IOP Conference Series: Materials
Science and Engineering

**PAPER • OPEN ACCESS**

# Classification of handwritten digits using the Hopfield network

IOP Conf. Series: Materials Science and Engineering **862** (2020) 052048 doi:10.1088/1757-899X/862/5/052048

# Classification of handwritten digits using the Hopfield network

**M A Belyaev[1] and A A Velichko**

Institute of Physics and Technology, Petrozavodsk State University, 33 Lenin str., 185910, Petrozavodsk, Russia

[1] E-mail: biomax89@yandex.ru

**Abstract.** The paper presents the results of the classification of handwritten digits from the MNIST database using the Hopfield network. A strong correlation of training binary patterns does not allow the use of the standard Hebbian learning method. The application of the Storkey learning method increases the capacity of associative memory, and the optimized pattern binarization threshold and pattern size reduce the correlation of patterns. By optimizing these parameters, a network achieved a classification accuracy of 56.2% on a set of validation data used for network training. The selection of the optimal binarization threshold for a separate set of test images increased the classification accuracy to 61.5%.

## 1. Introduction

A long history of the research on artificial neural networks originates from the simulation of an artificial neuron by McCallock and Pitts in 1943 [1]. The rapid development of cybernetics and computer science has led to the emergence of a wide variety of neuron models and architecture of neural networks [2]. Some types of neural networks were created by copying complex biological processes [3,4], while other networks used simple models of neurons with the necessary functionality [5,6]. One of the most important functions of higher nervous activity is the storage of information, which can be implemented using recurrent artificial networks [7,8]. Feedback functionality allows the implementation of own memory within recurrent networks, and this memory is used during network training and operation [9]. The memory of a recurrent network has associative properties, which allow the mapping of one of the stored associations to any input action. A Hopfield network, described in 1982 [10], is considered to be the first network of this type. It consists of one layer of neurons, and each neuron is recurrently connected with the remaining neurons of the network. Network neurons take binary values (1 and -1) and have a stepwise activation function. The main advantage of the Hopfield network is the possibility of training the network (calculating coupling weights) in one action using patterns from the training set and the Hebbian rule [10]. To check a test pattern, the state of network neurons is set in accordance with the value of this test pattern. The network sequentially updates the state of neurons until their state matches one of the training patterns. In this way, the association between the test pattern and the training set is built.

Recently, Hopfield networks have come into focus due to actively developing oscillatory neural networks (ONN) [11], which can be implemented in the form of electronic circuits with a fully connected array of micro- and nano-oscillators. Feedback in oscillators is implemented through the processes of their interaction. Information in the ONN can be encoded by the frequency of the oscillators

[12], by the value of synchronization [13], or by the phase difference between the signals of the oscillators [14].

A network becomes similar to the Hopfield network, if two conditions are applied. First, the information can be represented in binary form, for example, the phase can take values 0 and $\pi$. Second, each oscillator can affect the state of the associated oscillator depending on the coupling coefficient. For example, the oscillations of the associated oscillator can be in phase or out of phase. Hoppensteadt and Izhikevich [15] proposed this analogy for the first time based on the PLL (Phase lock loop) array. The idea was continued in [16–18], where options for the implementation of the Hopfield network using a small number of harmonic oscillators on discrete elements and a nonlinear input signal were presented. In a number of studies, relaxation oscillators based on bistable elements were used to implement the network [11,14]. These studies build a mathematical analogy between the dynamics of the array of oscillators and the dynamics of changes in the states of neurons in the Hopfield-like network. Thus, the design of the ONN functionality can be simplified using Hopfield network modelling.

The pattern classification and recognition, which is one of the most important tasks of machine learning, can be resolved using the Hopfield network and subsequently transferred to a hardware implementation based on ONN. Using the Hopfield network, the implementation task consists of two parts: the first part is to develop hardware that implements mathematical similarity, and the second part is to model the classification process based on the Hopfield network. Attempts to implement the network hardware have been made earlier [16,17,19–21], while the available solutions to the classification problem are based on simplified examples [22–25]. The current literature lacks the application of Hopfield networks to classify handwritten characters from standardized databases. In this study, we demonstrate the capabilities of Hopfield networks on the classic example of classification of handwritten digit's images from the MNIST database [26].

## 2. Network architecture

The MNIST database is a sample of 70 thousand digitized images of handwritten digits. Each image has a size of 28 by 28 pixels and is presented in grayscale. Each image pixel can take values from 0 (black) to 255 (white). The image database is divided into two parts: a training part (60 thousand samples) and a test part (10 thousand samples). The former is used for training and network validation, and the latter is for assessing the accuracy of the classification of a trained network.

As a pre-processing of images of digits with different tilt angles, we used the deskewing method. The method allows adjusting the linear and angular displacement of the image based on the brightness distribution of the pixels in the picture. In the processed image, the digit is located in the center of the picture and has a normal tilt (the tilt angle corresponds to ~ 90 °). Figure 1 demonstrates the images before processing and after the deskewing method application.

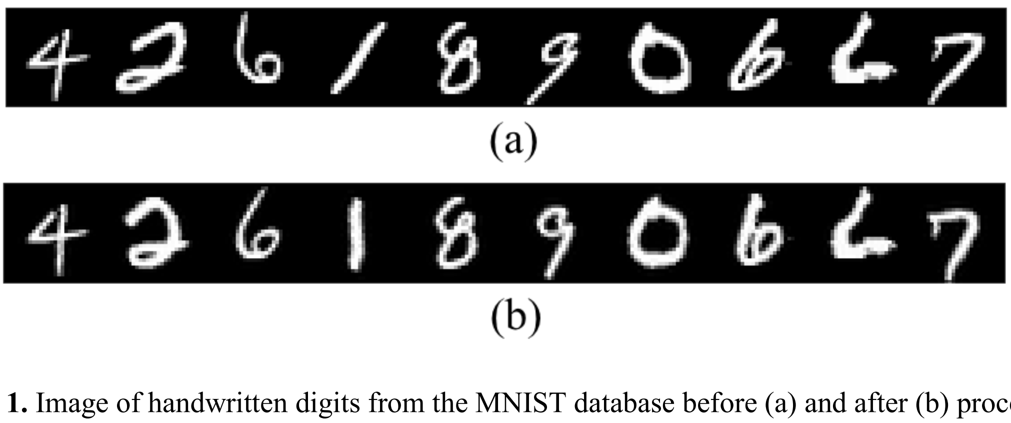


**Figure 1.** Image of handwritten digits from the MNIST database before (a) and after (b) processing by the deskewing method.

Next, the images were averaged from the training sample for each class (digit). To calculate the weight matrix of the Hopfield network, the obtained two-dimensional images were represented in the form of one-dimensional arrays $\mathbf{P}_{\mathrm{mem}_i}$ using string concatenation (where $i$ = 1..10 is index of the memorized pattern). As Hopfield networks operate with binary values (+1 or -1), one-dimensional arrays $\mathbf{P}_{\mathrm{mem}_i}$ must be binarized. The initial values of each element of the array $\mathbf{P}_{\mathrm{mem}_i}$ belong to the range from 0 to 255, and the following equation is used to convert them to binary format:

$$x_{mem_{i,j}} = \begin{cases} 1, \ if \ p_{mem_{i,j}} \geq p_{\mathrm{th}} \\ -1, \ if \ p_{mem_{i,j}} < p_{\mathrm{th}} \end{cases}, \tag{1}$$

where $p_{mem_{i,j}}$ is the $j$-th ($j$ = 0..784) element of the array $\mathbf{P}_{\mathrm{mem}_i}$, and $p_{\mathrm{th}}$ is the binarization threshold. Initially, the middle of the range ($p_{\mathrm{th}}$ = 127) is chosen as the threshold. The resulting binary one-dimensional vectors $\mathbf{X}_{\mathrm{mem}_i}$, corresponding to digits from 0 to 9 and called patterns, are used to calculate the weight matrix $\mathbf{W}$ according to the Hebbian learning rule:

$$\boldsymbol{W} = \frac{1}{N} \cdot \sum_{i=1}^{N} \boldsymbol{X}_{mem_i} \cdot \boldsymbol{X}_{mem_i}^{T}, \tag{2}$$

where $N$ is the number of memorized patterns. Classification in the Hopfield network is an iterative process of changing the state of network neurons, when the classified vector $\mathbf{X}_{\mathrm{test}}$ must be converted to the closest pattern $\mathbf{X}_{\mathrm{mem}_i}$ from the training set. Mathematically, this process is represented as a recursive equation for multiplying the vector $\mathbf{X}_{\mathrm{s}}$, which describes the $s$-th state of the network, by the weight matrix $\mathbf{W}$:

$$\mathbf{X}_{s+1} = \theta \left(\mathbf{X}_{s} \cdot \mathbf{W}\right), \tag{3}$$

where $\theta$ is the matrix activation function of network neurons. As $\theta$, a threshold function is used in Hopfield networks that converts each element of the array to 1 if the element is non-negative, or to -1, otherwise. The initial state of the network is set equal to the classified vector $\mathbf{X}_0 = \mathbf{X}_{\mathrm{test}}$. In case of correct operation of the network, its state after several iterations will correspond to one of the memorized patterns $\mathbf{X}_{\mathrm{mem}_i}$. For each memorized pattern, the following equation must be fulfilled:

$$\mathbf{X}_{\mathrm{mem}_i} = \theta \left(\mathbf{X}_{\mathrm{mem}_i} \cdot \mathbf{W}\right), \tag{4}$$

that is, all memorized patterns must be stable network states. In some cases, condition (4) is not satisfied and the initial state of the network with $\mathbf{X}_0 = \mathbf{X}_{\mathrm{mem}_i}$ will be unstable. As a result, after several iterations, the system will transition to a new state called the spurious pattern. Typically, this behavior occurs when recording a large number of patterns. When using the Hebbian learning rule, the network can operate correctly when writing no more than $0.138 \cdot M$ patterns into it, where $M$ is the number of pattern elements equal to the number of neurons [27].

When classifying images from the MNIST database, the number of pixels in the image is $M = 28 \cdot 28 = 784$, which allows recording no more than 108 patterns. However, in the course of this study, a problem was discovered that the network does not operate correctly when inputting 10 recorded $\mathbf{X}_{\mathrm{mem}_i}$ patterns. Probably, it can be explained by the strong correlation between the patterns [28,29]. Because of this problem, the number of patterns that can be written to the network drops significantly, and the correct operation of the network was possible when recording no more than 2 patterns. There are two ways to minimize this effect: modifying the network training methodology (calculating the weight matrix $\mathbf{W}$) and reducing the correlation between $\mathbf{X}_{\mathrm{mem}_i}$.

The capacity of associative memory in the Hopfield network could be increased by several other training methods [30,31]. The Storkey method [30] is a modification of the Hebbian learning rule and demonstrates the best results. In the method, when calculating the matrix $\mathbf{W}$, the local field form is used, where the change in the synoptic connection between neurons depends only on the state of the neurons connected by this connection. The equation for calculating the weight matrix by the Storkey method can be expressed through the recurrence formula:

$$\mathbf{W}_i = \mathbf{W}_{i-1} + \frac{1}{N}\mathbf{X}_{\mathrm{mem}_i} \cdot \mathbf{X}^{\mathrm{T}}_{mem_i} - \frac{1}{N}\cdot \mathbf{X}_{\mathrm{mem}_i} \cdot (\mathbf{W}_{i-1} \cdot \mathbf{X}_{\mathrm{mem}_i})^{\mathrm{T}} - \frac{1}{N}\cdot (\mathbf{W}_{i-1} \cdot \mathbf{X}_{\mathrm{mem}_i}) \cdot \mathbf{X}^{\mathrm{T}}_{mem_i}. \quad (5)$$

During network training using the Storkey method, the correct operation of the network was observed when already 4 patterns were recorded, or the associative memory capacity was doubled in comparison with the Hebbian method.

Further optimization of the network was directed to decrease the correlation between the $\mathbf{X}_{\mathrm{mem}_i}$ patterns by evaluating the similarity of all pairs of recorded vectors $\mathbf{X}_{\mathrm{mem}_i}$. To evaluate the similarity of patterns, we used the Hamming distance $D_{\mathrm{h}}$, which for binary vectors is the number of positions (pixels) that differ between patterns from each pair. The Hamming distance calculated for each pair of patterns is presented in figure 2. The most similar digits are “4” and “9” ($D_{\mathrm{h}}$ = 46), “7” and “9” ($D_{\mathrm{h}}$ = 58), “3” and “5” ($D_{\mathrm{h}}$ = 74), “4” and “7” ($D_{\mathrm{h}}$ = 76).

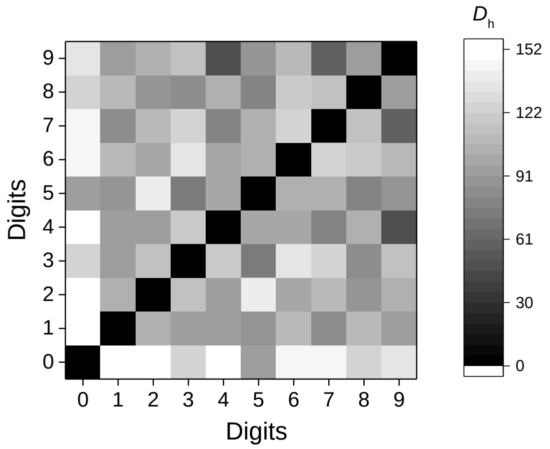
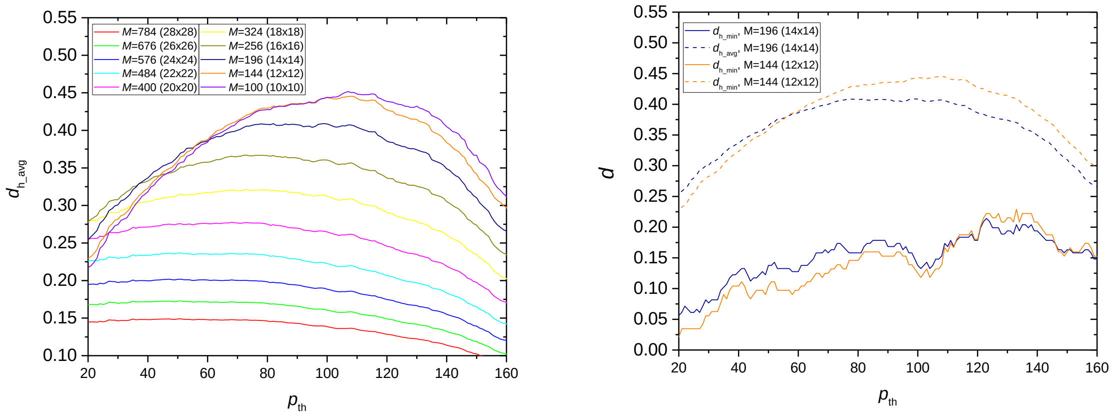



**Figure 2.** The value of the Hamming distance $D_{\mathrm{h}}$ between all pairs of patterns.

In the analysis of the similar positions between the digits, areas of maximum similarity were found that are located on the borders of the patterns. This area is a black frame around the digit (see figure 1), which does not carry useful information, but increases the similarity of $\mathbf{X}_{\mathrm{mem}_i}$ patterns. When cropping this frame, $D_{\mathrm{h}}$ will not change, since the number of different pixels will remain unchanged. At the same time, the difference in the patterns $\mathbf{X}_{\mathrm{mem}_i}$ depends on the length of the vector $M$. Therefore, to evaluate the similarity of the vectors, it is necessary to calculate the relative Hamming distance $d_{\mathrm{h}} = D_{\mathrm{h}} / M$. In addition, the binarization threshold $p_{\mathrm{th}}$ can influence the similarity of vectors. Binarization threshold determines which pixels will correspond to the white silhouette of a digit, when moving to a black-and-white image. As a metric for the similarity of patterns, we use the average $d_{\mathrm{h\_avg}}$ and the minimum $d_{\mathrm{h\_min}}$ relative distance between all pairs of patterns. Figure 3a demonstrates a plot of $d_{\mathrm{h\_avg}}$ versus $p_{\mathrm{th}}$ for various values of $M$. With a decrease in $M$ from 784 (28x28) to 144 (12x12), there is a clear increase in the maximum value of $d_{\mathrm{h\_avg}}$ with a shift of the maximum towards larger $p_{\mathrm{th}}$ values. When $M$ decreases, the number of patterns that the network can remember is linearly reduced. Based on figure 3a, the optimal value of the pattern size lies in the range from 12x12 to 14x14. In order to choose from two options for cropping the pattern, figure 3b demonstrates the dependences of $d_{\mathrm{h\_avg}}$ and $d_{\mathrm{h\_min}}$ on $p_{\mathrm{th}}$ at $M$ = 196 and $M$ = 144. The $d_{\mathrm{h\_avg}}$ plots from $p_{\mathrm{th}}$ have maxima at $p_{\mathrm{th}}$ = 82 for $M$ = 196 and $p_{\mathrm{th}}$ = 108 for $M$ = 144, which are significantly offset from the $d_{\mathrm{h\_min}}$ maxima. Therefore, when choosing the optimal $p_{\mathrm{th}}$ value, it is necessary to find the value of the combination of these parameters. For example, we used the extremum of the product function $d_{\mathrm{h\_avg}} \cdot d_{\mathrm{h\_min}}$. For $M$ = 196, the optimal value is $p_{\mathrm{th}}$ = 86, and for $M$ = 144, the optimal value is $p_{\mathrm{th}}$ = 123.

IOP Conf. Series: Materials Science and Engineering **862** (2020) 052048 doi:10.1088/1757-899X/862/5/052048




(a) (b)

**Figure 3.** The dependence of $d_{\mathrm{h\_avg}}$ on $p_{\mathrm{th}}$ for various values of $M$ (a) and the dependence of $d_{\mathrm{h\_avg}}$ and $d_{\mathrm{h\_min}}$ on $p_{\mathrm{th}}$ for $M = 196$ and $M = 144$ (b).

Using the specified parameters and the Storkey training method, expression (4) is satisfied, and the recorded patterns correspond to the stable states of the Hopfield network. Despite a slightly larger $d_{\mathrm{h\_avg}}$ value at $M = 144$, a significant decrease in the number of neurons at $M = 196$ leads to a large number of spurious patterns that reduce the classification accuracy. Therefore, we demonstrate the results of the network at $M = 196$ and $p_{\mathrm{th}} = 86$. Figure 4 presents images of training patterns.

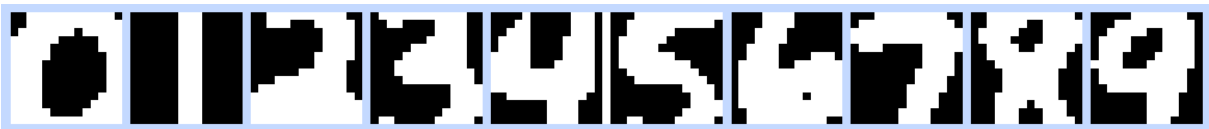


**Figure 4.** The training patterns used to calculate the weight matrix $\mathbf{W}$.

## 3. Image classification

Validation of the classification results was performed on 10000 randomly selected samples from the training dataset. During validation, the parameters are fine-tuned before checking the classification accuracy on the test sample. A deskewing procedure was performed for each input image, then the image was cropped to a size of 14x14 pixels. Next, the image was converted to black and white, with the binarization threshold $p_{\mathrm{th}} = 86$, and the test vector $\mathbf{X_{test}}$ was formed by concatenating the strings. The neurons of the network were set to the initial state corresponding to the test vector, and the network iteratively converged to a stable state, according to equation (3). After that, the classification was checked for correctness. The image was considered successfully recognized, if the final network state corresponded to one of the memorized patterns, and the digits depicted on the pattern and test vector coincided. Figure 5 presents examples of the correct and incorrect recognition of handwritten digits indicating the source image and the evolution of the network state during the process. The digit ‘3’ was recognized correctly in just two iterations of the network. The digit ‘9’ was incorrectly classified by the network as the digit ‘1’. When classifying the digit ‘8’, the network went into a state that does not correspond to any of the recorded patterns (spurious pattern). In assessing the classification accuracy, the transition to a spurious pattern was considered an incorrect result.

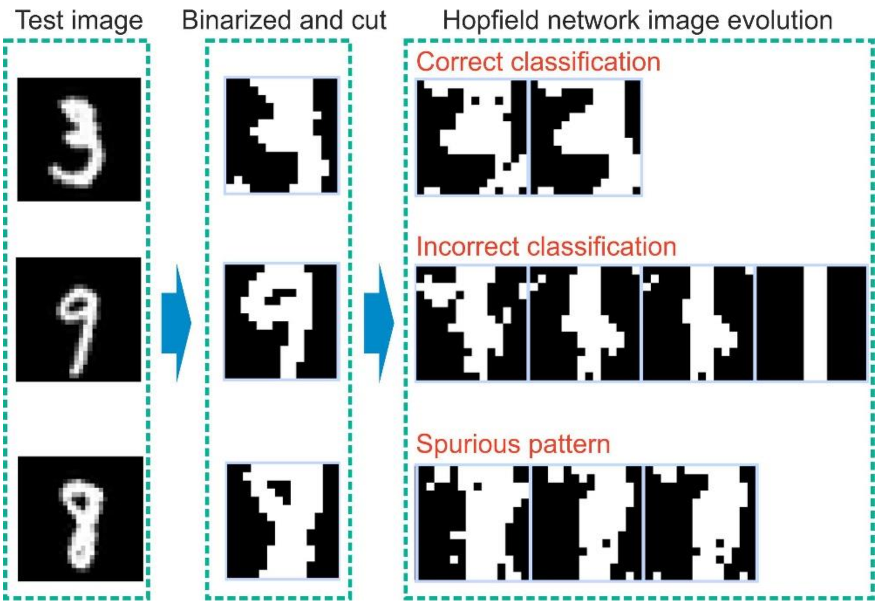



**Figure 5.** Examples of correct and incorrect classification of images from the MNIST database.

The result of operation on validation data demonstrated the classification accuracy of 56.2%. The accuracy depends on the binarization threshold $p'_{th}$ of the classified digits. Figure 6 reflects the dependence of classification accuracy using validation data on $p'_{th}$. The maximum value of the classification accuracy reached 59.8% at $p'_{th} = 38$. For optimal classification, the binarization threshold of the test images should differ from the binarization threshold of the patterns ($p_{th} = 86$).

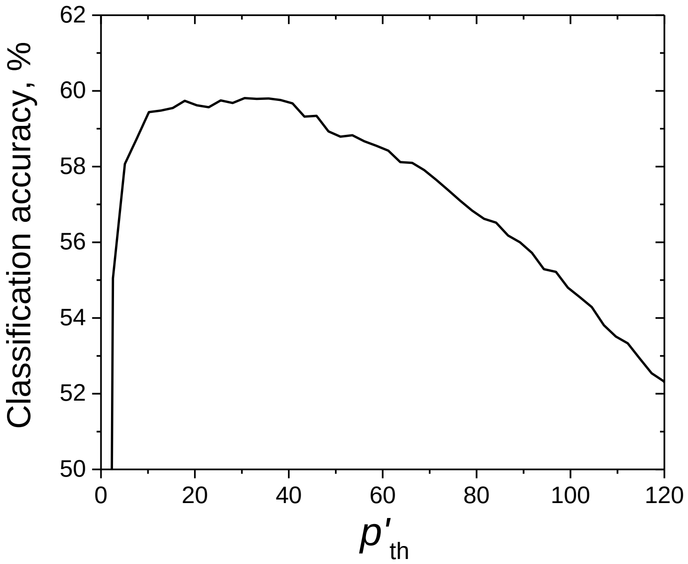



**Figure 6.** The dependence of the classification accuracy on the binarization threshold of classified images $p'_{th}$.

Table 1 presents the percentages of the correct and incorrect classification, and the percentage of occurrence of spurious patterns at $p'_{th} = 38$. For the results of incorrect classification, the most common erroneous result is indicated.

**Table 1.** Classification results of the MNIST database for single digits.

| Digit | Percent of correct classification, % | Percent of incorrect classification, % | The most common erroneous result | Percent of spurious patterns, % |
|---|---|---|---|---|
| 0 | 69 | 5.49 | 6 | 25.5 |
| 1 | 90.2 | 0.35 | 7 | 9.49 |
| 2 | 73.7 | 4.44 | 1 | 21.9 |
| 3 | 67.7 | 8.72 | 2 | 23.5 |
| 4 | 0 | 16.1 | 6 | 83.9 |
| 5 | 52.4 | 9.04 | 6 | 38.5 |
| 6 | 80.4 | 3.94 | 1 | 15.7 |
| 7 | 69.6 | 4.86 | 2 | 25.5 |
| 8 | 57.3 | 13.3 | 3 | 29.3 |
| 9 | 29.9 | 25.4 | 7 | 44.8 |

All test digits have a different percentage of correct classification. The digits '1', '2', '6', '7' have the highest classification probabilities, because, most likely, their patterns are strong attractors of the network. This assumption confirms the fact that these digits are the most likely erroneous result of the classification of other digits. The digit '1' is most accurately recognized, and this is, probably, due to the small variability of its way of writing. Digits '9' and '4' have the smallest average relative Hamming distance with respect to other digits, so they have the largest percentage of incorrect classification. The digit '4' has never been correctly classified, and most of the erroneous results relate to spurious patterns. The most common spurious pattern differs from the memorized pattern by only 1 pixel. It means that both of these patterns are attractors of the network, but the spurious pattern is a stronger attractor, and in many cases the network converges to it. We decided to replace the original pattern of the digit '4' (figure 7) and re-calculate the weight matrix **W** and validate the network.

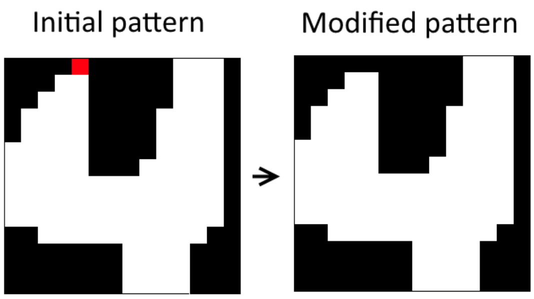



**Figure 7.** Image of original and modified digit '4' patterns.

After changing the pattern, the accuracy of the classification of the digit '4' increased to 15.8% (see table 2), while the accuracy of the classification of the digit '9' fell to 27.2% because of an increase in the similarity of the patterns of the digits '4' and '9'. As a result, the accuracy of the network classification during validation increased to 61%.

Next, we tested the network on a test sample, which consists of 10000 images. The classification accuracy on the test data was 61.5%, it is close to the results of the classification of validation data. The distribution of classification accuracy of individual digits is similar to table 2. The classification accuracy of digits '4' and '9' was 16.9% and 27.6%, respectively.

**Table 2.** Classification results of the MNIST database for single digits after changing the pattern of digit ‘4’.

| Digit | Percent of correct classification, % | Percent of incorrect classification, % | The most common erroneous result | Percent of spurious patterns, % |
|---|---|---|---|---|
| 0 | 68.6 | 5.49 | 6 | 25.9 |
| 1 | 90.8 | 0.35 | 7 | 8.87 |
| 2 | 73.5 | 4.84 | 1 | 21.7 |
| 3 | 66.9 | 8.33 | 2 | 24.8 |
| 4 | 15.8 | 16.3 | 6 | 67.9 |
| 5 | 52.8 | 8.57 | 6 | 38.6 |
| 6 | 80.7 | 3.94 | 1 | 15.4 |
| 7 | 69.5 | 5.89 | 2 | 24.6 |
| 8 | 56.7 | 13.5 | 3 | 29.9 |
| 9 | 27.2 | 31.1 | 7 | 41.7 |

The results indicate this network can be used to recognize handwritten digits, but the obtained classification accuracy is significantly lower compared to other direct distribution neural networks (multilayer, convolutional, etc.) [26]. The accuracy of network classification can be improved by using pre-processing of the input data and by the correct selection of memorized patterns. Spurious patterns introduce a significant error, and the amount of spurious patterns can be minimized using diluted Hopfield networks [32]. In such networks, part of the coefficients in the coupling matrix **W** is reset to zero, and it reduces the probability of the creation of a spurious pattern [33]. A set of networks with fewer recorded patterns can be used, but in this case, a mechanism is needed to resolve situations, when the networks in the set will provide different results. One of such mechanisms may be the winner-takes-all principle, when the result of the network that converges to the pattern most quickly is selected.

## 4. Conclusion

The study demonstrates the possibility of recognizing handwritten digits from the MNIST database using Hopfield networks. The deskewing operation was used as the image pre-processing, which allows reducing the angle of inclination and linear displacement of images. The memorized patterns were calculated by averaging the brightness of pixels for each digit, and then, the patterns were binarized. Using the standard Hebbian training method, the network does not operate correctly because of the strong correlation between memorized patterns. Alternatively applied the Storkey method allows the capacity of the associative memory of the network to increase when writing highly correlated patterns to it. The correlation between the patterns was reduced in two ways: by changing the size of the pattern $M$ due to cropping of external pixels and by changing the binarization threshold $p_{th}$. At the optimal values of these parameters, $M$ = 196 (14x14 pixels) and $p_{th}$ = 86, the network operated correctly after writing all 10 patterns. The accuracy of the network classification on validation data was 56.2%. The use of the optimal binarization threshold of test images $p'_{th}$ = 38 and a small correction of the training pattern for the digit ‘4’ increase the classification accuracy during validation up to 61%. Operating the network on test data, which were not used in the network training and validation, demonstrated the classification accuracy of 61.5%. Although this accuracy value is lower compared to the results of other modern neural networks [26], it can be improved with the help of a diluted Hopfield network or a competitive operation of a set of networks.

## Acknowledgment

This research was supported by the Russian Science Foundation (grant no. 16-19-00135).

The authors express their gratitude to Dr. Andrei Rikkiev for the valuable comments in the course of the article translation and revision.

**References**

[1] McCulloch W S and Pitts W 1943 A logical calculus of the ideas immanent in nervous activity *Bull. Math. Biophys.* **5** 115–33
[2] Abiodun O I, Jantan A, Omolara A E, Dada K V, Mohamed N A E and Arshad H 2018 State-of-the-art in artificial neural network applications: a survey *Heliyon* **4** e00938
[3] Sarkar S T, Bhondekar A P, Macaš M, Kumar R, Kaur R, Sharma A, Gulati A and Kumar A 2015 Towards biological plausibility of electronic noses: a spiking neural network based approach for tea odour classification *Neural Networks* **71** 142–9
[4] Medathati N V K, Neumann H, Masson G S and Kornprobst P 2016 Bio-inspired computer vision: towards a synergistic approach of artificial and biological vision *Comput. Vis. Image Underst.* **150** 1–30
[5] Nassif A B, Shahin I, Attili I, Azzeh M and Shaalan K 2019 Speech recognition using deep neural networks: a systematic review *IEEE Access* **7** 19143–65
[6] Rawat W and Wang Z 2017 Deep convolutional neural networks for image classification: a comprehensive review *Neural Comput.* **29** 2352–449
[7] Liang M and Hu X 2015 Recurrent convolutional neural network for object recognition *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Piscataway: IEEE) pp 3367–75
[8] Tanaka G, Yamane T, Héroux J B, Nakane R, Kanazawa N, Takeda S, Numata H, Nakano D and Hirose A 2019 Recent advances in physical reservoir computing: a review *Neural Networks* **115** 100–23
[9] Yu Y, Si X, Hu C and Zhang J 2019 A review of recurrent neural networks: lstm cells and network architectures *Neural Comput.* **31** 1235–70
[10] Hopfield J J 1982 Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U. S. A.* **79** 2554–8
[11] Csaba G and Porod W 2020 Coupled oscillators for computing: a review and perspective *Appl. Phys. Rev.* **7** 011302
[12] Nikonov D E, Csaba G, Porod W, Shibata T, Voils D, Hammerstrom D, Young I A and Bourianoff G I 2015 Coupled-oscillator associative memory array operation for pattern recognition *IEEE J. Explor. Solid-State Comput. Devices Circuits* **1** 85–93
[13] Velichko A 2019 A method for evaluating chimeric synchronization of coupled oscillators and its application for creating a neural network information converter *Electronics* **8** 756
[14] Corti E, Gotsmann B, Moselund K, Stolichnov I, Ionescu A and Karg S 2019 Resistive coupled $VO_2$ oscillators for image recognition *Proc. IEEE Int. Conf. on Rebooting Computing* (Piscataway: IEEE) pp 1-7
[15] Hoppensteadt F C and Izhikevich E M 2000 Pattern recognition via synchronization in phase-locked loop neural networks *IEEE Trans. Neural Networks* **11** 734–8
[16] Hölzel R W and Kostorz K 2011 Pattern recognition with simple oscillating circuits *New J. Phys.* **13** 73031
[17] Kostorz K, Hölzel R W and Krischer K 2013 Distributed coupling complexity in a weakly coupled oscillatory network with associative properties *New J. Phys.* **15** 83010
[18] Hoppensteadt F C and Izhikevich E M 1999 Oscillatory neurocomputers with dynamic connectivity *Phys. Rev. Lett.* **82** 2983–6
[19] Torrejon J *et al* 2017 Neuromorphic computing with nanoscale spintronic oscillators *Nature* **547** 428–31
[20] Pufall M R, Rippard W H, Csaba G, Nikonov D E, Bourianoff G I and Porod W 2015 Physical implementation of coherently coupled oscillator networks *IEEE J. Explor. Solid-State Comput. Devices Circuits* **1** 76–84

[21] Kumar A and Mohanty P 2017 Autoassociative memory and pattern recognition in micromechanical oscillator network *Sci. Rep.* **7**
[22] Malik P and Dixit R 2014 Handwritten character recognition using wavelet transform and hopfield network *Proc. Int. Conf. on Machine Intelligence Research and Advancement* (Piscataway: IEEE) pp 125–9
[23] Sarangi P K, Sahoo A K and Ahmed P 2012 Recognition of isolated handwritten oriya numerals using hopfield neural network *Int. J. Comput. Appl.* **40** 36–42
[24] Bansal S and Dixit R 2016 Performance enhancement of the pattern recalling efficiency of hopfield neural network using genetic algorithm for cursive handwritten character recognition *Int. J. Appl. Pattern Recognit.* **3** 59
[25] Kumar S and Singh M P 2010 Pattern recalling analysis of english alphabets using hopfield model of feedback neural network with evolutionary searching *Int. J. Bus. Inf. Syst.* **6** 200–18
[26] *MNIST handwritten digit database* Retrieved from: http://yann.lecun.com/exdb/mnist/
[27] Hertz J A, Krogh A S and Palmer R G 1991 *Introduction to the theory of neural computation* (Boston: Addison-Wesley Pub. Co)
[28] Löwe M 1998 On the storage capacity of hopfield models with correlated patterns *Ann. Appl. Probab.* **8** 1216–50
[29] Storkey A and Valabregue R 1997 Hopfield learning rule with high capacity storage of time-correlated patterns *Electron. Lett.* **33** 1803–4
[30] Storkey A 1997 Increasing the capacity of a hopfield network without sacrificing functionality *Lecture Notes in Computer Science* **1327** 451–6
[31] Szandała T 2015 Comparison of different learning algorithms for pattern recognition with hopfield's neural network *Procedia Computer Science* **71** 68–75
[32] Athithan G and Dasgupta C 1997 On the problem of spurious patterns in neural associative memory models *IEEE Trans. Neural Networks* **8** 1483–91
[33] Maffezzoni P, Bahr B, Zhang Z and Daniel L 2016 Analysis and design of boolean associative memories made of resonant oscillator arrays *IEEE Trans. Circuits Syst. I Regul. Pap.* **63** 1964–73