Development of Intelligent Information Technology of Computer Processing of Pedagogical Tests Open Tasks Based on Machine Learning Approach

Arina Herasymova[1], Dmytro Chumachenko[1], Halyna Padalko[2]

[1] National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine
[2] Lutsk National Technical University, Lutsk, Ukraine
dichumachenko@gmail.com

Abstract. Recently, such teaching methods as distance learning, e-learning and self-study have been actively developed. The widespread adoption of the Internet in all aspects of information technology has also affected the organization of the learning process. Now computer courses are very popular, which are electronic textbooks and provide for independent study. They are being replaced by distance and online learning.
Concerning, this paper presents the current development of methods for computer processing of open tasks to conduct an effective analysis of the learning of educational material.
Purpose of work was to analyze the methods and approaches used for computer processing of natural language, as well as tools to solve the problem of identifying test messages. The subject of research was mathematical models and methods of computer processing of natural language.

Keywords: Machine learning, Supervised learning, Artificial neural network, Word embedding, Linear classifier, Gradient descent, Softmax, Maximum likelihood estimation.

1 Relevance and Description of the Problem of Processing Open Tasks of Pedagogical Tests

1.1 Relevance of Work

The world pandemic of the new coronavirus has changed the usual way of life and approaches to education [1]. Over the past few months, the number of students affected by the closure of schools and universities in 138 countries has nearly quadrupled to reach 1.37 billion. This means that over 3 out of 4 children and young people around the world are not able to attend educational institutions. The closure of educational institutions also affected nearly 60.2 million teachers [2].

However, the pandemic only accelerated the process of digitalization of the educational process, and trends in the transition of training to online have been observed for several years [3-4]. And now trends of digitalization not only in organization of educational process, but in Public Health [5-6], medical diagnostics [7-8], security [9-10], retail [11], financial branch [12] and other spheres of human being are developed with high speed.

One of the integral parts of the existence of mankind is the creation, transfer of accumulated knowledge and skills, that is, the transfer of information. Recently, such teaching methods as distance learning, e-learning and self-study have been actively developed [13-14]. The widespread adoption of the Internet in all aspects of information technology has also affected the organization of the learning process and effective human resource management [15-17].

Now computer courses are very popular, they represent electronic textbooks and provide independent study, but they are being replaced by distance and online learning [18]. In the process of education, the interaction between the teacher and the student is very important, the teacher needs to understand how well the student understands, assimilates the material [19].

Traditional methods of remote control of knowledge use closed types of test tasks, based on the usual calculation of the correct answers, and this does not always correctly reflect the student's knowledge [20]. In this regard, this paper presents the current development of methods for computer processing of open tasks to conduct an effective analysis of the learning of educational material.

1.2 Aims and Objectives

The aim of the research is to analyze the methods and approaches used for computer processing of natural language, as well as tools to solve the problem of identifying test messages, and to build intelligent information technology for open tasks processing.

For tasks of this type, machine learning approach is usually used [21]. Artificial neural network (ANN) is a mathematical model, as well as its software or hardware implementation, built on the principle of organization and functioning of biological neural networks - networks of nerve cells of a living organism [22].

Learning ability is one of the main advantages of neural networks over traditional algorithms. Technically, training consists in finding the coefficients of connections between neurons. In the learning process, the neural network is able to identify complex relationships between input and output, as well as perform generalization. This means that if training is successful, the network will be able to return the correct result based on data that were not in the training sample, as well as incomplete and / or "noisy", partially distorted data.

Artificial neural networks are used to solve various classes of problems, such as: decision making and management, clustering, forecasting, approximation, data compression, data analysis, optimization [23]. In order for them to form a result from raw data, artificial networks must go through the following stages of solving problems:

- data collection for training;
- preparation and normalization of data;
- experimental selection of training parameters;
- own training;

– verification of the adequacy of training;
– parameter adjustments, final training.

2 Natural Language Computer Processing Methods

2.1 Vector Representations of Words.

To teach word selection without pre-marked data, we first need to solve several problems [24]:

– create data tuples in the format [input word, source word], each word is represented as a lifelong vector of length n, where the i value is encoded by one at the i position and zeros at all the others;
– create a model that receives one-hot vectors on input and output;
– determine the loss function; predicts the right word to optimize the model;
– determine the quality of the model, making sure that similar words have similar vector representations.

Take this example: The cat pushed the glass off the table. The data we need will come out as in figure 1. Each bracket denotes a single context window. The blue field indicates the input one-hot vector (target word), the red field indicates the output one-hot vector (any word in the context window except for the target word, the so-called context word). Two data elements come out from one context window (there are two neighboring ones per target word).

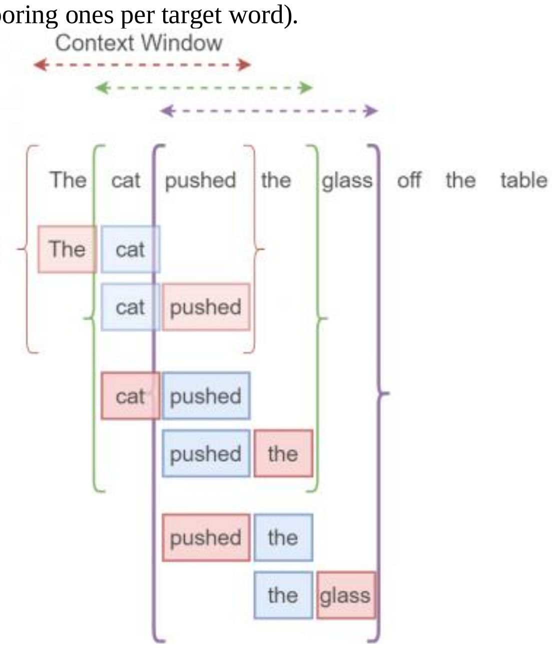

Fig. 1. Context window.

Embedding layer stores the vectors of all words in the dictionary, where the number of words x is the dimension of the space of the compressed vector representation of words [25]. This embedding size is a custom parameter. The larger it is, the better the model (but after reaching a certain embedding size you will not get a big performance boost). This giant matrix is initialized randomly (like a neural network) and is configured bit by bit during the optimization process (fig. 2).

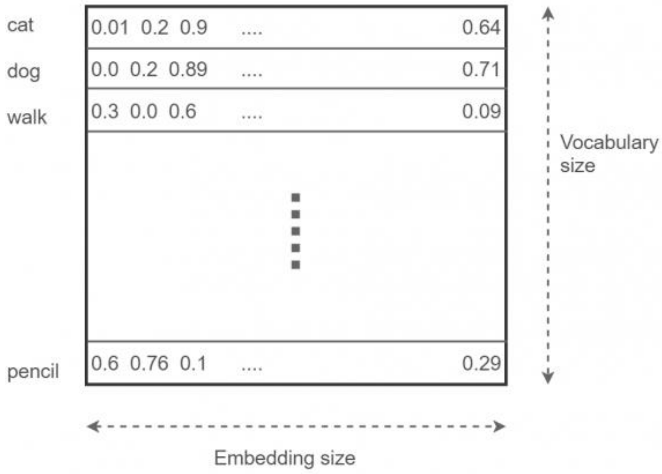

Fig. 2. Embedding layer algorithm diagram

After training the model, we can only save the embedding layer on the disk, after which we can use vectors with saved semantics at any time. The full algorithm looks like in figure 3.

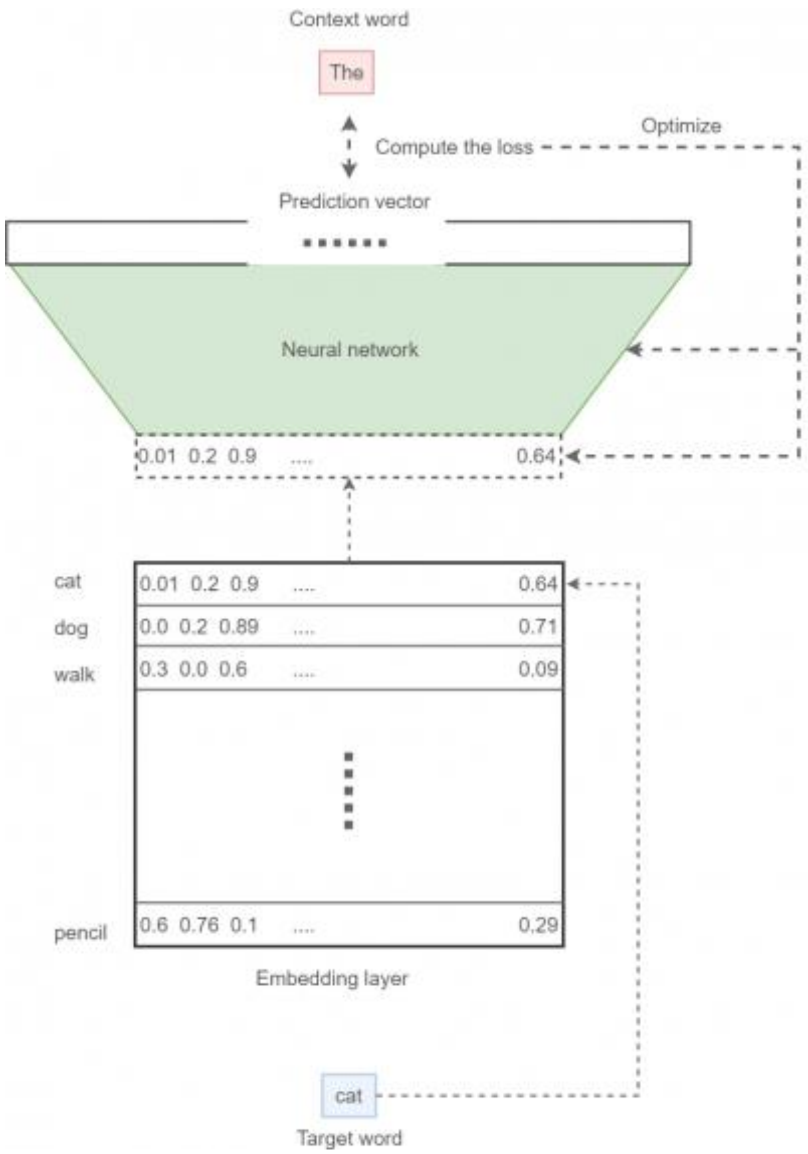

Fig. 3. Complete algorithm

2.2 Linear Classifier

Linear classifier is a way to solve classification problems, when a decision is made on the basis of a linear operator on the input data [26]. The class of problems that can be solved using linear classifiers, respectively, have the property of linear separability.

Let the vector $\vec{x}$ from real numbers be input, and at the output of the classifier, the exponent Y is calculated by the formula:

$$Y = f(\vec{w} \cdot \vec{x} + \vec{b}) = f(\sum_j w_j x_j + b_j), \quad (1)$$

where $\vec{w}$ is the real vector of weights, $\vec{b}$ is regularization coefficient, which does not allow parameters to go beyond reasonable limits (through retraining), f is the scalar product transformation function.

The values of the weights of the vector $\vec{w}$ are determined during machine learning on prepared samples. The function f is usually a simple threshold function that separates one class of objects from another. In more complex cases, the function f has the meaning of the probability of a solution.

The linear classification operation for two classes can be imagined as the reflection of objects in multidimensional space onto a hyperplane, in which those objects that fall on one side divides the lines, belong to the first class («yes»), and the objects on the other side - to the second class («no»).

Linear classifier is used when it is important to carry out fast calculations with high speed. It works well when the input vector x ⃗ is sparse. Linear classifiers can work well in multidimensional space, for example, to classify documents according to the word-birth matrix. In such cases, objects are considered to be well regularized.

2.3 Gradient Descent

Let the objective function have the form:

$$F(\vec{x}) : \mathbb{X} \to \mathbb{R} \quad (2)$$

And the optimization task is defined as follows:

$$F(\vec{x}) : \min_{\vec{x} \epsilon \mathbb{X}} f(x) \quad (3)$$

In the case when you need to find the maximum, instead of $F(\vec{x})$, $-F(\vec{x})$ is used. The main idea of the method is to go in the direction of the steepest descent, and this direction is given by the anti-gradient - ∇F:

$$\vec{x}^{[j+1]} = \vec{x}^{[j]} - \vec{\lambda}^{[j]} \nabla F(\vec{x}^{[j]}) \quad (4)$$

where λ ⃗ ^ [j] sets the gradient descent speed and can be selected:

- constant (in this case, the method may diverge);
- descending in the process of gradient descent;
- guarantees quick descent.

To find the minimum of $F(\vec{x})$ we get:

$$\lambda^{[j]} = argmin_{\lambda} F(\vec{x}^{[j+1]}) = argmin_{\lambda} F(\vec{x}^{[j]} - \lambda \nabla F(\vec{x}^{[j]})) \quad (5)$$

To find the maximum of $F(\vec{x})$ we get:

$$\lambda^{[j]} = argmax_{\lambda} F(\vec{x}^{[j+1]}) = argmax_{\lambda} F(\vec{x}^{[j]} + \lambda \nabla F(\vec{x}^{[j]})) \quad (6)$$

Gradient Descent Algorithm:

1. set the initial approximation and calculation accuracy $\vec{x}^0, \varepsilon$;
2. calculate $\vec{x}^{[j+1]}$ according to the formula 4 and i $\lambda^{[j]}$ according to the formula 5.
3. check the stop condition:

if

$$\left|\vec{x}^{[j+1]} - \vec{x}^{[j]}\right| > \varepsilon, \quad \left|F(\vec{x}^{[j+1]}) - F(\vec{x}^{[j]})\right| > \varepsilon \quad (7)$$

then j=j+1going to step 2;

otherwise $\vec{x} = \vec{x}^{[j+1]}$ is a solution.

2.4 Softmax

Softmax [27] is a generalization of the logistic function for the multidimensional case. The function converts a vector z of dimension K into a vector σ of the same dimension, where each coordinate σ_i of the resulting vector is represented by a real number in the interval [0,1] and the sum of the coordinates is 1.

The coordinates σ_i are calculated as follows:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \tag{8}$$

The Softmax function is used in machine learning for classification problems when the number of possible classes is more than two (for two classes a logistic function is used). The coordinates σ_i of the resulting vector are interpreted as the probabilities that the object belongs to class i. The column vector z is calculated as follows:

$$z = w^T x - \theta \tag{9}$$

where x is the column vector of features of an object of dimension $M \times 1$, w^T is transposed matrix of weighting coefficients of features, has dimension $K \times M$, θ is a column vector with limit values of dimension $M \times 1$, K is the number of classes of objects, M is the number of features of the objects.

Typically, Softmax is used for the last layer of deep neural networks for classification tasks. To train the neural network, cross entropy is used as an option for losses.

2.5 Maximum Likelihood Method

The maximum likelihood method (also known as highest probability method) [28-29] in mathematical statistics is a method for estimating an unknown parameter by maximizing the likelihood function. It is based on the assumption that all information about the statistical sample is contained in this function. Maximum likelihood assessment is a popular statistical method that is used to create a statistical model based on data and provide an estimate of model parameters.

Let us have a sample $X_1, \dots, X_n$ according to the distribution P_θ, where $\theta \in \Theta$ is an unknown parameter. Let $f(x|\theta) : \Theta \to \mathbb{R}$ be the likelihood function, where $x \in \mathbb{R}$. Point estimate - called the maximum likelihood estimate of the parameter θ.

$$\hat{\theta}_{\text{МП}} = \hat{\theta}_{\text{МП}}(X_1, \dots, X_n) = \arg\max_{\theta \in \Theta} f(X_1, \dots, X_n|\theta) \tag{10}$$

Thus, the maximum likelihood estimate is an estimate that maximizes the likelihood function for a fixed sample implementation.

3 Implementation and Results

3.1 Development with TensorFlow Library

TensorFlow [30] is an open machine learning software library developed by Google to solve the problems of building and training a neural network to automatically detect and classify images, achieving the quality of human perception.

The library allows developers to quickly and easily get started with deep learning in the cloud. The platform has wide industry support and has become a common research solution for deep learning and application development, especially in areas such as machine vision, natural language understanding, and speech translation.

TensorFlow comes with a complete set of visualization tools that simplify understanding, tuning and optimizing applications. Thanks to the support of various styles (from images and audio to histograms and graphs) you can quickly and easily create large deep neural networks.

3.2 Result of Development

While the system is learning, we need to check how well the network works in terms of accuracy. We do this with a test suite, taken from testing data, so it does not have duplication with training data.

Using validation sets based on testing data allows you to better understand how well the network can generalize what it is studying and apply it to other contexts. If we check the training data, the network may be oversaturated - in other words, finding out specific examples and remembering the answers to them does not help the network answer new questions.

After a little workout, let's take a look inside and see what answers we get from the network. In the diagrams below, we visualize the attention for each of the episodes (rows) for all sentences (columns) in our context. Darker colors pay more attention to this sentence on this episode. (fig. 4).

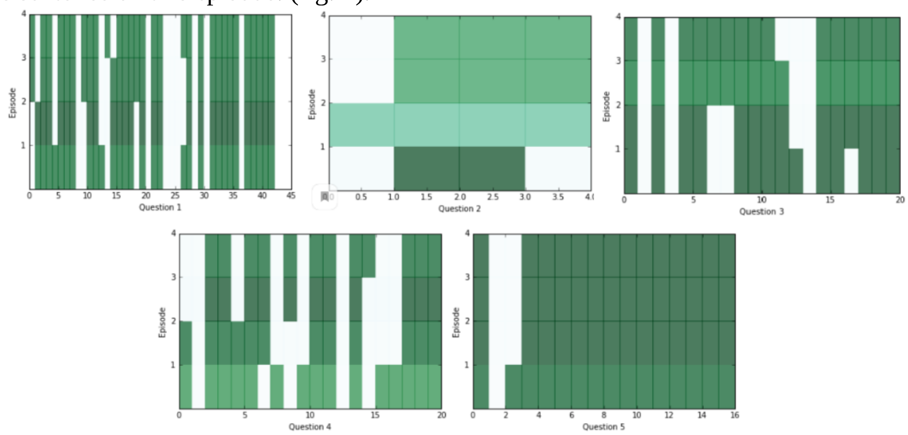

Fig. 4. Network testing.

You should see a change in attention between at least two episodes for each question, but sometimes the attention will be answered within one and sometimes it will take all four episodes. If the attention turns out to be empty, it can be saturated and pay attention to everything at once.

4 Conclusions

The information presented in this work may form the basis for further development and improvement of the intelligent information technology of tasks processing for open tests. According to the developed models, a software product was created for further practical use. Within this research theoretical material on natural language processing was analyzed.

The following steps are used to solve the neural network training problems: data collection for training, preparation and normalization of data, experimental selection of training parameters, own training, validation of training, adjustment of parameters, final training.

The software was implemented in Python and the TensorFlow library was used. A neural network-based quality control system was created using a closed domain.

References

1. Choi, K.R., Heilemann, M.V., Fauer, A., Mead, M.: A Second Pandemic: Mental Health Spillover From the Novel Coronavirus (COVID-19), Journal of American Psychiatric Nurses Association, 1078390320919803 (2020).
2. Plancher, K.D., Shanmugam, J.P., Petterson, S.C.: The Changing Face of Orthopedic Education: Searching for the New Reality After COVID-19. Arthroscopy, Sports Medicine and Rehabilitation, 10.1016/j.asmr.2020.04.007 (2020).
3. Chumachenko, D., Balitskii, V., Chumachenko, T., Makarova, V., Railian, M.: Intelligent expert system of knowledge examination of medical staff regarding infections associated with the provision of medical care. CEUR Workshop Proceedings,Vol. 2386, 2019, pp. 321-330. (2019).
4. Llamas, M., et. al.: Engineering education in Spain: Seven years with the Bologna Process: First results, 2018 IEEE Global Engineering Education Conference (EDUCON), Tenerife, pp. 1775-1780 (2018).
5. Polyvianna, Yu., et. al.: Computer Aided System of Time Series Analysis Methods for Forecasting the Epidemics Outbreaks, 2019 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), pp. 7.1-7.4 (2019).
6. Chumachenko, D., Chumachenko, T.: Intelligent Agent-Based Simulation of HIV Epidemic Process. Advances in Intelligent Systems and Computing, vol. 1020, c. 175-188. (2019).
7. Meniailov, I., et. al: Using the K-means method for diagnosing cancer stage using the Pandas library. CEUR Workshop Proceedings, vol. 2386, pp. 107-116. (2019).
8. Bazilevych, K., et.al.: Determining the Probability of Heart Disease using Data Mining Methods. CEUR Workshop Proceedings, vol. 2488, pp. 1-12. (2019).
9. Chumachenko, D., Chumachenko, K., Yakovlev, S.: Intelligent Simulation of Network Worm Propagation using the Code Red as an Example. Telecommunications and Radio Engineering, Vol. 78, Iss. 5, pp. 443-464 (2019).
10. Chumachenko, D., Yakovlev, S.: On Intelligent Agent-Based Simulation of Network Worms Propagation. 2019 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), pp. 3.11-3.13 (2019).
11. Piletskiy, P., et. al.: Development and Analysis of Intelligent Recommendation System Using Machine Learning Approach. Advances in Intelligent Systems and Computing, vol. 1113, pp. 186-197. (2020).

12. Bazilevych, K. et al.: Stochastic modelling of cash flow for personal insurance fund using the cloud data storage. International Journal of Computing, vol. 17, iss. 3, pp. 153-162. (2018).
13. Tretinjak, M.F.: Moving teaching from blackboard to the learning management system – Helping absent students learn from home. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, pp. 0500-0502. (2018).
14. Hidayat, N., Wardoyo, R., Azhari, S.: Educational Data Mining (EDM) as a Model for Students' Evaluation in Learning Environment. 2018 Third International Conference on Informatics and Computing (ICIC), Palembang, Indonesia, pp. 1-4. (2018).
15. Dotsenko, N., et. al.: Project-oriented management of adaptive teams' formation resources in multi-project environment. CEUR Workshop Proceedings, vol. 2353, pp. 911-923 (2019).
16. Dotsenko, N., et. al.: Management of critical competencies in a multi-project environment. CEUR Workshop Proceedings, vol. 2387, pp. 495-500. (2019).
17. Dotsenko, N., et. al.: Modeling of the process of critical competencies management in the multi-project environment. 2019 IEEE 14th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2019 – Proceedings, vol. 3, pp. 89-93. (2019).
18. Chang, C., et. al.: Effectiveness of Dual-Screen e-Learning Material Studying in 360-Degree VR Environment. 2019 International Symposium on Educational Technology (ISET), Hradec Kralove, Czech Republic, pp. 65-69. (2019).
19. Joseph Maria, I., Thirupathi, D., Rajendran, R., Velumani, B.: Technologies, Challenges and Tools for Digital Learning. 2019 IEEE Tenth International Conference on Technology for Education (T4E), Goa, India, pp. 268-269. (2019).
20. Soto, C., et. al.: Digital Educational Resources to Motivate Environmental Education in Rural Schools. 2019 XIV Latin American Conference on Learning Technologies (LACLO), San Jose Del Cabo, Mexico, pp. 265-271. (2019).
21. Chen, H., et. al.: Learning Student Networks via Feature Embedding. IEEE Transactions on Neural Networks and Learning Systems, pp. 1-11. (2020).
22. Aracena, J., Demongeot, J., Goles, E.: Positive and negative circuits in discrete neural networks. IEEE Transactions on Neural Networks, vol. 15, no. 1, pp. 77-83. (2004).
23. Somasundaram, S., Gobinath, R.: Current Trends on Deep Learning Models for Brain Tumor Segmentation and Detection – A Review. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, pp. 217-221. (2019).
24. Hains, G.J.D.R., Khmelevsky, Y., Tachon, T.: From natural language to graph queries. 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada, pp. 1-4. (2019).
25. Zhang, K. Research on the Optimizing Method of Question Answering System in Natural Language Processing. 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Jishou, China, pp. 251-254. (2019).
26. Saeed, B., et. al.: Comparative Analysis of Classifiers for EMG Signals. 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada, pp. 1-5. (2019).
27. Rao, Q., Yu, B., He, K., Feng, B.: Regularization and Iterative Initialization of Softmax for Fast Training of Convolutional Neural Networks. 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, pp. 1-8. (2019).

28. Mashtalir, V.P., Shlyakhov, V.V., Yakovlev, S.V.: Group structures on quotient sets in classification problems. In: Cybernetics and Systems Analysis, Vol. 50, №4, pp. 507-518 (2014).
29. Mashtalir, V.P., Yakovlev, S.V.: Point-set methods of clusterization of standard information. In: Cybernetics and Systems Analysis, Vol. 37, №3, pp. 295-307 (2001).
30. Zeng, Z., Gong, Q., Zhang, J.: CNN Model Design of Gesture Recognition Based on Tensorflow Framework. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, pp. 1062-1067. (2020).