Roman N. KVYETNYY[1], Yuriy Yu. IVANOV[1], Volodymyr V. PIVOSHENKO[1], Yaroslav A. KULYK[1], Bogdan P. KNYSH[1] , Andrzej SMOLARZ[2], Kuanysh MUSLIMOV[3], Yerbol TURGYNBEKOV[4]

Vinnytsia National Technical University (1), Lublin University of Technology (2),
Kazakh National Research Technical Iniversity named after K.I.Satpayev (3), Taraz State University after M.Kh.Dulaty (4)

doi:10.15199/48.2019.04.29

Low computational complexity algorithm for recognition highly corrupted QR codes based on Hamming-Lippmann neural network

Abstract. *This article describes the architecture of the Hamming-Lippmann neural network and the math of the modified learning-recognition algorithm and presents some practical aspects for using it for solving an image recognition task. We have created software using C# programming language, that utilized this network as an additional error-correcting procedure, and have solved the task of recognition highly corrupted QR codes (with a connection to the database). Experimental results, of finding the optimal parameters for this algorithm, are presented. This neural network doesn't require time-consuming computational procedures and large amounts of memory, even for high-resolution and big size images.*

Streszczenie. *W tym artykule opisano architekturę sieci neuronowej Hamminga-Lippmanna oraz matematykę zmodyfikowanego algorytmu rozpoznawania uczenia się, a także przedstawiono kilka praktycznych aspektów korzystania z niej w celu rozwiązania zadania rozpoznawania obrazu. Stworzyliśmy oprogramowanie wykorzystujące język programowania C #, który wykorzystał tę sieć jako dodatkową procedurę korekty błędów i rozwiązaliśmy zadanie rozpoznawania wysoce uszkodzonych kodów QR (w połączeniu z bazą danych). Przedstawiono wyniki eksperymentalne poszukiwania optymalnych parametrów dla tego algorytmu. Opisywana neuronowa nie wymaga czasochłonnych procedur obliczeniowych i dużej ilości pamięci, nawet w przypadku obrazów o wysokiej rozdzielczości i dużych rozmiarach. (**Algorytm o niskiej złożoności obliczeniowej do rozpoznawania wysoce uszkodzonych kodów QR w oparciu o sieć neuronową Hamminga-Lippmanna**).*

Keywords: neuron, Hamming-Lippmann neural network, learning-recognition algorithm, image processing, sliding window mode, computational complexity, image recognition, error-correction, QR codes.
Słowa kluczowe: neuron, sieć neuronowa Hamminga-Lippmanna, algorytm rozpoznawania, przetwarzanie obrazu, przesuwne okno, złożoność obliczeniowa, rozpoznawanie obrazu, korekcja błędów, kody QR, optymalne parametry.

Introduction

The beginning of the neural networks history is connected with the "connectionist models" or "parallel distributed processing", that are considered in the publication in 1943 of McCulloch and Pitts. These scientists proposed general approaches and specific mathematical models of the biological neural networks and their components – neurons, that became fundamental in the artificial neural networks theory. These neurons were presented as models of the biological neurons and as conceptual components for circuits, that performed computational tasks. They used threshold elements with two stable states, which are called "McCallock-Pitts neurons" [1]. The task of developing models and systems, which are based on the threshold elements was so unusual and complex, that only in 1956 was appeared the first capable artificial neural network – Rosenblatt perceptron. Its demonstrated the possibility of creating technical patterns, based on the models of the human brain, for image recognition. However, further researches of perceptrons showed, that their usage in the recognition systems is associated with many difficulties. In 1969 Minsky and Papert published their book, in which they described the deficiencies of the perceptron model and showed their fundamental nature. The negative prognosis of the authoritative scientists caused a decline in the interest in neural networks, which lasted more than ten years, but in the 80's after some important theoretical results, the neural networks began to rebound. The renewed interest is reflected in many researches, the amounts of funding, the number of conferences and journals associated with neural networks [2,3]. At the same time, neurocomputing begins to develop, which allows solving problems from different areas of knowledge using neural networks, which are modelled on ordinary computers [4].

The machine interpretation of the neural network came into the world of Computer Science directly from the biology, where neural networks are able not only to analyze incoming information, but also to reproduce it from its memory. The main connections to the biology are stated in that approximately 86 billion neurons (computational units), which transmit information in the form of electrical impulses, can be found in the human nervous system and they are connected with approximately 1014 - 1015 synapses [5]. The definition of the artificial neural network is provided by Hecht-Nielsen, which is the inventor of one of the first neurocomputers. He defines the neural network as: "...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs" [6]. In general neural network architecture consists of some number of neurons (nodes, units), their connections (synapses) with some weights and layers between the neurons. The activation (transfer) function of the node defines the output of that node, given an input or set of inputs from other neurons. One of the main network part is the learning (training) algorithm or rule, which modifies the parameters of the neural network, in order for a given input to the network to produce a favored output. The learning process typically amounts to modifying the weights and thresholds [7,8]. The Fig. 1 shows the drawing of the biological neuron (left) and the common mathematical model (right).

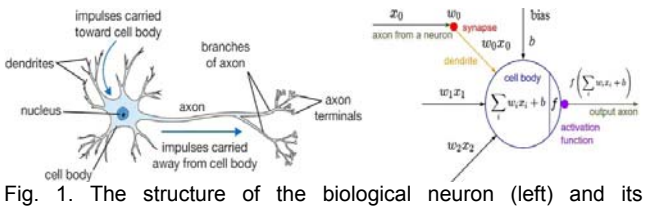

Fig. 1. The structure of the biological neuron (left) and its mathematical model (right)

Neural networks can simulate the function of any complexity, therefore, they currently provide the best solutions to many problems in the classification and

clustering, recognition, prediction (approximations and extrapolation tasks), optimization, etc. For example, doctors use neural networks to diagnose and predict treatment; customs officers – to detect prohibited things; meteorologists – for weather prediction; software specialists – to combat with computer viruses; bankers – to assess credit risks, bonds, stocks and currencies, to provide security for plastic card transactions, for handwritten signatures on the checks; industrialists – to control the quality of products and packaging, for the management of nuclear power plants; military – for the purpose of recognition and support, combat operations, piloting of damaged aircraft, information encoding and decoding, etc. But the solution of these tasks requires considerable computational resources and significant time costs [5,9].

So, in this article we work with modified Hamming-Lippmann neural (HLN) network, which allows to solve recognition task of highly corrupted images (for example QR or quick response codes with the connection to the internet or user database) without requiring time-consuming computational procedures and large amounts of memory. The work objective is an experimental research of the created neural network and finding some optimal parameters.

The rest of the article is structured as follows. Section II describes the architecture of the neural network and modified learning algorithm. In Section III we present some experimental results for processing images with QR codes. At last, the paper is concluded in Section IV.

Hamming-Lippmann neural network architecture

The HLN network was proposed in 1987 by R. Lippmann [10]. It is a relaxation multilayered network with feedbacks between individual layers, which recognizes and classifies the images. This network is used as an associative memory. In essence, it consists of two subnets ─ Hamming and MaxNet or Hopfield net. The neural network implements some classifier, based on the Hamming's distance [11] used in this network as a similarity measure of the input and reference or learning images. MaxNet is motivated by the large numbers of connections in biological neural nets and by interconnected networks described by Kohonen. Although this network is similar in structure to the Hopfield net, it uses threshold-logic nodes instead of hard-limit nodes and feeds the output of each node back to its input instead of disallowing this feedback path. Weight values and thresholds of HLN network are determined from the task conditions, therefore such network is the artificial neural network with fixed connections [12]. The key feature is the fact, that the network outputs isn't the value of the reference vector, which is most similar to the input, but the number of the corresponding learning sample from MaxNet by the "WTA (winner takes all) principle" [5]. Thus only one neuron-winner remains active at the output of the all net. HLN network does not require hard computational procedures for its training. A notable disadvantage of this network: it does not allocate two or more reference images, that have the same maximum similarity measures [9,13]. The architecture of the HLN network is shown in Fig. 2.

The HLN network consist of m input (S), n hidden (Z, A), n output (Y) neurons and 2 layers. The value of n is equal to the number of reference images, stored by the network and presented as m-dimensional bipolar vectors $S = (s_1,\ldots,s_m)$. Also this network use bias values b. The weight matrix W of dimension $m\times n$ contains the values of the synaptic connections of the direct propagation network, the matrix E of $n\times n$ dimension stores the values of the synaptic connections of MaxNet network neurons. The activation function of neurons is threshold.

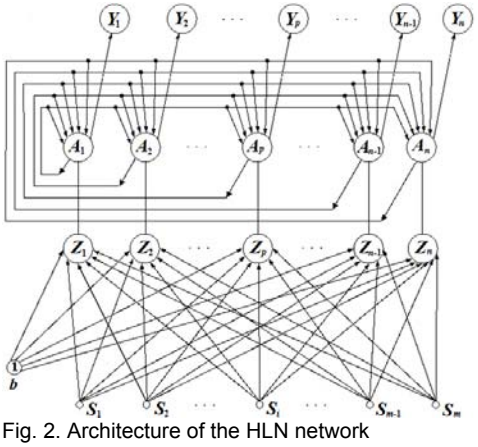

Fig. 2. Architecture of the HLN network

Image recognition using Hamming-Lippmann network

Task. There is an initial reference set of images (database), designed for network's training. It is necessary to make a comparison of the noisy image, submitted to the input of the neural network, with all reference patterns and to conclude, that it recognized as one of the samples, or that it can't be recognized exactly.

Image processing and learning-recognition algorithm. The process of training HLN network consists of two parts (image processing [9,14] and training-recognition [15-18]) and includes the following sequence of steps:

1. Analyse pixel matrix P with size H×B, where H and B are the height and width of the image respectively, by counting the number of dark pixels. In order to determine whether the analysed pixel is dark, it is necessary to make the transformation into a grayscale by the formula [14]:

(1) $Y = 0{,}299\cdot R + 0{,}587\cdot G + 0{,}114\cdot B$,

where values R, G, B are the corresponding colour components.

2. Choose a certain boundary value L for colour.

3. Compare Y with L. If $0 \le Y \le L$, then the corresponding pixel is considered as dark, otherwise ─ white.

4. Write the received binary matrix of the image with size m = H×B. Thus, for all reference images n, it is possible to determine the corresponding binary vectors, that will be used later for network learning and recognition of unknown samples.

5. Choose a value K of the sliding window size. This value is equal to the number of rows or columns, that will be processed (Fig. 3). Changing the window size K, it will be possible to change the accuracy of the image analysis.

6. Choose a threshold value Q.

7. Process K rows and calculate Ni ─ a number of black pixels. If $Ni \ge Q$, then the i-th element of the binary vector Si is assigned as "+1", otherwise as "-1". This step is repeated until all rows are processed. Now we have some vector S with m elements for our input neurons.

8. Perform the learning process for the network, using steps 1-7 and database with reference images (QR codes in our case).

9. On the base of step 8 (for all images) create the weight values matrix W with the size

(2) $M = m\times n = \lceil X / K \rceil \times n$,

where X is the height H or width B of the image.

So, this modification will greatly reduce the input vector length (especially for high-resolution and big sized images),

and it will simplify the computational complexity of the network.

10. Choose some image for the recognition and perform image processing with the usage of steps 1-7.

11. The output signal from the input S neurons is equal to

(3) $U_{output\,S_i} = f_S(U_{input\,S_i}) = S_i$.

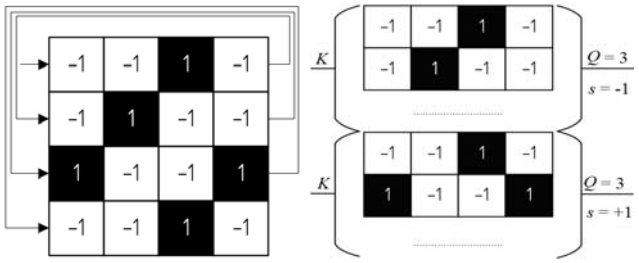

Fig. 3. Image processing using standard row by row (left) and sliding window (right) modes

12. Upon presentation of the input vector, each of Z neurons from 1 to n calculates its input signal in accordance with an expression of the form

(4) $U_{input\,Z_k} = b + \sum_{i=1}^{m} w_{ik} \cdot U_{output\,S_i} = \frac{m}{2} + \frac{1}{2} \cdot \sum_{i=1}^{m} w_{ik} \cdot s_i$,

where b — the bias value; wik — weight value from i input neuron to k neuron Z (it is bipolar value +1 or -1).

13. Determine the mode of the activation function for hidden Z neurons, using next relation

(5) $U_{output\,Z_k} = f_Z(U_{input\,Z_k}) = \begin{cases} 0, & if \quad U_{input\,Z_k} \le 0; \\ v \cdot U_{input\,Z_k}, & if \quad 0 < U_{input\,Z_k} \le U; \\ U, & if \quad U_{input\,Z_k} > U, \end{cases}$

where v — normalization parameter, U = 1/v — constant.

14. The output signal from Z neurons is the input signal of the subnet MaxNet. This is equal to the equation

(6) $U_{input\,A_k} = U_{output\,Z_k}$.

15. This subnet starts the iterative process of extracting the maximum output signal in the next form

$$U_{output\,A_k}(0) = U_{output\,Z_k},$$

(7) $$U_{input_{A_k}}(t+1) = \sum_{i=1}^{n} w_{ik} \cdot U_{output_{A_i}}(t) = U_{output_{A_k}}(t) - \varepsilon \cdot \sum_{i=1, i \ne k}^{n} U_{output\,A_i}(t),$$

where t and (t+1) is previous and next iteration, 0 < ε ≤ 1, and wik = 1 for MaxNet and wik = 0 for Hopfield net, if i = k, and -ε in otherwise.

16. Determine the mode of the activation function for hidden A neurons, using next relation

(8) $U_{output_{A_k}}(t) = f_A(U_{input_{A_k}}(t)) = \begin{cases} U_{input_{A_k}}(t), & if \quad U_{input_{A_k}}(t) > 0; \\ 0, & if \quad U_{input_{A_k}}(t) \le 0. \end{cases}$

17. The iterative process (steps 15 and 16) is repeated until the norm of two vectors is more than some small value eps1. So, the stabilization condition is

(9) $$\| U_{output_{A_k}}(t+1) - U_{output_{A_k}}(t) \| = \sqrt{\sum_{k=1}^{n} (U_{output_{A_k}}(t+1) - U_{output_{A_k}}(t))^2} \le eps_1 .$$

18. The output signals of A neurons enter as the inputs to Y neurons

(10) $U_{input\,Y_k} = U_{output\,A_k}$.

19. Determine the mode of the activation function for output Y neurons, using some small value eps2 (for example 10-2-10-5) or zero in ideal case. This can be represented in the next formula

(11) $U_{output\,Y_k} = f_Y(U_{input\,Y_k}) = \begin{cases} 1, & if \quad U_{input\,Y_k} > eps_2; \\ 0, & if \quad U_{input\,Y_k} \le eps_2, \end{cases}$

or, if there are many iterations are needed, simply choose the max value of UinputY.

20. The positive element index $k = \overline{1,n}$ of the output neurons signal UoutputY will indicate the number of the reference image, that best corresponds to the input test signal. This value is the desired result.

The output of the network in the ideal case should be the vector with one positive and all other zero elements. But, as often happens, the input sample can be very noisy, and several positive values may appear at the network output. In this case, it is usually believed, that network could not uniquely classify the input image [13]. In the next section of the article, we will show the operability of the presented algorithm in practice [22,23,24].

Experimental results for recognition highly corrupted QR codes

It should be noted, that the rapid growth of the information has led to the creation of combined compression systems, noise protection and information storage. These systems include matrix codes or two-dimensional bar codes, such as Data Matrix, Aztec and the most popular QR codes. The main advantage of QR code is the easy recognition of scanning equipment (for example, a smartphone camera) and large information capacity (7089 digits, 4296 letters), which makes it possible to use this code in trading, manufacturing, logistics, etc. Now QR code is found almost everywhere, for example, according to the web-portal comScore, more than 20 million USA citizens use smartphones to scan this code; in Japan it is applied to all booklets, handbooks and various goods [19]. The fundamental math for encoding and decoding of this code is presented in standard ISO 18004[20].

The main purpose of this research was to find the optimal value of the parameters K and Q, using which the HLN network with the sliding window mode correctly recognizes QR codes. For finding the decision of this task on the programming language C# [21] has been created the universal software (Fig. 4). To determine these values, we carried out some experiments, in which have been used 3 sizes of images (200×200, 512×512 and 1024×1024) with high level of noise, lack of parts or colour fusion.

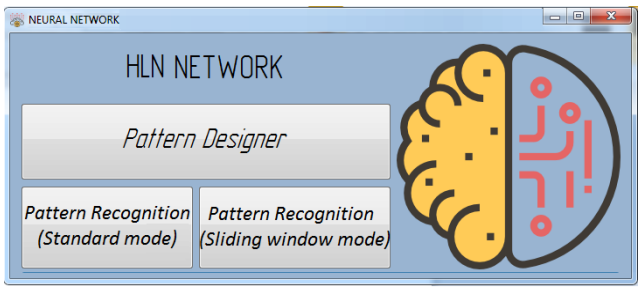

Fig. 4. General software GUI

Example. In Fig 5 and 6 are presented 5 training and test images with highly corrupted QR codes. These images are not recognized by the decoding software, since the structure of element detectors is corrupted or error-correcting ability of the Reed-Solomon codes does not allow to correct necessary part of errors. So, we propose to apply an additional error-correcting procedure, using HLN net and images database.

Fig. 5. Training samples in the database

Fig. 6. Test samples with highly corrupted QR codes

The experimental results of the recognition test QR codes from Fig. 6 with different image sizes are presented in Tables 1-3. We use linear dependence $Q = \alpha \cdot K + \beta$ for threshold value Q and sliding window size K, and experimentally chose the values of constants α and β as [α; β] = [(50; 0), (200; 0), (400; 100)].

In Table 4 is demonstrated the fact, that, when the threshold Q smaller than the value of sliding window size K or equal to it, the HLN network cannot correctly perform the recognition.

Table 1. The results of the experiment with linear dependence $Q = 50 \cdot K$ for images with 200×200 size.

K	Q	1	2	3	4	5	Number of successful recognition	Result
1	50	T	T	T	T	T	5	Good
2	100	T	T	T	T	T	5	Good
3	150	T	T	T	T	T	5	Good
4	200	T	T	T	T	T	5	Good
5	250	T	T	T	T	T	5	Good
6	300	T	T	T	T	T	5	Good
7	350	T	T	T	T	T	5	Good
8	400	T	T	T	T	T	5	Good
9	450	T	F	T	T	T	4	Bad, tuning
10	500	T	T	T	T	F	4	Bad, tuning
11	550	T	F	F	F	F	1	Bad
12	600	T	T	F	T	F	3	Bad
13	650	T	F	F	F	F	1	Bad
14	700	T	F	F	F	F	1	Bad
15	750	F	T	F	F	F	1	Bad
16	800	T	F	F	F	F	1	Bad

Table 2. The results of the experiment with linear dependence $Q = 200 \cdot K$ for images with 512×512 size.

K	Q	1	2	3	4	5	Number of successful recognition	Result
9-15	1800-3000	T	T	T	T	T	5	Good
16	3200	T	T	T	T	T	5	Good
17	3400	T	F	T	T	T	4	Bad, tuning
18	3600	T	F	T	T	T	4	Bad, tuning
19	3800	F	T	T	T	T	4	Bad, tuning
20	4000	T	F	T	T	T	4	Bad, tuning
21	4200	T	F	F	T	T	3	Bad
22	4400	T	F	T	T	T	4	Bad, tuning
23	4600	T	F	T	T	T	4	Bad, tuning
24	4800	T	T	F	T	T	4	Bad, tuning
25	5000	T	F	T	F	T	3	Bad
26	5200	T	F	T	T	T	4	Bad, tuning
27	5400	F	T	T	F	F	2	Bad
28	5600	F	T	T	F	F	2	Bad

As can be seen from Tables 1-3, the most optimal values of coefficients K and Q for test images with sizes H×B are [H×B; K; Q] = [(200×200; 8; 400), (512×512; 16; 3200), (1024×1024; 26; 10500)]. Using these values, the HLN network correctly recognizes all test QR codes. But in order to achieve lower computational complexity and greater accuracy, it can be performed some tuning procedure for parameter Kt, when (n-1) test samples was recognized, for example [H×B; {Kt}] = [(200×200; {9; 10}), (512×512; {17-20; 22-24; 26}), (1024×1024; {27, 28})]. In Table 5 we show the example of the tuning procedure, based on the intervals from Table 1, for sliding window size $K = 9$ (images 200×200), which is bigger, than optimal value, using which all QR codes are recognized successfully.

Table 3. The results of the experiment with linear dependence $Q = 400 \cdot K + 100$ for images with 1024×1024 size.

K	Q	1	2	3	4	5	Number of successful recognition	Result
1-25	500-10100	T	T	T	T	T	5	Good
26	10500	T	T	T	T	T	5	Good
27	10900	T	T	T	T	F	4	Bad, tuning
28	11300	T	T	T	T	F	4	Bad, tuning

Table 4. The results of the experiment for images with 200×200 size, $K \geq Q$.

K	Q	1	2	3	4	5	Number of successful recognition	Result
1	1	T	T	F	F	F	2	Bad
4	2	T	T	F	F	F	2	Bad
4	4	T	T	F	F	F	2	Bad
9	4	T	F	F	F	F	1	Bad
9	9	T	F	F	F	F	1	Bad
16	8	T	T	F	F	F	2	Bad
16	16	T	F	F	F	F	1	Bad
25	12	T	F	F	F	F	1	Bad
25	25	T	F	F	F	F	2	Bad
36	18	T	F	F	F	F	1	Bad
36	36	T	F	F	F	F	1	Bad
49	24	T	F	F	F	F	1	Bad
64	32	T	F	F	F	F	1	Bad
81	40	T	F	F	F	F	1	Bad
100	50	T	F	F	F	F	1	Bad

Table 5. The results of the tuning procedure in software for images with 200×200 size, $K = 9$.

Q	1	2	3	4	5	Number of successful recognition	Result	Optimal value
400	T	T	T	F	F	3	Bad	$K = K_t - 1 = 8$
455	T	F	T	T	F	4	Bad	
460	T	T	T	T	F	4	Bad	
465	T	F	T	T	F	3	Bad	
470	T	T	T	T	F	4	Bad	
475	F	T	T	T	F	3	Bad	
480	T	T	T	T	F	4	Bad	
485	T	T	F	T	F	3	Bad	
490	T	T	F	T	F	3	Bad	
495	T	T	F	T	F	3	Bad	
500	T	T	T	T	F	4	Bad	
510	T	T	F	T	F	3	Bad	

So, the results of these experiments are shown the high efficiency of the presented algorithm. It has good performance to use it in distributed computer systems as an additional error-correcting procedure for highly corrupted QR codes.

Conclusions

The design of the artificial neural networks requires from the neural network architect the high knowledge of computer science, empirical experience and the ability to

work with different programming languages. In this paper we presented theoretical information about the history and architecture of the artificial neural networks. We have created the architecture of HLN network, which can recognize different types of multimedia data, for example images with simple objects, such as signatures, QR codes, etc. An actual task of highly corrupted QR codes recognition has been performed with the usage of HLN network with modified learning-recognition algorithm and image database. So, we presented some additional error-correcting procedure, that doesn't require hard computational efforts, a lot of time and large amounts of memory. Also, the experimental research of the optimal parameters for sliding window size K and threshold value Q has been performed for different image sizes. The results can be used for further hardware-software implementation in distributed computer systems of different functional purposes.

***Authors**: D.Sc., Professor of Automatics and Information-Measuring Techniques Department Roman N. Kvyetnyy, Vinnytsia National Technical University, Khmelnytsky Hwy, 95, 21021 Vinnytsia, Ukraine, e-mail: rkvetny@sprava.net; Ph.D. Yuriy Yu. Ivanov, Vinnytsia National Technical University, Khmelnytsky Hwy, 95, 21021 Vinnytsia, Ukraine, e-mail: Yura881990@i.ua; Student of Automatics and Information-Measuring Techniques Department Volodymyr V. Pivoshenko, Vinnytsia National Technical University, Khmelnytsky Hwy, 95, 21021 Vinnytsia, Ukraine, e-mail: volodymyr.pivoshenko@gmail.com; Ph.D. Yaroslav A. Kulyk, Vinnytsia National Technical University, Khmelnytsky Hwy, 95, 21021 Vinnytsia, Ukraine, e-mail: Yaroslav_Kulik@i.ua; Ph.D. Bogdan P. Knysh, Vinnytsia National Technical University, Khmelnytsky Hwy, 95, 21021 Vinnytsia, Ukraine.*

REFERENCES

[1] McCulloch W., Pitts W., A Logical Calculus of Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, 5 (1943), 115-133
[2] Smagt van der P., Krose, B., An Introduction to Neural Networks, *Netherlands, University of Amsterdam, Faculty of Mathematics and Computer Science* (1996), 15-19.
[3] Haykin S., Neural Networks: A Comprehensive Foundation, *New Jersey, Prentice Hall*, (1994), 89-97
[4] Norvig P., Russel S., Artificial Intelligence, *New Jersey, Prentice Hall*, (2009), 863-902
[5] Król K., Inverse problem solution using neural network, *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska - IAPGOS*, 2(2012), no. 1., 9-11
[6] Caudill, M., Neural Networks Primer, Part I, *AI Expert*, 2(1987), 46-52
[7] Goyal S., Goyal G.K., Advanced Computing Research on Cascade Single and Double Hidden Layers for Detecting Shelf Life of Kalakand: An Artificial Neural Network Approach, *International Journal of Computer Science and Emerging Technologies*, 2(2011), 292-295
[8] Mason J., Ellacott S.W., Mathematics of Neural Networks, *Springer*, (1997), 3-15
[9] Wójcik W., Kotyra A., Golec T. et al., Vision based monitoring of coal flames, *Przegląd Elektrotechniczny*, 87 (2008), n.3, 241-243
[10] Lippmann, R., An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, 4(1987), 4-22
[11] Morelos-Zaragoza R., The Art of Error Correction Coding, *John Wiley&Sons*, (2006), 27-39
[12] Lippmann R., Gold B., Malp M.L., A Comparison of Hamming and Hopfield Neural Nets for Pattern Classification, Cambridge, Massachusetts Institute of Technology, Technical Report 769, 41 p. (1987)
[13] Koutroumbas K., Kalouptsidis N., Generalized Hamming Networks and Applications, Neural Networks, 18(2005), 896-913
[14] Gonzalez, R.C., Woods, R.E., Digital Image Processing, *Prentice-Hall*, (2008), 206-253
[15] Fausett L., Fundamentals of Neural Networks: Architectures, Algorithms, and Applications, *Prentice-Hall*, (1994), 158-169
[16] Callan R., The Essence of Neural Networks, *Prentice Hall*, (1999), 108-116
[17] Luger G., Artificial Intelligence: Structures and Strategies for Complex Problem Solving, *Addison Wesley*, 29-38 (2005).
[18] Rojas R., *Neural Networks*, *Springer Science & Business Media*, (1996), 23-27
[19] Stolarczuk P.; Yatsuk V.; Majewski J., Małaczywski P., Michalewa M. Usage of capacitive sensors for fast checking of parameters of multicomponent solutions. *Przegląd Elektrotechniczny*, 86 (2010), No. 10, 92-95.
[20] Toliupa S., Kravchenko Y., Trush A., Organization of implementation of ubiquitous sensor networks, *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska - IAPGOS*, 8(2018), no. 1., 36-39
[21] Heaton J., Introduction to Neural Networks for C#, *Heaton Research*, (2008), 39-137
[22] Wójcik W., Smolarz A., Information Technology in Medical Diagnostics, *Taylor &Francis Group CRC Press Reference*, (2017), 210
[23] Vassilenko V., Valtchev S., Teixeira J.P., Pavlov S., Energy harvesting: an interesting topic for education programs in engineering specialities, *Internet, Education, Science*, (2016), 149-156
[24] Krak Yu.V., Dynamics of manipulation robots: Numerical-analytical method of formation and investigation of computational complexity, *Journal of Automation and Information Sciences*, 31(1999), no. 1-3, 121-128