

CONVOLUTIONAL NEURAL NETWORKS TRAINING FOR AUTONOMOUS ROBOTICS

Alexander LOZHKIN, Konstantin MAIOROV
Kalashnikov Izhevsk State Technical University, Izhevsk,

Pavol BOZEK
Slovak University of Technology, Trnava

Abstract:

The article discusses methods for accelerating the operation of convolutional neural networks for autonomous robotics learning. The analysis of the theoretical possibility of modifying the neural network learning mechanism is carried out. Classic semiotic analysis and the theory of neural networks is proposed to union. An assumption is made about the possibility of using the symmetry mechanism to accelerate the training of convolutional neural networks. A multilayer neural network to represent how space is an attempt has been made. The conclusion was based on the laws on the plane obtained earlier. The derivation of formulas turned out to be impossible due to the problems of modern mathematics. A new approach is proposed, which involves combining the gradient descent algorithm and the stochastic completion of convolutional filters by the principles of symmetries. The identified algorithms allow increasing the learning rate from 5% to 15%, depending on the problem that the neural network solves.

Key words: autonomous robots, convolutional neural networks, learning, automorphisms, symmetry mechanism, semiotic analyze

INTRODUCTION

Modern autonomous robots cannot be imagined without artificial intelligence systems, namely neural networks. The robot must solve strictly defined tasks, therefore the complexity of the neural network must be very large and the learning time corresponding to the complexity of the process. The solution for this problem is multilayer convolutional neural networks. Such and similar tasks have been addressed by the authors in their scientific works [1, 2, 3, 4, 5]. The training time for such a network can be very long. Convolutional neural networks are currently the main algorithm for tasks such as image classification and object recognition. Since the 2010s [6, 7, 8], neural network architectures have become overgrown with more and more layers and more and more time has been required to train them. Despite the fact that there are already a lot of methods for accelerating learning, the task is still relevant.

The five main groups of methods for accelerating the training of convolutional networks exist.

1. Tensor decompositions of weights and activations of convolutional layers
2. Quantization or binarization of weights
3. Converting a large network into a small one by thinning weights.
4. Adaptive neural network architectures.
5. Training a small network using a large network.

Let us take a closer look at each method.

LITERATURE REVIEW

Modern principles of training neural networks

1. Tensor decompositions

The standard expression for convolution (see Figure 1) can be represented:

$$O(x, y, k) = \sum_{i=0}^d \sum_{j=1}^d \sum_{c=1}^C W(i, j, c, k) I(x + i, y + i, c) \quad (1)$$

where:

I – three-dimensional tensor of input data,

O – three-dimensional tensor of output,

W – four-dimensional tensor of weights [9, 10, 11] (filter in the context of convolutional networks).

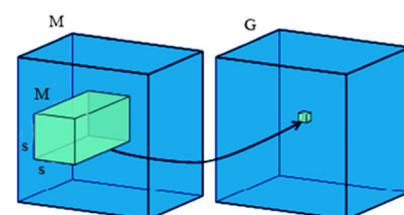


Fig. 1 Full convolution

Tensor expansion consists in approximating the tensor W using two-dimensional tensors. The expression for the canonical decomposition of rank R will look like this:

$$W = \sum_{r=1}^R H_r^x(i) H_r^y(j) H_r^c(c) H_r^k(k) \quad (2)$$

where:

R rank is a tunable parameter that controls the trade-off between accuracy preservation and decomposition efficiency.

The decomposition is shown in Fig. 2.

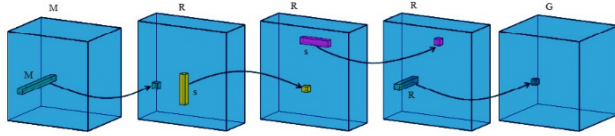


Fig. 2 Canonical decomposition

2. Quantization or binarization of weights

The main idea behind these approaches is to compress weights to increase the speed of work and reduce memory consumption.

In the case of scalar quantization, all elements of the weight matrix W are clustered by the k-means method, after which the centroids of the obtained clusters will become the new values of the weights.

$$\widehat{W}_{ij} = c_t \quad (3)$$

where:

$$t = \underset{z}{\operatorname{argmin}} |W_{ij} - c_z|$$

Large weight matrices W , must be to split into submatrices W_1, W_2, \dots, W_n from the beginning and then clustering inside each submatrix independently [12].

The process of training a neural network with compressed weights is:

- 1) Obtained the compressed values of the weights by applying the quantifying procedure q :

$$\widehat{W} = q(W) \quad (4)$$

- 2) Made prognosis with compressed weights and calculate the error function. We keep all neuron activations in their original precision.

- 3) Backpropagate error gradients with compressed weights and initial activation values. Update the original weights.

3. Converting a large network into a small one by thinning weights

Weights decimation is the most common way to transform a large network into a small one. The importance of each neuron is calculated according to the selected criterion in a pre-trained network. Then the least important neurons are clipped and the network is retrained, minimizing drops in accuracy.

4. Adaptive neural network architectures

The learning rate can be increased by using neural networks with stochastic depth [13]. The network has a variable number of layers in the learning process, when at the testing stage everything is fully used. The acceleration will be especially noticeable for very deep models.

The responsive framework complements the capabilities of the ResNet architecture. Each k -th residual block is determined by the formula:

$$H_k = \operatorname{Re} \cdot \operatorname{lu}(b_k f_k(Z_{k-1}) + id(Z_{k-1})) \quad (5)$$

where:

b_k is a Bernoulli random variable taking the value 0 or 1.

The architecture of an adaptive network based on ResNet is presented in Figure 3, where the probabilities of that decrease linearly with each new layer of the network.

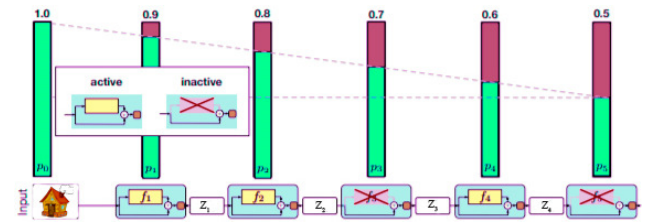


Fig. 3 Adaptive network

5. Training a small network using a large network

The principle of this approach is to train a small neural network with fewer parameters on the forecasts of a large full-fledged network. To train the student neural network, a vector with a posteriori probability distribution of the output layer of the teacher network is used. Such a vector conveys much more information about the sample data than a deterministic forecast [14, 15, 16].

Let z_i be the prediction of the neural network-teacher, then the probabilistic response p_i will be calculated using the softmax formula:

$$p_i = \frac{e^{\frac{z_i}{\tau}}}{\sum_j e^{\frac{z_j}{\tau}}} \quad (6)$$

where:

τ is a tunable parameter that, at high values, acts as a regularizer of the student model, and at low values, it provides a more complete transfer of information.

The student model is trained to simultaneously predict the true y values and the teacher model response from the following loss function:

$$L = E(p^S, p^Y) + \lambda E(p^S, p^T) \quad (7)$$

where:

E is the cross entropy and p^Y is the one-hot vector of true values.

Also, a similar approach can be used to train the hidden layers of the learner network.

Most of the considered approaches allow the neural network to obtain a forecast faster, but additionally increase the time required for its training. The problem of accelerating the learning stage remains relevant.

METHODOLOGY OF RESEARCH

Semiotic analyze Of neural networks

Modern physiologists believe that a person makes decisions long before mental activity [17]. A piece of DNA was identified that transmits information about orientation in space without the participation of vision in the course of biochemical studies. Based on this, it can already be concluded that some mechanisms of spatial orientation can work without visual contact with the environment. We can assume that such a mechanism is the automorphisms (symmetries) of the space [18].

The basis of the approach for analyzing the correctness of solving various problems is pragmatic analysis (the stage of

semiotic analysis). This approach was first proposed by Ershov. Ershov showed all the commands of the processor and actually proved the possibility of describing any object or process by the linguistic method. The sixth technological order is possible due to this.

Let us connect the linguistic method with geometry on the basis of the postulate that all objects and all processes are not representable outside of space. The theoretical foundations of neural networks were formulated by Kholmogorov and Arnold on the plane in the same way.

The authors of the article do not deny the presence of mental activity in living nature, therefore, let us turn to the main line of research. The mathematical basis of neural networks is the solution to the Hilbert thirteenth problem [19]. Arnold chooses a strict order of proof and invites followers to choose the one they need. The curve is divided into very short sections determined by inequality (8). The curve tree is built additionally.

$$|f_h^n(x_k) - f_h^{n_r}(x_k)| \leq (3 + \frac{n-n_r}{n_{r+1}-n_r}) \frac{1}{r^2} \quad (8)$$

where:

$n_r < n \leq n_{r+1}$ (order on the tree).

Kholmogorov and Arnold investigated functions of three variables. Analytical formulas for the transformation of functions in two variables did not exist at that time. These formulas are obtained by special functions now. Functions represented by a system of parametric equations can be transformed and obtained an analytical formula. The solution method is based on preserving the fundamental property of space, namely, the symmetry of the permutation [20, 21, 22].

RESULTS OF RESEARCH

Splitting the curve into sections destroys the semantics of the process. This allows you to build a strict learning curve for a limited class of problems. Any modern neural network has several layers. Each layer can be viewed as a certain plane in three-dimensional space. A set of parallel planes has no space properties. The problem must be solved in space directly. Solving the problem can improve all the parameters of the neural network, and most importantly, the creator will understand the decision-making process for any input signals.

Let us have space curve by system parametric equations:

$$\begin{cases} x = f_x(t) \\ y = f_y(t), \\ z = f_z(t) \end{cases}$$

where:

$t \in [\tau_1 \tau_2]$.

Let the sequence of input signals be represented by a matrix:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

Permutation is one of the simpler symmetries. It imposes strict restrictions on the dimension of space. The dimension of space must be even, as in Hamiltonian mechanics. The researcher must solve the problem in four-dimensional space-time then system 9 takes the form:

$$\begin{cases} x = f_x(t) \\ y = f_y(t) \\ z = f_z(t) \\ v = f_v(t) \end{cases}$$

The solution of the parametric system is reduced by the symmetry of the permutation to:

$$\begin{cases} k_x f_x(t + \alpha_1) = a_{11} f_x(t + \beta_1) + a_{12} f_y(t + \beta_1) + \\ a_{13} f_z(t + \beta_2) + a_{14} f_v(t + \beta_2) \\ k_y f_y(t + \alpha_1) = a_{21} f_x(t + \beta_1) + a_{22} f_y(t + \beta_1) + \\ a_{23} f_z(t + \beta_2) + a_{24} f_v(t + \beta_2) \\ k_z f_z(t + \alpha_2) = a_{31} f_x(t + \beta_1) + a_{32} f_y(t + \beta_1) + \\ a_{33} f_z(t + \beta_2) + a_{34} f_v(t + \beta_2) \\ k_v f_v(t + \alpha_2) = a_{41} f_x(t + \beta_1) + a_{42} f_y(t + \beta_1) + \\ a_{43} f_z(t + \beta_2) + a_{44} f_v(t + \beta_2) \end{cases} \quad (9)$$

where:

$\alpha_1, \alpha_2, \beta_1, \beta_2$ angles of vectors of non-orthogonal basis in planes xOy and zOv .

Angles α are proper angles, and corners β are symmetry angles of the permutation. Formula 9 unfolds into the system:

$$\begin{cases} k_x f_x(t) \cos(\alpha_1) - k_y f_y(t) \sin(\alpha_1) = \\ a_{11}(f_x(t) \cos(\beta_1) - f_y(t) \sin(\beta_1)) + \\ a_{12}(f_x(t) \cos(\beta_1) + f_y(t) \sin(\beta_1) + \\ a_{13}(f_z(t) \cos(\beta_2) - f_v(t) \sin(\beta_2)) + \\ a_{14}(f_z(t) \cos(\beta_2) + f_v(t) \sin(\beta_2)) \\ k_x f_x(t) \cos(\alpha_1) + k_y f_y(t) \sin(\alpha_1) = \\ a_{21}(f_x(t) \cos(\beta_1) - f_y(t) \sin(\beta_1)) + \\ a_{22}(f_x(t) \cos(\beta_1) + f_y(t) \sin(\beta_1) + \\ a_{23}(f_z(t) \cos(\beta_2) - f_v(t) \sin(\beta_2)) + \\ a_{24}(f_z(t) \cos(\beta_2) + f_v(t) \sin(\beta_2)) \\ k_z f_z(t) \cos(\alpha_2) - k_v f_v(t) \sin(\alpha_2) = \\ a_{31}(f_x(t) \cos(\beta_1) - f_y(t) \sin(\beta_1)) + \\ a_{32}(f_x(t) \cos(\beta_1) + f_y(t) \sin(\beta_1) + \\ a_{33}(f_z(t) \cos(\beta_2) - f_v(t) \sin(\beta_2)) + \\ a_{34}(f_z(t) \cos(\beta_2) + f_v(t) \sin(\beta_2)) \\ k_z f_z(t) \cos(\alpha_2) + k_v f_v(t) \sin(\alpha_2) = \\ a_{41}(f_x(t) \cos(\beta_1) - f_y(t) \sin(\beta_1)) + \\ a_{42}(f_x(t) \cos(\beta_1) + f_y(t) \sin(\beta_1) + \\ a_{43}(f_z(t) \cos(\beta_2) - f_v(t) \sin(\beta_2)) + \\ a_{44}(f_z(t) \cos(\beta_2) + f_v(t) \sin(\beta_2)) \end{cases}$$

The concept was introduced by analogy with the Euclidean plane. The transformation matrix for 3D space will be:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and system 9 is:

$$\begin{cases} k_x f_x(t + \alpha_1) = a_{11} f_x(t + \beta_1) + a_{12} f_y(t + \beta_1) + \\ a_{13} f_z(t + \beta_2) \\ k_y f_y(t + \alpha_1) = a_{21} f_x(t + \beta_1) + a_{22} f_y(t + \beta_1) + \\ a_{23} f_z(t + \beta_2) \\ k_z f_z(t + \alpha_2) = a_{31} f_x(t + \beta_1) + a_{32} f_y(t + \beta_1) + \\ a_{33} f_z(t + \beta_2) \\ k_v f_v(t + \alpha_2) = f_z(t) \cos(\beta_2) + \sin(\beta_2) \end{cases}$$

The system of proportions is cumbersome to publish, so we give a generalized formula. Considering the basis of space as a union of two bases of perpendicular planes leads to an analytical solution of a system of two proportions, which contain a linear combination of trigonometric functions.

Formula is:

$$\begin{cases} \frac{\sum b_{ij}\phi(\beta_k)}{\sum d_{ij}\phi(\beta_k)} = \frac{\sum c_{ij}\phi(\beta_k)}{\sum e_{ij}\phi(\beta_k)} \\ \frac{\sum m_{ij}\phi(\beta_k)}{\sum h_{ij}\phi(\beta_k)} = \frac{\sum g_{ij}\phi(\beta_k)}{\sum l_{ij}\phi(\beta_k)} \end{cases}$$

where:

$$\sum b_{ij}\phi(\beta_k) \neq \sum d_{ij}\phi(\beta_k) \neq \sum c_{ij}\phi(\beta_k) \neq \sum e_{ij}\phi(\beta_k) \neq \sum m_{ij}\phi(\beta_k) \neq \sum h_{ij}\phi(\beta_k) \neq \sum g_{ij}\phi(\beta_k) \neq \sum l_{ij}\phi(\beta_k)$$

$\varphi \in \{\sin, \cos\}$,

$\beta_1 \beta_2$ – permutation symmetry angles in each basis of perpendicular defining planes

$bdcemhgl$ – coefficients.

Unfortunately, this system does not have an analytical solution at the present time. Therefore, a complete replacement of a neural network with a set of functionals is not yet possible. Let us try to apply some principles of symmetry to neural networks.

DISCUSSION

Simple modification for neural systems

The idea behind a new approach to speed up the training of convolutional networks is to use this symmetry mechanism for faster network convergence. The approach consists of two stages: symmetric initialization of convolutional filters and stochastic symmetric filtering.

Symmetric filter initialization takes place in several steps:

1. Initialize only half of the desired number of filters in the standard way [22, 23].
2. Perform forward and backward propagation in one mini-batch.
3. Initialize the second half of the filters by performing symmetric transformations on the first half.

Stochastic symmetric filtering is used in conjunction with the usual gradient algorithms.

1. At each k -th mini-batch, build up the filters in turn, making symmetric transformations and calculating the network error.
2. If the network error has decreased, then apply the transformation data, otherwise use the standard gradient descent step.

This approach will accelerate the convergence of a convolutional neural network, express the role of symmetries and automorphisms in neural networks.

CONCLUSIONS

The concept of a neural network originated from Arnold's mathematical works. They are devoted to breaking a complex curve into short segments. An earlier [24, 25] the authors of the article obtained transformation results for most of the curves used in engineering calculations on a plane. Thus, it became possible to create a single-layer neural network with decision analysis. In other words, you can reveal the decision-making process explicitly or move from chaos to harmony. The theoretical foundation is symmetry, as a fundamental property of space. The results of such a classical branch of AI theory as semiotic analysis were used additionally.

A multilayer neural network can be thought of as a set of parallel planes in space. Therefore, the next task was to find the exact parameters of the spatial curve. The solution to

this problem would make it possible to find an explicit algorithm for neural networks of any complexity and train the network in the shortest possible time. Unfortunately, the derivation of formulas ran into a dead end associated with some problems of modern mathematics [20, 25]. The original systems of parametric equations are given.

Symmetries are a property of both space and text in any language. They are named universals in linguistics. Algorithms for applying symmetries for existing convolutional networks are identified. They allow increasing the learning rate from 5% to 15%, depending on the problem that the neural network solves.

The robot in the space now has the following problems: the uncertainty of the trajectory of movement and errors in the operation of the neural network due to the opaque algorithm for obtaining the solution [26, 27, 28]. Both of these problems can be resolved using the method described in this article. The authors may have made mistakes in their reasoning, so they invite a wide range of researchers to obtain the desired result.

ACKNOWLEDGMENTS

This publication has been written thanks to support of the research project VEGA 1/0019/20 and project KEGA 013TUKE-4/2019 "Modern educational tools and methods for shaping creativity and increasing the practical skills and habits of graduates of technical departments of universities".

REFERENCES

- [1] P. Bozek, Z. Ivandic and others. "Solutions to the characteristic equation for industrial robot's elliptic trajectories". *Tehnicki Vjestnik – Technical Gazette*, vol. 23, pp. 1017-1023, 2016.
- [2] A. Kilin, P. Bozek, and others. "Experimental investigations of a highly maneuverable mobile omnivheel robot". *International Journal of Advanced Robotic Systems*. Vol. 14, iss. 6 (2017), pp. 1-9.
- [3] R. Pirnik, M. Hrubaš and others. "Integration of inertial sensor data into control of the mobile platform". in *Advances in Intelligent and Soft Computing, SDOT 2015*, vol. 511, pp. 271-282.
- [4] T. Dodok, N. Cubonova and others. "Utilization of strategies to generate and optimize machining sequences in CAD/CAM". 12th International Scientific Conference of Young Scientists on Sustainable, Modern and Safe Transport. *Procedia Engineering*. Volume: 192, pp. 113-118.
- [5] M. Saga, M. Vasko and others. Chosen numerical algorithms for interval finite element analysis. Modelling of Mechanical and Mechatronic Systems. *Procedia*
- [6] J. Peterka, P. Pokorny and S. Vaclav. CAM strategies and surface accuracy. *Annals of DAAAM and Proceedings*. 2008, pp. 1061-1062.
- [7] M. Beno, M. Zvoncan and others. Circular interpolation and positioning accuracy deviation measurement on five axis machine tools with different structures. *Tehnicki Vjestnik – Technical Gazette*. 2013, 20, 3, pp. 479-484.
- [8] A. Nemethova, D. Borkin and G. Michalconok. Comparison of Methods for Time Series Data Analysis for Further Use of Machine Learning Algorithms. In *Proceedings of the Computational Methods in Systems and Software*. Springer, Cham, 2019. pp. 90-99.

- [9] M. Nemeth, A. Nemethova and G. Michalconok. Determination issues of data mining process of failures in the production systems. Book Series: *Advance in Intelligent Systems and Computing*. 2019. Vol 985, pp. 200-207.
- [10] A. Nemethova, M. Nemeth and others. Identification of KDD problems from medical data. Series: *Advance in Intelligent Systems and Computing*. 2019. Vol 985, pp. 191-199.
- [11] P. Anderson, P. Culley and T.J. Parker. *Marketing Research*. London: Hansen Publisher, 2003.
- [12] D.G. Smith and R.G. Rhodes. "Specification Formulation". *Journal of Engineering*, December 2001, Vol. 2, No. 2. pp. 223-228.
- [13] V.I. Arnol'd. "On functions of three variables". *Amer. Math. Soc. Transl.* (2) 28, 1963, pp. 51-54.
- [14] R. Rigamonti, A. Sironi, and others. "Learning separable filters". In Conference on Computer Vision and Pattern Recognition (CVPR), 2013.
- [15] T. Krenicky. "Implementation of Virtual Instrumentation for Machinery Monitoring". *Scientific Papers: Operation and Diagnostics of Machines and Production Systems Operational States: Vol. 4*. RAM-Verlag, Lüdenscheid, 2011, pp. 5-8. ISBN 978-3-942303-10-1.
- [16] Z. Murcinkova and T. Krenicky. "Implementation of virtual instrumentation for multiparametric technical system monitoring". *SGEM 2013: 13th Int. Multidisciplinary Sci. Geocnf. Vol. 1*. 16-22 June, 2013, Albena, Bulgaria. Sofia: STEF92 Technology, 2013. pp. 139-144. ISBN 978-954-91818-9-0.
- [17] S. Anwar, K. Hwang, and W. Sung. "Fixed point optimization of deep convolutional neural networks for object recognition". In *Acoustics, Speech, and Signal Processing (ICASSP)*, International Conference on, 2015.
- [18] G. Huang, Y. Sun and others. "Deep networks with stochastic depth". in European Conference on Computer Vision (ECCV), 2016.
- [19] V.I. Arnol'd. "On functions of three variables." In *Amer. Math. Soc. Transl.* 1963, Vol. 28, No.2. pp. 51-54.
- [20] G.E. Hinton, O. Vinyals and J. Dean. "Distilling the knowledge in a neural network". NIPS Deep Learning Workshop, 2014.
- [21] P. Bozek, A. Lozhkin, and others. "Information technology and pragmatic analysis". *Computing and informatics*. 2018. Vol. 37, Issue 4, C, pp. 1011-1036
- [22] A. Lozhkin, A. Korobeynikov and R. Khaziyakhmetov "The Newton problem solution of the transformed complex curve parameters". In *Journal of Physics: Conference Series*, 2019, Vol. 1399, Applied Physics, doi:10.1088/1742-6596/1399/2/022004.
- [23] Y. Goodfellow, A. Bengio and A. Courville. "Deep Learning". The MIT Press, 2016, pp. 84-91
- [24] P. Bozek and E. Pivarciova. "Registration of Holographic Images Based on Integral Transformation". *Computing and Informatics*. Vol. 31, No. 6 (2012), pp. 1369-1383.
- [25] P. Bozek and G. Chmelikova. "Virtual Technology Utilization in Teaching". *ICL 2011, 14th International Conference on Interactive Collaborative Learning and 11th International Conference Virtual University*. Piscataway: IEEE, 2011, pp. 409-413.
- [26] J. Zhao. "Exponential stabilization of memristor-based neural networks with unbounded time-varying delays". *Science China Information Sciences*, Volume 64, Issue 8, 1 August 2021,
- [27] S. Ivanovna. "Neural Network Modeling of Productive Intellectual Activity in Older Adolescents". *Advances in Intelligent Systems and Computing*, Volume 1250, IntelliSys 2020, London; United Kingdom; pp. 399-406.
- [28] Z. Murcinkova and T. Krenicky. "Applications utilizing the damping of composite microstructures for mechanisms of production machines and manipulator devices". *SGEM 2013: 13th Int. Multidisciplinary Sci. Geocnf. Vol. 1*. 16-22 June, 2013, Albena, Bulgaria. Sofia: STEF92 Technology, 2013. pp. 23-30. ISBN 978-954-91818-9-0.

Alexander Lozhkin

ORCID ID: 0000-0001-9048-2469
 Kalashnikov Izhevsk State Technical University
 Institute of Informatics and Hardware
 Software department
 Student Street, 7, Izhevsk, Russia
 e-mail: lag.izh@gmail.com

Pavol Bozek

ORCID ID: 0000-0002-3891-3847
 Slovak University of Technology
 Faculty of Materials Science and Technology
 Institute of Production Technologies
 J. Bottu 25, 917 24 Trnava, Slovak Republic
 e-mail: pavol.bozek@stuba.sk

Konstantin Maiorov

ORCID ID: 0000-0003-4285-7697
 Kalashnikov Izhevsk State Technical University
 Institute of Informatics and Hardware
 Software department
 Student Street, 7, Izhevsk, Russia