns to spot regularities in data, how it separates signal from noise, and how it builds the internal representations that make generalization possible. Without this lens, it is easy to mistake memorization for learning.

And then there is neuroinformatics, which ties everything together. This field sits at the crossroads of neuroscience, information science, and machine learning. It gives us a way to think about neural networks not just as mathematical objects, but as systems that are trying to do something remarkably human, which is to learn from experience and apply that knowledge in new situations.

Together, these methods give us a coherent framework for exploring the connection between interpolation and generalization. More importantly, they give us a practical path forward for building AI systems that can handle the messiness of real engineering and industrial problems, where data is noisy, conditions change, and there is no such thing as a perfect training set

2.1 The relationship between generalization in artificial intelligence systems and mathematical interpolation

The mechanisms of "strong" artificial intelligence allow for reliable classification of new samples that are not included in the training samples. Although the term "classification" is not applicable here, since classification is a division (taxonomy) and assignment to a given. In our case, the sought-after function of artificial intelligence implements automatic synthesis of a description of a new sample based on an input tuple of previously unobserved signals. "Previously unobserved" refers to both absolute values at the input of the

information system and their combinations (Michelucci, 2020).

The basis for the high reliability of machine learning models is the generalization effect, when examples that are not part of the training sample are well recognized as part of the verification or test sample (Bishop, 2020).

Multilayer and other network architectures, due to the continuous, continuously differentiable and nonlinear activation function of neurons, as well as other architectural constructions, allow us to approximate the input training vectors so that the values of the components of the verification or test vectors are interpolated (in the narrow sense) quite accurately (Khaikin, 2006).
In fact, such an interpolating network is represented by a single nonlinear multidimensional operator (Imomov, 2001).

The network performs correct interpolation mainly due to the continuity (differentiability), nonlinearity of the neuron activation functions, and also through certain mathematical procedures, which for multilayer feedforward networks ensures the continuity of the overall output function (Figure 1) (Carmona et al., 2022).

However, the continuity of the overall output function and the reliability of the trained network are different concepts. Continuity ensures reliability only when it eliminates the " overtraining " of the network. That is, when the training samples are similar, and to distinguish them it is necessary to increase the accuracy of the approximation, or when there are too many examples and it is necessary to increase the number of layers, neurons in each layer, then, according to the training results, the network (with certain training algorithms) may be overtrained (it accurately restores the training sample examples, but practically does not recognize similar samples). This effect is observed during training in additive or (even worse) multiplicative noise, which is present in the training examples, but is not characteristic of the modeled mapping function itself (Aggarwal, 2020).

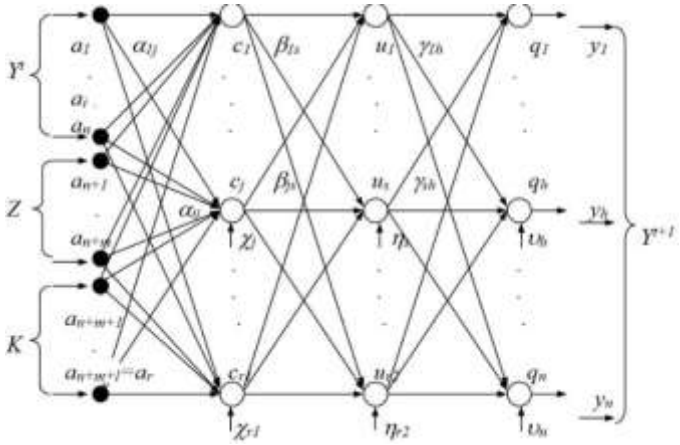


Figure 1. The architecture of the BProp neural network

Due to the redundancy of the output layer, which implements the technology of multidimensional nonlinear interpolation, interpreted as a generalization to inter-interval examples that are not included in the training sample (Y^t is the vector of the current value, Z is the state vector, K is the vector of disturbing effects, Y^{t+1} is the vector of the desired vector)/

With regard to the generalization of complex (closed and disconnected) topologies in the parameter space of training samples, which are encountered when solving problems of control and designing complex technical systems, the continuity and nonlinearity of the activation functions of neurons is no longer sufficient to ensure the reliability of the synthesis of new input samples.

The problem of generalization in multilayer fully connected neural networks, in isolation from mathematical interpolation, was one of the first to be experimentally studied by F. Rosenblatt (Rosenblatt, 1965). He believed that "… a direct propagation neural network is capable of moving from a selective reaction to one image to a generalizing reaction to a "similar" image, the features of which can be completely different from the previous one due to the activation of the "ability" to extract the most frequently encountered transformations from the environment and then to recognize similarity and difference in relation to these transformations of previously unencountered images ."

In the course of research (on input images similar, mainly, in a group of one-to-one transformations), the fourth direction of development of generalizing characteristics of multilayer NNs was established - generalization by similarity, the recognition rule for which states: *"Any fragment of the NN that reacts to the cause X_1 will also react to all mappings $\Xi(X_1)$ of this cause in the group of transformations $\mathfrak{N}$with probability p."*

From a mathematical point of view, the third and fourth types of transformation recognition methods by

generalization by similarity and dominance are a neural network interpretation of mathematical nonlinear interpolation in a multidimensional feature space.

2.2 Multidimensional nonlinear neural network interpolation

The problem of interpolation is one of the main tasks of numerical methods (Romm et al., 2022; Bakhvalov et al., 2020). The hierarchy of approximation concepts is illustrated in Figure 2.

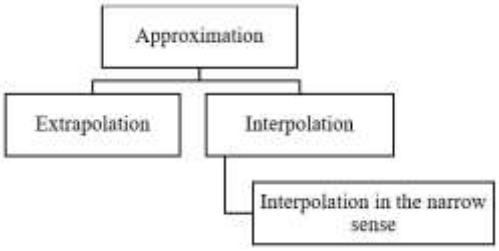


Figure 2. Interpolation in a narrow sense potentially allows for the generalization ability of artificial intelligence systems based on machine learning

The set of these points is sometimes given to us by external circumstances, in the case of machine learning – by parameterization of examples of the training sample (Konopatsky et al., 2021).

One of the special cases of approximation is interpolation, which implies that if the desired function $y(x)$ is given by the training sample in the form of a table, i.e. in the plan of the experiment on the grid $\{x_n, n = 0.1, ...\}$, at the nodes of which the values $y_{n\ are\ known} = y(x_n)$, then the problem is to construct a function that restores the values of $y(x)$ at an arbitrary point x. In this case, we must require a fairly simple behavior of $y(x)$: the function should not have "spikes" between neighboring nodes. Mathematically, this means that $y(x)$ should have a sufficient number of higher derivatives that are not too large in magnitude (Gasca & Sauer, 2022). Let us choose a system of linearly independent functions $\{f_m(x), m = 0.1, ...\}$. A linear combination of such functions is called a generalized polynomial $\Phi(x)$. Approximation of $y(x)$ by a generalized polynomial:

$$y(x) \approx \Phi_N(x) \equiv \sum_{m=0}^{N} c_m f_m(x), \quad (1)$$

where c_m are coefficients chosen with the condition that the generalized polynomial $\Phi_N(x)$, containing $N+1$ coefficients accurately conveys the tabulated values of the function at the $(N+1)$-th node:

$$\sum_{m=0}^{N} c_m f_m(x) = y_n, \qquad 0 \le n \le N. \quad (2)$$

This method of approximation is called interpolation, and the coefficients c_m are found from the solution of the linear system (2). For its solvability it is necessary that

$$\det\left[f_m(x_n) \right] \neq 0. \quad (3)$$

However, condition (3) is not necessary in the case of neural network approximation in the multilayer BProp architecture or modern varieties of deep learning networks (Figure 3) within the framework of machine learning technologies with multilayer networks (Lekun, 2021; Shalev-Schwartz & Ben-David, 2019; Asimov Institute, n.d.).

In the analysis under consideration, interpolation in the narrow sense of the word is of interest, which implies that the sought values of the function $f_m(x_1, x_2, x_3, \dots x_z)$ do not coincide with the nodes and lie in the intervals.

$$\begin{bmatrix} \left[x_1^1, x_1^2\right] & \left[x_1^2, x_1^3\right] & \left[x_1^3, x_1^4\right] & \dots & \left[x_1^{i-1}, x_1^i\right] & \dots & \left[x_1^{N-1}, x_1^N\right] \\ \left[x_2^1, x_2^2\right] & \left[x_2^2, x_2^3\right] & \left[x_2^3, x_2^4\right] & \dots & \left[x_2^{i-1}, x_2^i\right] & \dots & \left[x_2^{N-1}, x_2^N\right] \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \left[x_j^1, x_j^2\right] & \left[x_j^2, x_j^3\right] & \left[x_j^3, x_j^4\right] & \dots & \left[x_j^{i-1}, x_j^i\right] & \dots & \left[x_j^{N-1}, x_j^N\right] \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \left[x_M^1, x_M^2\right] & \left[x_M^2, x_M^3\right] & \left[x_M^3, x_M^4\right] & \dots & \left[x_M^{i-1}, x_M^i\right] & \dots & \left[x_M^{N-1}, x_M^N\right] \end{bmatrix}, \quad (4)$$

where $(N-1)$ is the number of intervals between data points of each of the M variables.

Problems of solving the problem of multidimensional interpolation in the narrow sense of the word arise when:

1) the measurement results are not carried out on a uniform time scale;
2) the arguments of the desired multidimensional function $f_m(x_1, x_2, x_3, \dots x_z)$ are not independent.

The areas of application of interpolation are very diverse, interpolation is especially relevant where it is necessary to develop solutions for managing complex technological

processes or designing new types of equipment based on similarity or analogies (the terms are also associated with generalization).

3. APPLICATION OF GENERALIZATION THROUGH INTERPOLATION

In many branches of science and technology there is a need to use the methodology of decision-making “by analogy” with the involvement of the methodological apparatus of similarity theory and methods of mathematical interpolation.

Most often, interpolation through generalization is required when solving problems of designing new types of equipment or control in new conditions of its application.

When solving numerous problems of dynamic systems control, it becomes necessary to evaluate the characteristics of the control object in order to improve the quality of regulation and control efficiency. This is often due to the fact that the model of the control object in new conditions (combinations of values of disturbance and control effects) is not identified.

This task is complicated by the fact that the function is usually observed against the background of noise of varying intensity, and with increasing noise intensity, the quality of the function derivatives (especially high-order derivatives) recovery deteriorates. An equally important issue arising in the problems of dynamic system control is the ability to predict the "future" state of the system, which most often leads to the need to solve the problem of predicting the signal arriving at the input of the dynamic system. Closely related to the above-mentioned problems is the problem of choosing the optimal precedent control based on the results of observing the behavior of the dynamic system. For example, PID controllers are widely used, which have optimal settings for a certain combination of characteristics of the control object variables and environmental factors. Changes in the operating modes of the object (abnormal, emergency) cause a decrease in the quality of regulation, accordingly, it is necessary to reconfigure the control coefficients, which is not always possible using labor-intensive methods of parametric synthesis. Assigning the desired PID control coefficients by means of extrapolation (or interpolation, depending on where the desired point is located relative to known precedents) allows solving the problem more quickly and no less reliably.

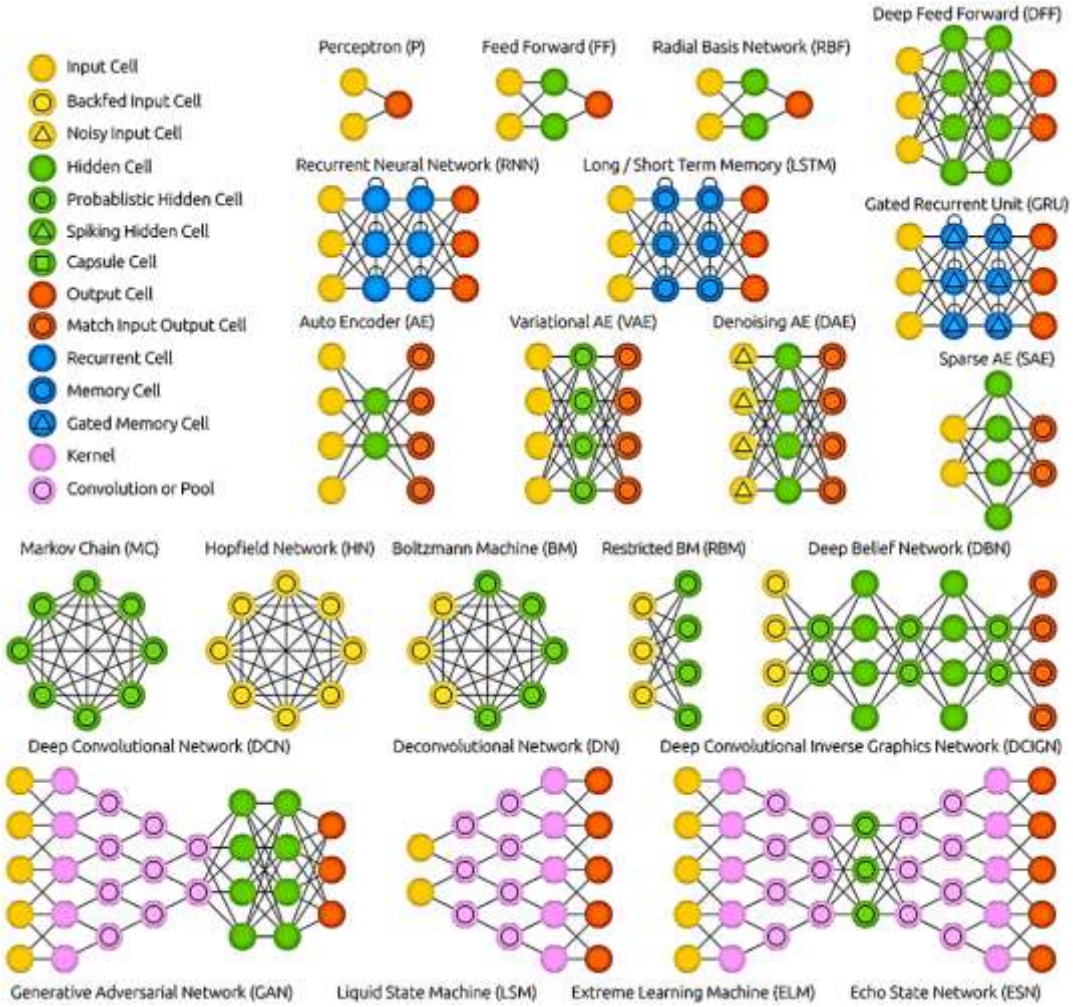


Figure 3. Modern neural network architectures with generalization

The control problem can be represented as the problem of transforming the values of the control situation parameters - X , described by three concatenated vectors (states of the control object, observed parameters of the object, disturbing effects) into a parameterized description of the quasi-optimal control action Y . Information is collected using a system of sensors located on the object and in its environment. The decision made on the basis of this information is the control Y , which should be implemented in the object in order to achieve the specified goals (they are assumed to be constant here). The relationship between X and Y in the control system is implemented using the control algorithm F^0:

$$Y = F^0(X), \tag{5}$$

for the operation of which, within the framework of approximation methods, it is not necessary to have a model of the control object.
If there is a training sample of control precedents, i.e. a matrix of N precedents:

$$I = \{X_i Y_i\}, \quad i = 1,...N, \tag{6}$$

then it is advisable to use the precedent matrix (6), where X_i is the parameterized description of *the i* -th situation, and Y_i is the vector of correct control actions in *the i* -th situation. That is, to extrapolate the implementation of dependence (5) to the new situation X_{N+1} in order to obtain control Y_{N+1} , capable of at least approximately achieving the control goals without using the object model, etc.

Cause X and effect Y are measured in metric scales and are vectors

$$\begin{aligned} X &= [x_1,..., x_n], \\ Y &= [y_1,..., y_m], \end{aligned} \tag{7}$$

Defined in the corresponding Euclidean spaces of dimensions n and m : $X \in R^n$, $Y \in R^m$.
The equations of the elements of vector subspaces (hyperplanes) have the form

$$\begin{aligned} \{X'\} &= X_1 + \sum_{i=1}^{k-1} \lambda_i (X_{i+1} - X_i), \\ \{Y'\} &= Y_1 + \sum_{i=1}^{k-1} \mu_i (Y_{i+1} - Y_i). \end{aligned} \tag{8}$$

Then a function of the proximity of two control situations is chosen, for example a quadratic metric

$$\Phi\{X \; X'\} = \|X - X'\|^2. \tag{9}$$

On a two-dimensional plane, the subspaces of situations and solutions are described by the expressions:

$$\begin{aligned} \{X'\} &= X_1 + \lambda (X_2 - X_1), \\ \{Y'\} &= Y_1 + \mu (Y_2 - Y_1). \end{aligned} \tag{10}$$

The given form of the multidimensional extrapolation method allows to solve the interpolation problem quite reliably under conditions of information deficiency in low-dimensional spaces. With the growth of the space dimension, the computational difficulties grow, which are surmountable by the neural network architecture.

The nonlinearity and non-equidistance of interpolation nodes, as applied to a neural network system, under conditions of a finite training sample, causes the problem of ensuring a high probability of correct recognition of input samples that were not previously observed, i.e. geometrically parameterized outside the interpolation nodes.

At the same time, the multidimensionality and nonlinearity of production problems of mathematical interpolation, as well as the complexity of the topology of recognizable (as established above - synthesized) samples in the feature space - analogs does not allow the direct use of existing methods - both classical (such as the method of multidimensional linear extrapolation) and interpolation methods in the neural network computing basis. There is an important research task in the framework of machine learning technologies - the development of a scientific and methodological apparatus for the synthesis of problem-oriented architectures of neural networks and methods for their training, allowing the implementation of automatic synthesis of descriptions of new examples of designed systems and control actions of complex systems based on an input tuple of previously unobservable signals.

Examples of manifestations of the properties of prototype generation and generalization by similarity in various neural network architectures are already known, including dynamic NNs with feedback and self-organizing Kohonen maps. However, these effects have not been studied in detail when introducing structural redundancy in models.

3. CONCLUSION

This study has examined the fundamental relationship between mathematical interpolation and the generalization capabilities of neural networks within artificial intelligence systems. We have demonstrated that generalization, often treated as an elusive property of deep learning, can be systematically understood and engineered as a form of multidimensional nonlinear interpolation. This reframing is not merely a theoretical exercise. It carries profound implications for how we design, train, and deploy intelligent systems across a range of applied domains.

In practice, engineering and industrial decisions are rarely made from scratch. They build upon prior knowledge, analogies, and precedents, especially when dealing with novel systems, unknown operating conditions, or incomplete information. Our analysis shows that neural networks, when properly architected, can move beyond simple classification toward a more powerful form of synthesis. They can generate reasonable, context-aware responses to inputs they have never encountered before. This capability is essential for applications where retraining is impractical, data is inherently finite, and real-world variability is limitless.

What makes this approach particularly compelling is that it aligns with how human experts actually think. When an experienced engineer faces a new problem, they do not start from first principles every time. They draw on what they have seen before, they look for similarities, and they make educated guesses based on past experience. Neural interpolation does something remarkably similar. It creates a continuous mapping from known examples to unknown territory, effectively allowing the system to "fill in the gaps" in its knowledge. This is not just a technical convenience; it is a fundamental shift toward building AI that behaves more like a skilled practitioner and less like a rigid automaton.

The practical value of this approach extends across multiple disciplines. In industrial and manufacturing engineering, neural interpolation enables smarter adaptive control on the factory floor. It allows PID controllers and other regulatory systems to maintain performance under changing conditions without costly recalibration. In management information systems, it supports more robust forecasting, resource planning, and decision support by synthesizing reliable outputs from sparse or noisy data streams. In materials science and engineering, the methodology offers a pathway to reduce expensive physical testing by predicting material behavior in untested environments based on known precedents, thereby accelerating innovation while controlling costs. These are not speculative possibilities; they are tangible benefits that can be realized with existing technology, provided we are willing to rethink our approach to neural network design.

At the same time, we must acknowledge the limitations of our current understanding. While we have established a strong theoretical foundation for treating generalization as interpolation, there is still much work to be done in translating this insight into practical, scalable architectures. The challenge of high-dimensional spaces remains formidable, and the risk of overfitting in the presence of noise is ever present. However, these are not insurmountable obstacles. They are engineering problems, and like all engineering problems, they can be addressed through careful design, rigorous testing, and incremental improvement.

What we are ultimately advocating for is a shift in perspective. Rather than pursuing ever-larger datasets as the primary solution to generalization challenges, we should invest in neural network architectures that are intentionally designed to handle the unknown with grace. This involves leveraging structural redundancy, continuity, and nonlinearity not as accidental byproducts but as deliberate features that mirror the mathematical principles of interpolation. It also means embracing a more holistic view of the AI development process, one that considers not just the training data but the underlying structure of the problem being solved. The result is not just better-performing models, but more reliable, interpretable, and resource-efficient AI systems that can deliver tangible value in the high-stakes, data-constrained environments that characterize modern engineering and management practice.

Looking ahead, future research should focus on developing problem-oriented neural architectures that explicitly encode interpolation principles, along with training methodologies that preserve generalization under noise and high-dimensional conditions. There is also a need for more extensive empirical validation across different domains, from manufacturing to information systems to materials science, to confirm that the benefits we have identified are robust and repeatable. The intersection of mathematical rigor and applied intelligence holds great promise, and we hope this work contributes to bridging that gap in meaningful, practical ways. Ultimately, our goal is to see these ideas move from the research lab to the factory floor, from theoretical papers to real-world systems that make a difference in how we design, build, and control the technologies that shape our world.

References:

Aggarwal, C. (2020). *Neural networks and deep learning*. OOO Dialectika.

Bakhvalov, N. S., Zhidkov, N. P., & Kobelkov, G. M. (2020). *Numerical methods*. Laboratory of Knowledge.

Bishop, C. M. (2020). *Pattern recognition and machine learning*. OOO Dialectika.

Gasca, M., & Sauer, T. (2000). On the history of multivariate polynomial interpolation. *Journal of Computational and Applied Mathematics*, *122*(1), 23–35. https://doi.org/10.1016/s0377-0427(00)00353-8

Imomov, A. (2001). Explicit interpolation formulas for functions of several variables. In *Methods of spline functions: Abstracts of reports* (pp. 38–39). IM Publ.

Khaikin, S. (2006). *Neural networks: A complete course* (2nd ed.). Williams.

Konopatsky, E. V., Seleznev, I. V., Chernysheva, O. A., & others. (2021). Geometric modeling of adaptive algebraic curves passing through predetermined points. *Bulletin of Computer and Information Technologies*, *18*(9), 26–34.

Latorre Carmona, P., Sánchez, J. S., & Fred, A. (Eds.). (2013). *Mathematical methodologies in pattern recognition and machine learning* (Vol. 30). Springer.

Lekun, Y. (2021). *How a machine learns: Revolution in neural networks and deep learning*. Alpina PRO.

Luger, G. F. (2003). *Artificial intelligence: Strategies and methods for solving complex problems* (4th ed.). Williams.

Michelucci, U. (2020). *Applied deep learning: A case-based approach to understanding deep neural networks*. BHV-Petersburg.

Neural Network Zoo. (2024). *Data files and description*. Asimov Institute. https://www.asimovinstitute.org/neural-network-zoo

Romm, Ya. E., Dzhanunts, G. A., & Medvedkin, N. A. (2022). On the library of standard programs for calculating functions based on piecewise interpolation. *Modern Science-Intensive Technologies*, (11), 57–70.

Rosenblatt, F. (1965). *Principles of neurodynamics: Perceptrons and theory of brain mechanisms*. Mir.

Sadyraliev, Z., Zhumanbayeva, T. K., Dybina, O. V., Krutilin, A. A., & Amirkhanyan, G. M. (2025). Big data and artificial intelligence in information systems of product quality management. *Proceedings on Engineering Sciences, 7*(1), 119–128. https://doi.org/10.24874/PES07.01A.003

Shalev-Shwartz, S., & Ben-David, S. (2019). *Ideas of machine learning: From theory to algorithms* (A. A. Slinkina, Trans.). DMK Press.

Evgeny Andreevich Nekrasov
ITMO
Saint Petersburg
Russia
evgeniy@nekrasov.pw

Alexey Vladimirovich Degtyar
ITMO
Saint Petersburg
Russia
degtyar.avlad@gmail.com

Tatyana Anatolyevna Markina
ITMO
Saint Petersburg
Russia
markina_t@itmo.ru
ORCID 0009-0009-9146-433X

Ilya Borisovich Gosudarev
Faculty of Software Engineering and Computer Science
ITMO Saint Petersburg
Russia,
goss@itmo.ru

Nikolai Nikolaevich Zhukov
Faculty of Software Engineering and Computer Systems
ITMO Saint Petersburg
Russia,
nnzhukov@itmo.ru