ISSN 2079-3316 **PROGRAM SYSTEMS: THEORY AND APPLICATIONS** **vol. 13**, No 3(54), pp. 241–254
Research Article ARTIFICIAL INTELLIGENCE, INTELLIGENT SYSTEMS, NEURAL NETWORKS

UDC 004.932.75'1+004.89
10.25209/2079-3316-2022-13-3-241-254

Clustering of handwritten digits by Kohonen neural network

Dina Sergeevna **Latypova**[1], Dmitrii Nikolaevich **Tumakov**[2]
[1, 2] Kazan Federal University, Kazan, Russia
[1]✉*dina.latypova23@gmail.com* *(learn more about the authors on p. 254)*

Abstract. Clustering of handwritten digits is carried out for sixty thousand images contained in the training sample of the MNIST database. For clustering, the Kohonen neural network is used. For each handwritten digit, the optimal number of clusters (no more than 50) is determined. When determining the distance between objects (images of handwritten digits), the Euclidean norm is used. Checking the correctness of building clusters is carried out using data from the test sample of the MNIST database. The test sample contains ten thousand images. It is concluded that the images from the test sample belong to the "correct digit" cluster with a probability of more than 90%. For each digit, an F-measure is calculated to evaluate the clusters. The best F-measures are obtained for digits 0 and 1 (F-mean is 0.974). The worst values are obtained for the number 9 (F-mean is 0.903). A cluster analysis is also carried out, which allows drawing conclusions about possible errors in recognition by the Kohonen neural network. Intersections of clusters for images of handwritten digits are constructed. Examples of intersections of clusters are given, as well as examples of images that are incorrectly recognized by the neural network.

Key words and phrases: Kohonen neural network, clustering, MNIST

2020 *Mathematics Subject Classification*: 68T07; 68T45, 68T20

Acknowledgments:

[1] This paper has been supported by the Kazan Federal University Strategic Academic Leadership Program (PRIORITY-2030)

For citation: Dina S. Latypova, Dmitrii N. Tumakov. *Clustering of handwritten digits by Kohonen neural network* // Program Systems: Theory and Applications, 2022, **13**:3(54), pp. 241–254. http://psta.psiras.ru/read/psta2022_3_241-254.pdf

Эта статья по-русски: *http://psta.psiras.ru/read/psta2022_3_225-239.pdf*

Introduction

Handwriting recognition is a complex task that has not yet been fully resolved. The use of artificial intelligence in handwriting recognition significantly speeds up the process of its processing. One of the problems in pattern recognition by neural networks is related to the categorization of data. Data in the wrong category means incorrect information, which leads to errors. Image clustering is one of the effective approaches to solving this problem [1].

Pattern recognition acts as the main step to achieve clustering as this process analyzes the structure and vector value of each character in the dataset. For example, a consensus clustering algorithm for a set of handwritten digits was proposed in [2]. Robust continuous clustering was described in [3]. An algorithm to improve classification efficiency in recognizing handwritten digits was proposed in [4]. The c-means [5] and k-means [6, 7] approaches, as well as various neural networks, are often used to cluster handwritten digits. For example, in [8], handwritten letters were grouped using a Kohonen neural network. Different approaches for clustering: partition-based clustering, hierarchical clustering, density-based clustering, and mesh-based clustering were described in [9]. One of the most popular clustering algorithms, k-means, was described in [10]. Using this method, an accuracy of approximately 90% was achieved. In the case of large dimensions, subspace clustering [11, 12] can also be used.

In the present work, clustering of handwritten digits from the MNIST database is performed using the Kohonen neural network. The basis for our study was the results of the article [13], in which the clustering of handwritten digits was carried out, three clusters were selected for each digit, and it was shown that, due to the large size of the clusters, the digits from the test sample have a low hit in "their" clusters. Below, the number of clusters for each digit is chosen different, but not exceeding 50. The limitation of the number of clusters to 50 is due to the further use of clustering results for the Hopfield neural network, which has a limit on the number of input objects (clusters) [14]. To estimate the distance between images of digits, it is possible to use various metrics [15, 16], and we used the Euclidean norm. To assess the accuracy, a test sample of the MNIST database was taken, cluster intersections were determined, and F-measures were calculated for each digit.

Table 1. The number of images of each digit in the training and test samples from the MNIST dataset.

Digit	Training Set	Test Set
0	5923	980
1	6742	1135
2	5958	1032
3	6131	1010
4	5842	982
5	5421	892
6	5918	958
7	6265	1028
8	5851	974
9	5949	1009

1. MNIST Dataset

The *MNIST* database and its extensions are widely used to test various approaches to pattern recognition [17, 18], clustering [19], and other algorithms [20–22]. Many algorithms are tested against this database. For example, in pattern recognition, MNIST-based k-nearest neighbor methods give an error of 5%, convolutional neural networks less than 1%, and a multilayer perceptron about 2–5% depending on training methods and the number of layers.

The MNIST dataset is a database of handwritten digits that contains 60,000 images of digits in the training set and 10,000 images in the test set. Table 1 contains information on the number of images of each digit in the training and test sets. Each image in this dataset is represented in two forms: as a label and as a vector of pixel values, and is normalized by size. Each image is 28 by 28 pixels. The pixel value ranges from 0 to 255, where 0 is a black pixel and 255 is a white pixel.

2. Kohonen Neural Network

The Kohonen network is a special type of neural network for solving the clustering problem. It uses "unsupervised" learning and consists of a single layer of customizable weights. The weights of the neural network are changed in such a way that the vectors belonging to the same cluster drive the same output neuron [23]. The architecture of the Kohonen neural network is shown in Figure 1.

The outputs z_j of the Kohonen neural network are calculated via the

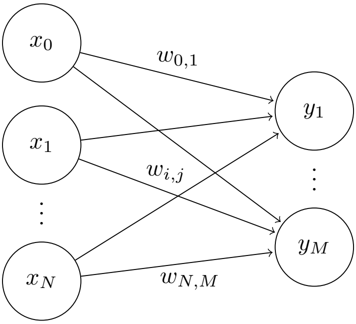


FIGURE 1. Kohonen Neural Network Architecture.

formula

$$z_j = \sum_{i=1}^{N=784} w_{ij} x_i,$$

where x_i are the inputs of the Kohonen neural network. Each image in the MNIST dataset has a size of 28×28, which means that the number of neurons in the Kohonen network will be $28 \times 28 = 784$. The weights w_{ij} are the centers of the clusters and participate in learning. Below is the Kohonen neural network training algorithm:

(1) The input vectors x^p are normalized.

(2) The values of the weights w_{ij} are chosen randomly from the vectors of the training set. This is due to the fact that in the case of an uneven distribution (if the weights are filled randomly), the weights may be located far from the input vectors and, for this reason, not take part in training.

(3) Distances between input vectors x^p with coordinates x_i^p and weights w_j are calculated using the following formula:

$$D_j = \sum_{i=1}^{N=784} ((x_i^p - w_{ij})^2)^{1/2}.$$

A winning vector, i.e. a vector w_j whose distance D_j to the input vector x^p is the smallest, is chosen.

(4) The coordinates of the winner vector selected in the previous step are changed according to the formula below, where θ is the learning rate:

$$w_l = w_l + \theta(x^p - w_l).$$

(5) Steps 3 and 4 are repeated for all vectors x^p.

The values of θ at each iteration are determined via the formula $\theta = \alpha\theta$, where $\alpha < 1$. If $\theta > \epsilon$, then transition to step 3 is carried out, and again after passing through all the input vectors, the weights are adjusted.

3. Number of clusters of each handwritten digit

The main problem with clustering handwritten digits is that all people have different handwriting, which means that the same digit can be written in different ways. Using the MNIST dataset as an example, one can see that there are many ways to write the same digit. The numbers have their own specific writing features, for example, the number 1 can be written with or without a slope. Each digit has a different number of such features. For this reason, the number of clusters into which a set of images must be divided for each digit may be different.

The number of clusters into which the input images must be divided is unknown. It is necessary to determine the optimal number of clusters for each digit. For automatic clustering, we will use the following algorithm [24]:

(*1*) Based on the Euclidean metric, we calculate the squared distances between all vectors in the training set. From the obtained values, we calculate the elements of the distance matrix

$$d^2(x^i, x^j) = \|x^i - x^j\|^2 = \sum_{i=1}^{N} (x_l^i - x_l^j)^2.$$

(*2*) Let us determine the maximum value of the elements of the matrix built at the previous step. This value is the maximum distance between all vectors: $\max(d^2(x^i, x^j))$.

(*3*) Let us give a definition to the allowable distance between vectors located in the same cluster, it is defined as a fixed percentage of the maximum distance between vectors.

(*4*) We choose an arbitrary ith column of the matrix (vector x^i). Then we mark all elements of the column with values less than allowed values as elements of the same cluster (row numbered j). We ignore the ith column, as well as all the jth columns.

(*5*) If the matrix is not empty yet, then transition to step 4 is carried out.

Figure 2 shows how the optimal number of clusters changes depending on the given percentage described in step 3. Let us build such a dependence and check for each digit.

Analysis of the obtained dependencies showed that they are the same for all digits. It is obvious that the smaller the percentage of the maximum

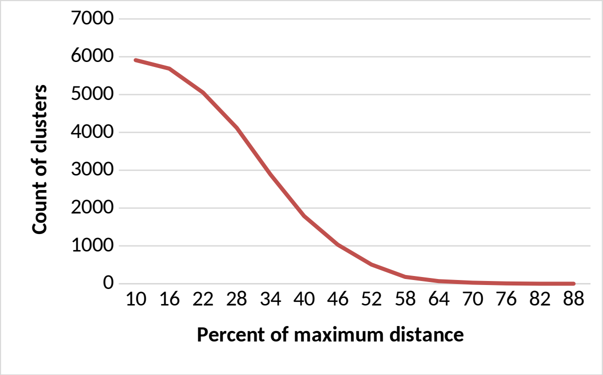


Figure 2. Dependence of the optimal number of clusters on a given percentage of the maximum distance.

Table 2. Optimal numbers of clusters for each digit.

Digit	Number of clusters
0	39
1	49
2	30
3	47
4	28
5	33
6	49
7	29
8	38
9	28

distance between the vectors, the more clusters into which it is necessary to divide the sample.

Let us examine how many clusters need to be selected, provided that their number does not exceed 50. We will select a fixed percentage for each indicator separately in the range from 65% to 70%. As a result, we obtain the number of clusters for each digit, which is shown in Table 2.

After determining the optimal number of clusters for each digit, we use the Kohonen neural network to find the clusters themselves.

4. Clustering with a Kohonen Neural Network

Let us carry out clustering of sixty thousand images of digits by the Kohonen neural network. The optimal parameters for network training

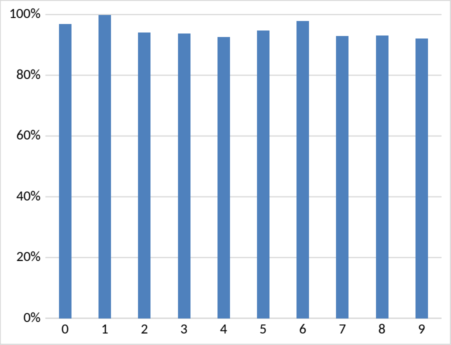


Figure 3. Percentage of digit images that fall from the test sample into its cluster.

Table 3. F-measure for clusters.

Digit	Recall	Precision	F-means
0	0.97	0.97	0.97
1	0.96	0.99	0.98
2	0.97	0.94	0.95
3	0.96	0.94	0.95
4	0.94	0.93	0.93
5	0.94	0.96	0.95
6	0.97	0.98	0.97
7	0.95	0.93	0.94
8	0.95	0.93	0.94
9	0.89	0.92	0.91

were obtained experimentally and have the following values: $\alpha = 0.96$, $\epsilon\ = 0.02$, $\theta = 0.6$.

Let us check the correctness of building clusters on a test data set. To do this, based on the MNIST test sample, we analyze the number of image vectors that fell into "own" cluster. Let us determine the distance from each image vector of a digit to all the resulting clusters for each of the digits. If the image vector is closest to the cluster that belongs to the group of clusters for the same picture, then we consider that the vector is in "own" cluster. Figure 3 shows the percentages of images from the test sample that fall into the clusters of their digits.

In addition, to analyze the accuracy of the results of clustering by the Kohonen neural network, we calculate the F-measure (see Table 3).

5. Cluster intersections

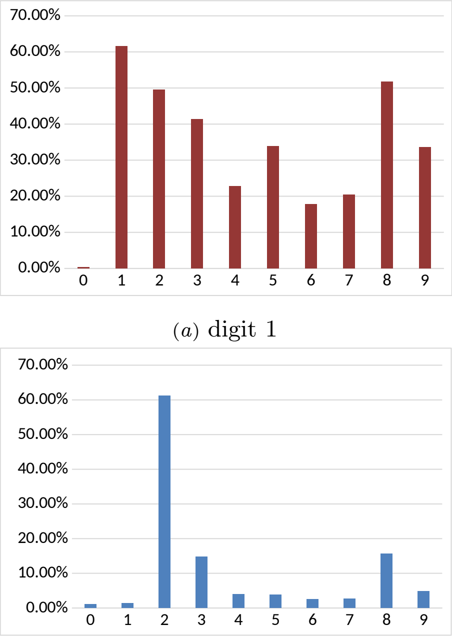
Now let us analyze the intersection of clusters with each other. By examining the intersections of clusters, one can answer the question of whether clustering using the Kohonen neural network is suitable for pattern recognition. It is obvious that the less intersections of clusters with each other, the more accurate the recognition.

The main problem with pattern recognition (and clustering) is that there are "infinite" ways to write the same digit. This leads to the fact that either the number of clusters increases significantly or the size of the clusters themselves increases. There is also another problem: different numbers are written in a similar or even almost identical way. This means that in the case of a small number of clusters, it can be expected that the clusters, due to their large size, will have multiple intersections with clusters of other digits.

To find intersections between clusters, it is necessary to determine how far from the center of each cluster is the digit from the sample. If the distance to the digit is less than the radius of the cluster, then we consider that the digit is in this cluster. We examine the intersections of the obtained clusters with images from the test sample.

- The clusters for digit 0 intersect with the clusters for digits 2 and 5.
- This is due to the fact that these numbers have the same "roundings".
- Clusters for digit 1 have more intersections than other clusters.
- This is due to the fact that the element of digit 1 is contained in the image of other digits.
- Clusters for digit 2 have significant overlaps with clusters for digits 3 and 8.
- Clusters for digit 3 have intersections with digits 2, 5 and 8.
- Clusters for digits 4 and 9 have many intersections with each other.
- Clusters for digit 5 overlap with clusters for digits 3 and 8.
- Clusters for digit 6 overlap with clusters for digit 2.
- Clusters for digit 7 overlap with clusters for digit 9.
- Clusters for digit 8 overlap with clusters for digits 2 and 9.

Figure 4 shows examples of the charts showing what percentage of the images of each digit from the test sample belongs to the cluster of the considered digit. For example, in Figure 4(a), the brick color (top right sector) shows that 61.64% of digit 1 images belong, in addition to their



(a) digit 1

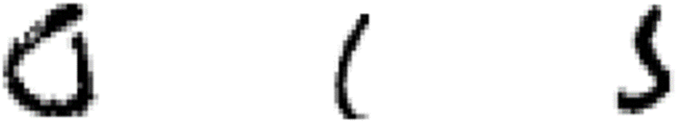
(b) digit 2

Figure 4. Examples of intersections of clusters for digits

own digit 1 cluster, to another digit 1 cluster. Effectively, this means that 61.64% images of digit 1 belong to two or more clusters of digit 1. If we move clockwise to the next (gray) sector for digit 2, we see that 49.57% of images of digit 2 fall in addition to their own cluster, into the cluster of digit 1.

Also note (this is not shown in the charts) that many images of digit 1 fall into clusters of different digits. An analysis of the intersection results showed that many images from the test sample of digit 1 fell into the intersection with images of other digits: almost 23% of the test sample of digit 1 fell into the intersection of clusters for digit 0; almost 50% — with clusters for digit 2 and so on. This is due to the fact that elements of the image of other digits contain the image of digit 1.

For digit 2, the situation is different (Figure 4(b)). There is one large intersection with clusters for digit 2 (self-intersections are typical for clusters of all digits) and small intersections with clusters of other digits.



(*a*) 0 at the intersection with 6

(*b*) 1 at the intersection with 6

(*c*) 5 at the intersection with 3

Figure 5. Examples of digits that are located at the intersections of clusters

Let us give examples of images that fell into the intersections of their cluster and the cluster of another digit (Figure 5). In these examples, the neural network correctly groups the images. In Figure 5(a), the digit was recognized as 0, but it was in the intersection with the cluster for digit 6. In Figure 5(b), the digit was recognized as digit 1, but also belongs to the cluster for digit 6, in Figure 5(c) the digit was recognized as digit 5, but also belongs to the cluster for digit 3.

Discussions

The main problem with clustering digit images is that the same digits are written differently. This leads to an increase in the size of individual clusters or their number. There is also a problem with the fact that different numbers are written in a similar way. Therefore, in the case of a small number of clusters, it is expected that due to their large size there will be intersections of clusters with each other.

After analyzing Figure 4 and Table 3, we can conclude that the Kohonen neural network can be used for clustering and recognizing handwritten digits, since a large percentage of digit images from the test sample fall into the required cluster. Note that digit 9 is recognized as the worst – the percentage of recognition is slightly more than 90%. Digit 1 is recognized by the Kohonen neural network in almost 100% of cases.

Other neural networks can also be used to solve recognition problems [**25**, **26**]. Conclusions about the belonging of an image to its cluster can be used as one of the stages of recognition. For example, clustering can be used as such a stage in a hierarchical neural network [**27**, **28**].

It should be noted that the proposed clustering method for a limited number of clusters can be used in the problem of recognition by the Hopfield neural network, since this Hopfield neural network has a limit on the number of memorized objects (clusters).

Conclusion

A cluster analysis of handwritten digits from the MNIST database is carried out using the Kohonen neural network. For each digit, the optimal number of clusters is determined. Images from the test sample belong to the correct cluster with a probability of more than 90% for each digit. It is concluded that clustering can be used to recognize handwritten digits. The best clustering is obtained for digits 0 and 1 (F-score is 0.97). The worst clustering is obtained for digit 9 — the F-score is 0.903.

Intersections of clusters are determined and it is analyzed which digits intersect with which clusters. Examples of individual characteristic images of handwritten digits that fell into the intersections of clusters are given.

The proposed approach can be used to cluster large amounts of data of various types and complexity.

References

[1] D. Latypova. *Neural networks using for handwritten numbers recognition*, Master's Thesis, Czech Technical University, Prague, 2020, 77 pp. URL ↑242

[2] M. Rexy, K. Lavanya. "Handwritten digit recognition of MNIST data using consensus clustering", *International J. of Recent Technology and Engineering*, **7**:6 (2019), pp. 1969–1973. URL ↑242

[3] S. Nhery, R. Ksantini, M. B. Kaaniche, A. Bouhoula. "A novel handwritten digits recognition method based on subclass low variances guided support vector machine", *Proc. of the 13th Int. Joint Conf. on Computer Vision, Imaging and Computer Graphics Theory and Application*. V. 4, VISIGRAPP 2018 (27–29 January 2018, Funchal, Madeira, Portugal), 2018, ISBN 978-989-758-290-5, pp. 28–36. URL ↑242

[4] S. A. Shal, V. Koltun. "Robust continuous clustering", *Proc. the Natl. Acad. Sci. USA*, **114**:37 (2017), pp. 9814–9817. doi ↑242

[5] E. Miri, S. M. Razavi, J. Sadri. "Performance optimization of neural networks in handwritten digit recognition using Intelligent Fuzzy C-Means clustering", 2011 1st International eConference on Computer and Knowledge Engineering (ICCKE) (13–14 Oct. 2011, Mashhad, Iran), 2011, pp. 150–155. doi ↑242

[6] S. Pourmohammad, R. Soosahabi, A. S. Maida. "An efficient character recognition scheme based on k-means clustering", 2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO) (28–30 April 2013, Hammamet, Tunisia), 2013, pp. 1–6. doi ↑242

[7] B. Y. Li. “An experiment of k-means initialization strategies on handwritten digits dataset”, *Intelligent Information Management*, **10**:2 (2018), pp. 43–48. doi ↑242

[8] L. C. Munggaran, S. Widodo, A. M. Cipta, Nuryuliani. “Handwritten pattern recognition using Kohonen neural network based on pixel charactacter”, *International J. of Advanced Computer Science and Applicatons*, **5**:11 (2014), 6 pp. doi URL ↑242

[9] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Zomaya, S. Foufou, A. Bouras. “A survey of clustering algorithms for big data: taxonomy and empirical analysis”, *IEEE Transactions on Emerging Topics in Computing*, **2**:3 (2014), pp. 267–279. doi ↑242

[10] Y. Bi, P. Wang, X. Guo, Z. Wang, S. Cheng. “K-means clustering optimizing deep stacked sparse autoencoder”, *Sensing and Imaging*, **20**:1 (2019), 6, 19 pp. doi ↑242

[11] Y. Chen, C.-G. Chen, C. You. “Stochastic sparse subspace clustering”, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (13–19 June 2020, Seattle, WA, USA), 2020, pp. 4155–4164. doi ↑242

[12] S. Zhang, C. You, R. Vida, C. G. Li. “Learning a self-expressive network for subspace clustering”, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (20–25 June 2021, Nashville, TN, USA), 2021, pp. 12393–12404. doi ↑242

[13] D. Latypova, D. Tumakov. “Opredeleniye osnovnykh klasterov rukopisnykh tsifr”, Tsifrovaya obrabotka signalov i yee primeneniye. DSPA - 2020 (14–15 aprelya 2020 goda, Moskva, Rossiya), 2020, pp. 620–625.↑242

[14] D. Latypova, D. Tumakov. “Peculiarities of image recognition by the Hopfield neural network”, IEMAICLOUD 2021: International Conference on Intelligent Emerging Methods of Artificial Intelligence & Cloud Computing, Smart Innovation, Systems and Technologies, vol. **273**, ed. Garcia Marquez F.P., Springer, Cham, 2021, ISBN 978-3-030-92904-6, pp. 34–47. doi ↑242

[15] S. McConnell, R. Sturgeon, G. Henry, A. Mayne, R. Hurley. “Scalability of self-organizing maps on a GPU cluster using OpenCL and CUDA”, *High Perfomance Computer Symposium, J. Physics: Conf. Series*, **341** (2012), 012018, 10 pp. doi ↑242

[16] Y. Xu, W. Zhang. “On a clustering method for handwritten digit recognition”, 2010 Third International Conference on Intelligent Networks

and Intelligent Systems (01–03 November 2010, Shenyang, China), 2010, pp. 112–115. doi ↑242

[17] G. Cohen, S. Afshar, J. Tapson, A. van Schaik. “EMNIST: extending MNIST to handwritten letters”, 2017 International Joint Conference on Neural Networks (IJCNN) (14–19 May 2017, Anchorage, AK, USA), 2017, pp. 2921–2926. doi ↑243

[18] A. Baldominos, Y. Saez, P. Isasi. “A survey of handwritten character recognition with MNIST and EMNIST”, *J. of Applied Science*, **9**:15 (2019), 3169, 16 pp. doi ↑243

[19] A. F. Agarap, A. P. Azcarraga. “Improving k-means clustering performance with disentangled internal representations”, 2020 International Joint Conference on Neural Networks (IJCNN) (19–24 July 2020, Glasgow, UK), 2020, pp. 1–8. doi ↑243

[20] K. Cheng, R. Tahir, L. C. Eric, M. Li. “An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset”, *J. Multimedia Tools and Applications*, **79** (2020), pp. 13725–13752. doi ↑243

[21] J. Kossen, S. Farquhar, Y. Gal, T. Rainforth. “Active testing: sample-efficient model evaluation”, *Proc. of the 38th International Conf. on Machine Learning* (18-24 July 2021, Virtual), Proceedings of Machine Learning Research, vol. **139**, 2021, pp. 5753–5763. URL ↑243

[22] R. Zhang, P.-C. Chang. “Robustness against adversary models on MNIST by deep-Q reinforcement learning based parallel-GANs”, 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (14–17 December 2021, Tokyo, Japan), 2021, pp. 1590–1597. URL ↑243

[23] F. Murtagh, M. Hernandez-Pajares. “The Kohonen self-organizing map method: An assessment”, *J. of Classification*, **12** (1995), pp. 165–190. doi ↑243

[24] I. S. Sen’kovskaya, P. V. Sarayev. “Automatic clustering in data analysis based on Kohonen self-organizing maps”, *Vestnik MGTU im. G.I. Nosova*, 2011, no. 2(34), pp. 78–79 (in Russian). URL ↑245

[25] P. Y. Simard, D. Steinkraus, J. C. Platt. “Best practices for convolutional neural networks applied to visual document analysis”, *Proc. Seventh International Conference on Document Analysis and Recognition* (06 August 2003, Edinburgh, UK), 2003, ISBN 0-7695-1960-1, pp. 958–963. doi ↑250

[26] D. Ciresan, U. Meier, J. Schmidhuber. “Multi-column deep neural networks for image classification”, 2012 IEEE Conference on Computer

Vision and Pattern Recognition (16–21 June 2012, Providence, RI, USA), 2012, pp. 3642–3649. doi ↑250

[27] Z. Kayumov, D. Tumakov, S. Mosin. "Hierarchical convolutional neural network for handwritten digits recognition", *Proc. Computer Science*, **171** (2020), pp. 1927–1934. doi ↑250

[28] Z. Kayumov, D. Tumakov, S. Mosin. "Combined convolutional and perceptron neural networks for handwritten digits recognition", 2020 22th International Conference on Digital Signal Processing and its Applications (DSPA) (25–27 March 2020, Moscow, Russia), 2020, pp. 74, 5 pp. doi ↑250

Received 28.06.2022;
approved after reviewing 12.08.2022;
accepted for publication 10.09.2022.

Recommended by *prof. A. M. Elizarov*

Information about the authors:

Dina Sergeevna Latypova

Ph.D. student, Institute of Computational Mathematics and Information Technologies, Kazan (Volga Region) Federal University. Research interests include artificial intelligence, neural networks, high performance computing

iD 0000-0002-3282-8545
e-mail: *dina.latypova23@gmail.com*

Dmitrii Nikolaevich Tumakov

Ph.D. (Physics and Mathematics), Deputy Director for Research, Institute of Computational Mathematics and Information Technologies, Kazan (Volga Region) Federal University. Research interests include machine learning, artificial intelligence, machine vision, pattern recognition, mathematical modeling, high performance computing.

iD 0000-0003-0564-8335
e-mail: *dtumakov@kpfu.ru*

The authors contributed equally to this article.

The authors declare no conflicts of interests.