Bull. Nov. Comp. Center, Comp. Science, 33 (2012), 85–95

Investigation of a sigmoid neural network based on its state visualization

M.S. Tarkov, O.A. Kozhushko

Abstract. Based on a scatterogram of projections onto a plane of states of the output and hidden layers of a sigmoid neural network, its behavior in solving the problem of image recognition of letters is studied. In particular, it was found that the method of conjugate gradients for the entire training process affects both the weights of a hidden layer and the output layer, while the steepest descent method determines the weights of a hidden layer on the first iteration, and in the course of further education has practically no effect on them. An estimate for the number of neurons in the hidden layer that is sufficient for a quality solution of the problem of recognition is obtained. Visualization of the states of neurons is a powerful visual tool for developing and studying the behavior of different classes of neural networks.

1. Introduction

Artificial neural networks (NNs) [1–3] are used for solving a wide set of problems, such as pattern recognition, approximation, interpolation, data compression, prediction, identification, control and association. The most important NN properties are: parallel data processing by all neurons, learning and knowledge of the generalization abilities. Nevertheless, the NN solutions do not give an accurate description of the solution technique. The neural network looks like "black box", and we have no possibility to look into it. Much used numerical methods of estimating the NN operation (mean-square error, evaluation of correspondence of network solution to analytical solution, estimation of probability of obtaining a correct solution, etc.) are not clear. It is sufficiently difficult to analyze them. Graphical evaluation methods are more efficient from the practical viewpoint. In papers [4, 5], Wlodzislav Duch proposed algorithms for visualization of neuron states. These algorithms are based on projecting points of a multidimensional space onto a two-dimensional plane. The algorithm results in a scatter plot known as a scatterogram demonstrating the quality of the NN (neural network) solution of the problem. In this paper, we consider a classic sigmoid neural network. Our objective is the usage of scatterograms for investigation of the network training for solving an image recognition problem.

2. Visualization of neuron states

A problem of the NN state visualization is reduced to construction of mapping multidimensional state vectors (output signals) of its hidden and out-

put layers onto a two-dimensional plane. The plane points, corresponding to multidimensional vectors, form a scatter plot, i.e., a scatterogram, on which we can distinguish clusters of vector images to receive information for visual estimation of the NN functioning efficiency.

The output neuron state visualization must save partitioning the output vectors onto clusters according to the NN solution. In [4], a linear transform satisfying this property is proposed. The transform maps a m-dimensional vector O onto a two-dimensional vector so that m vectors of the type $(0,\dots,0,1,0,\dots,0)$ and the vector $(1,1,\dots,1)$ are mapped onto the points (x_i,y_i), $i=1,\dots,m$, and (x_c,y_c), respectively, where

$$x_i=\frac{1}{2}+r\cos\Big(\varphi+\frac{2\pi i}{m}\Big),\qquad y_i=\frac{1}{2}\tan\Big(\frac{\pi}{2}-\frac{\pi}{m}\Big)+r\sin\Big(\varphi+\frac{2\pi i}{m}\Big),$$
$$x_c=\frac{1}{2},\qquad y_c=\frac{1}{2}\tan\Big(\frac{\pi}{2}-\frac{\pi}{m}\Big),$$
$$\varphi=-\frac{\pi}{2}-\frac{\pi}{m},\qquad r=\frac{1}{2\cos(\frac{\pi}{2}-\frac{\pi}{m})}.$$

The linear map has the form

$$\begin{pmatrix}x\\y\end{pmatrix}=AO+B, \tag{1}$$

where

$$B=\begin{pmatrix}b_1\\b_2\end{pmatrix},\qquad A=\begin{pmatrix}a_{11}&\dots&a_{1m}\\a_{21}&\dots&a_{2m}\end{pmatrix},$$
$$b_1=\frac{1}{m-1}\sum_{i=1}^{m}x_i-x_c,\qquad b_2=\frac{1}{m-1}\sum_{i=1}^{m}y_i-y_c,$$
$$a_{1i}=x_i-b_1,\qquad a_{2i}=y_i-b_2,\qquad i=1,\dots,m.$$

The unknown transform coefficients can be found by substitution of vectors of the type $(0,\dots,0,1,0,\dots,0)$ and the vector $(1,1,\dots,1)$ into system (1) and solving this system.

The algorithm makes possible to map nodes of m-dimensional cube corresponding to classes onto nodes of a regular m-gon (Figure 1). The clusters of images of correctly classified vectors are mapped onto the neighborhoods of the corresponding nodes of the m-gon. The points near to the center of the m-gon are interpreted as those corresponding to non-recognized patterns with the network output close to $(0,\dots,0)$ (there are no class including a pattern) or $(1,\dots,1)$ (all classes include the pattern).

Similarly, we can consider the points into which become vectors with multiple coordinates equal to one. The vector images located around these points are interpreted as those assigned by the network to several classes simultaneously. A curious fact is that the points symmetric with respect to

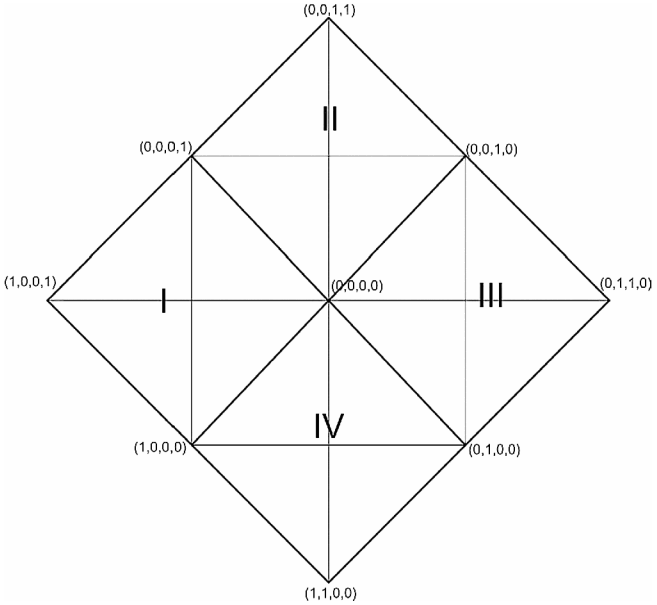

Figure 1. Characteristic points and lines for a four-dimensional case [4]

the image of point $(1,\dots,1)$ can be obtained one from another by inversion of components of the corresponding m-dimensional vectors. Segments connecting the images of vectors with components of zeros and ones with the image of the unit vector $(1,\dots,1)$, as well as the segments forming the m-gon, whose vertices are images of the vectors $(0,\dots,0,1,0,\dots,0)$, are called the characteristic lines of the m-gon obtained for the problem of separation of m-classes (see Figure 1).

To estimate the average scatter of points with respect to m given centers, a function is introduced similar to the sample variance:

$$E(P) = \frac{1}{K}\sum_{i=1}^{K} \min_{j=1,\dots,m} \rho_H(P_i, O_j), \qquad \rho_H(x,y) = \frac{\rho(x,y)}{R},$$

where P_i are points, O_j are centers, $\rho(x,y)$ is a classical Euclidean distance between two points, R is a maximum distance from the center to a certain midpoint. To estimate the scatter of points in m-dimensional space, the following data are used: P_i are output vectors, O_j are unit vertices of the m-dimensional cube, R is the Euclidean distance from the unit vertices of the cube to the zero point. We will consider the scatter of points in the m-dimensional space with the scatter of their projections on the two-dimensional plane (with a scatterogram).

The values of the function $E(P)$ belong to the segment $[0,1]$, and the value 0 corresponds to the exact classification (i.e., the output vectors have

the form $(0, 0, \ldots, 1, \ldots, 0)$), and value 1 corresponds to the absence of classification (i.e., all the output vectors are given by $(1, \ldots, 1)$). Qualitatively, the value of the function $E(P)$ can be described as a mean relative distance from the scatterogram points to the m-gon vertices. Since the value $E(P)$ is averaged over all vectors, its use can not completely replace the use of a scatterogram due to the loss of information. However, this function allows us to evaluate the correctness of the transformation, that is, to compare the scatter of points in an m-dimensional cube, and the scatter in a two-dimensional plane. It can also be used to estimate the average deviation of the network solution from the expected one.

The task of visualizing the state of neurons of the hidden layer is analogously reduced to the choice of converting the k-dimensional vector to a two-dimensional one, but the approach to the selection of a mapping is different. To visualize the output signals of neurons of the hidden layer, we use the orthogonal projection of the signal vectors onto the plane spanned by a pair of vectors $\{e_1, e_2\}$ forming an orthonormal basis. In [5], it is proposed to determine the direction e_1 as $(1, 1, \ldots, 1)^T$, and to compute the direction e_2 by applying the principal component analysis [6] to a set of projections of the vertices of a k-dimensional cube onto the subspace orthogonal to the vector e_1. In case $k = 4$, the basis has the form $e_1 = (1, 1, 1, 1)^T$ and $e_2 = (-0.2113, 0.7889, -0.5774, 0)^T$. The choice as e_1 of any other vector with coordinates equal to 1 or -1, gives a similar division of vertices on the plane. This visualization aims at studying the behavior of neurons in the hidden layer, and at an attempt to slightly open the "black box" of a sigmoidal neural network to gain some understanding of how it works.

3. The study of neuron states in the sigmoid network output layer by solving the image recognition problem

The visualization algorithm is carried out in an environment using MATLAB and the library NETLAB [7] including a set of models of neural networks and algorithms for the neural network training and applications more extensive than in the Neural Network Toolbox.

Experiments for the problem of recognizing monochrome images of letters A, B, D, R with different fonts were carried out (Figure 2). The choice in favor of these letters has been made for the following reasons: the letters B and D for some fonts have similar images, significantly different from the picture of the letter A. Therefore, it is expected that the network results in a clear separation of the class of the letter A and a partial mixing of classes of the letters B and D. The letter R is similar to the other three letters, therefore, the location of the output image vectors is supposed to be around the point of defining the class of R and the point of determining unrecognized vectors. As a test, two different samples are taken: one of them is obtained

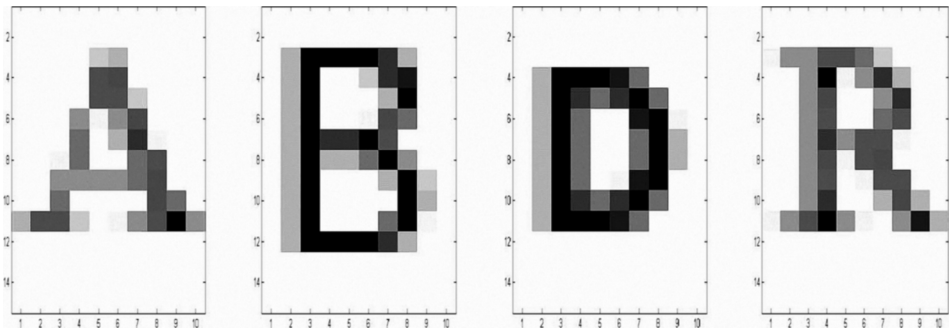

Figure 2. Examples of images of Latin letters

by mixing training sample with noise, therefore, a good separation of the classes is expected, the other is obtained using other fonts for the letters A, B, D, R, the expected separation of the classes being worse.

Each monochrome image can be represented by a vector of length $M \times N$, M and N are, respectively, the length and the width of the image, whose pixel values lie in the range $[0, 1]$. Images will be viewed as vectors in the Euclidean space. The similarity of the images X and Y will be evaluated as the function

$$H(X, Y) = \frac{(X, Y)}{|X||Y|},$$

where (X, Y) is a scalar product, $|X| = (X, X)^{1/2}$. The function takes a maximum value 1, when the images coincide. For two different letters, $H(X, Y) \in [0.3, 0.6]$.

The images used have the size of 10×15 pixels. Respectively, the number of neurons in the input layer of the sigmoidal network is 150. The number of neurons in the output layer (the number of classes) is 4. The number of neurons in the hidden layer is chosen to be 10 in the experiments without the number variation. As the main training method the method of conjugate gradients was used. The research allowed us to draw the conclusion about the choice of the NN training algorithm and its dynamics.

3.1. Dynamics of neural network training. When training the neural network by the conjugate gradient algorithm, a gradual refinement of weights occurs, which is reflected in the scatterogram. After initializing the network with small values of weights, the scatterogram is chaotic (Figure 3), but after 40 iterations of training there was obtained some separation of data into classes (clusters)(Figure 4).

In the subsequent training, the scatterogram points contract to vertices of the square, which indicates to a higher quality of image classification. It should be noted that the division into classes irregularly occurs. The first class separated in the scatterogram corresponds to the letter A ('$\circ$' symbol),

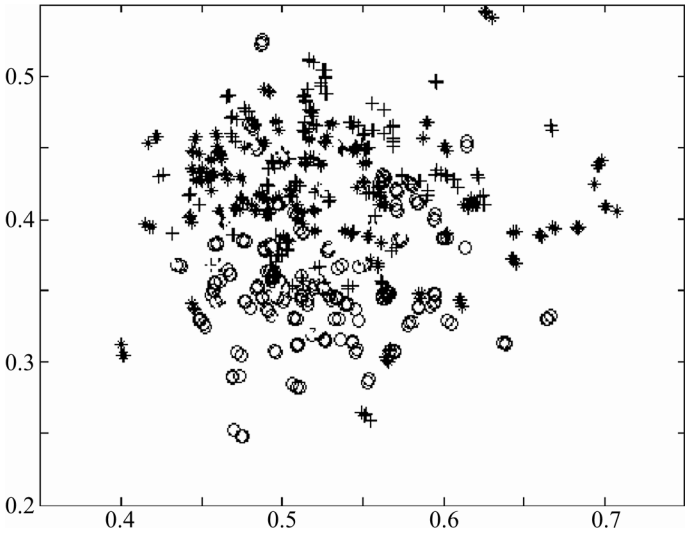

Figure 3. Initial scatterogram

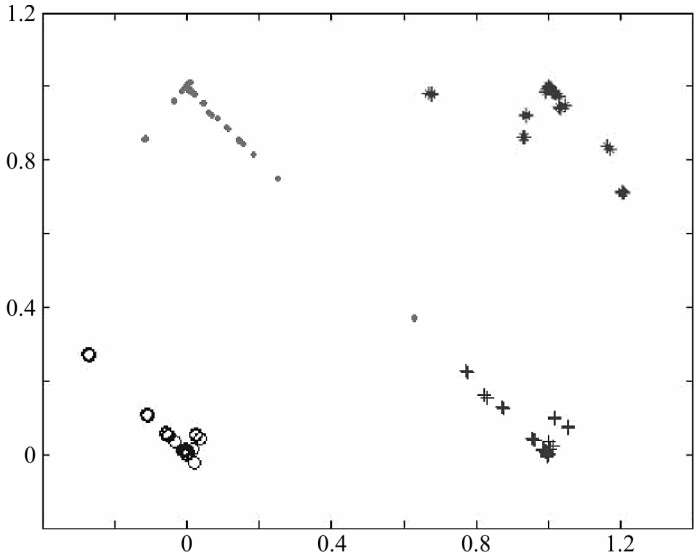

Figure 4. Scatterogram after 40 iterations of training

the next one corresponds to the letter R ('·' symbol), but the classes of the letters B ('+' symbol) and D ('∗' symbol) are partially mixed. Upon completion of the NN training, clusters of points are concentrated near the square vertices. Thus, a high quality of classification of the sample vectors is achieved. Hence, when the number of training iterations is insufficient, there is a set of points near the center of the m-gon in the scatterogram. If the number of training iterations is sufficient for a qualitative classification, there are clearly separated clusters of points around the vertices of the m-gon.

3.2. Robustness of the neural network solutions with respect to the test sample noise. The experiments varied the degree of the test sample noise, which was determined by using the similarity function $H(X, Y)$ of two images, where the image X is seen as a test sample, and the image Y is seen as a version of X-image noised by normal or uniform law. Figure 5 represent examples of versions of the letter A noised by the normal law. When the degree of noise is increased, the number of unrecognized or incorrectly recognized vectors also increases. In Figure 5, the image of the letter A, noised by 30 %, is not recognized by the human eye, so we are interested in less strong noise. Using a scatterogram, we can trace values of the noise degree retaining a high quality of the sample classification and evaluate the robustness of the NN solutions with respect to noise. In experiments with a variable number of hidden neurons, we detect that the network with 10 hidden neurons has a greater robustness than that with two hidden neurons. With 10 hidden neurons and the noise level of 3 and 9 %, the quality of the letter classification remains high, while for a network with two hidden neurons we observe the points treated as unrecognized in the center of the scatterogram.

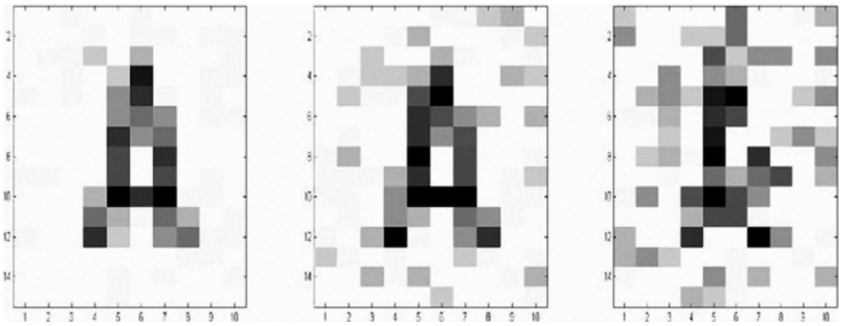

Figure 5. An example of noised training samples. From left to right: the degree of noise pollution 3, 9, and 30 %

3.3. The choice of the number of neurons in the hidden layer. The network was trained using the conjugate gradient algorithm by varying the number of neurons in the hidden layer for 150 iterations, after which the scatterogram was constructed for a slightly noised (3 %) training set. The scatterograms of outputs of trained neural networks, having the same error a the training set, show that their tests on noised samples give different results.

Based on the results of the experiment, the number of neurons in the hidden layer is determined, for which a high quality of classification is obtained with a fixed level of noise, i.e., the scatterogram points are divided into four clusters around the vertices, and there are no points in the center

of the square. This number belongs to the segment $[10, 20]$. As the scatterogram shows, further increase in the number of hidden neurons does not increase the quality of the separation of classes.

4. The study of neuron states in the hidden layer

A neural network having 150 neurons in the input layer (image size), 4 neurons in the hidden layer and 4 neurons in the output layer was considered. The number of neurons in the hidden layer is reduced to simplify a scatterogram. The scatterograms were built for two different learning algorithms: the conjugate gradient method and the method of gradient descent with a different number of iterations. The neuron weights were initialized with random small values displayed in the scatterogram as a cluster of vector images of the hidden weights located in the vicinity of the zero point. For different training algorithms, the neural network shows a different dynamics of the hidden layer in the scatterogram. In training by the conjugate gradient algorithm, several situations have occurred in the scatterogram:

1. The signals of hidden neurons form four clusters. In the training process, these clusters are gradually segregated as classes (Figure 6, top). The share of such point distributions in scatterogram is very small.
2. In the course of training, the lines occurred along the borders of the projections of the cube edges (Figure 6, bottom).
3. The vector images shrink to a point.

In the second and third case, there is a problem of uncertainty, since due to the reduction of data dimensionality, a part of information is lost, and in a four-dimensional space points can be placed on a plane or on a line. An attempt to solve this problem by projecting the vectors onto the plane spanned by the other axis, a similar behavior of points reveals in the scatterogram. In this case, as the basis vectors

$$e_1 = (-1, 1, 1, 1)^T, \quad e_2 = (-0.2113, 0.7889, -0.5774, 0)^T$$

and

$$e_1 = (1, 1, -1, -1)^T, \quad e_2 = (-0.2113, 0.7889, -0.5774, 0)^T$$

were taken.

In the NN training by the method of gradient descent, the first two situations did not take place. Images of the hidden vectors form a cluster in the vicinity of the projection of one of the vertices of the four-dimensional cube and were placed along projections of the edges adjacent to this vertex. This behavior of signals of the hidden layer neurons was detected after the first iteration of the algorithm and did not change during the network training.

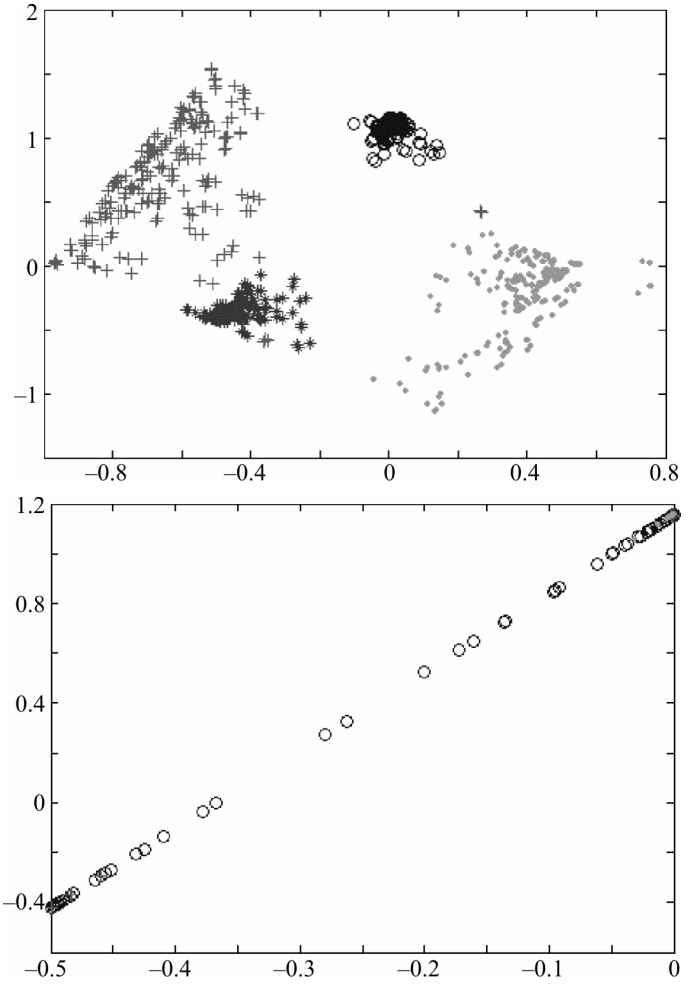

Figure 6. Different kinds of behavior of neurons in the hidden layer in the course of training by the conjugate gradient method

Thus, the visualization of neurons states in the hidden layer has identified the following kinds of behavior of the sigmoid NN:

1. In the course of training by the gradient descent, weights are adjusted so that the output vectors of hidden neurons are shifted to the vertices or faces of a k-dimensional cube, i.e. one or more components of the vector take extreme values.
2. Upon reaching the state, described in Section 1, the weights of the hidden layer neurons remain essentially unchanged, and further training of the network only leads to modifications of the weights of output layer neurons.

3. Having found that the conjugate gradient method for the entire training process affects both the weights of the hidden layer and the output layer weights for a while, as a method of gradient descent determines the weight of the hidden layer on the first iteration, and in the course of further education almost does not change them.

5. Conclusion

Based on scatterograms of the projections on the plane of the states of the output and of the hidden layers of sigmoid neural networks, the NN behavior is studied in solving the problem of recognition of the letters images. The following is established:

1. When the number of training iterations or the number of neurons in the hidden layer neurons is increased, the variance of the states decreases.

2. Noising the test samples increases the variance of the neuron states.

3. For a fixed number of iterations, the dispersion of the neuron states depends on the choice of a training algorithm: the conjugate gradient method gives a lesser variance than the method of steepest descent.

4. The conjugate gradient method consistently improves classification, while for the method of gradient descent it is not true, in general.

5. The conjugate gradient method for the entire training process affects both the weights of the hidden layer and the output layer, while the gradient descent method sets the weights of the hidden layer on the first iteration, and in the course of further education does not change them.

Based on the experiments performed, an estimate is obtained for the number of neurons in the hidden layer, to be sufficient for a quality solution to the problem of recognition. The peculiarities of the behavior of neurons in the hidden layer in the training process are identified.

The scatterogram of projections of state vectors of neurons is a powerful tool for assessing the performance of the neural network. The scatterogram facilitates the process of the NN development and can be used as a tool for studying the behavior of different classes of neural networks. The conclusion, based on the scatterogram, can be immediately obtained after training the network and used for its modification and optimization. The scatterogram tool is clear and easy to understand.

References

[1] Haykin S. Neural Networks. A Comprehensive Foundation. — Prentice Hall Inc., 1999.

[2] Freeman J.A., Skapura D.M. Neural Networks: Algorithms, Applications, and Programming Techniques. — Addison-Wesley Publishing Company, Inc., 1991.

[3] Tarkov M.S. Neurocomputer Systems. — Moscow: Internet University of Inf. Technologies: Binom. Knowledge laboratory, 2006 (In Russian).

[4] Duch W. Coloring black boxes: visualization of neural network decisions // Int. Joint Conference on Neural networks, Portland, Oregon. — 2003. — Vol. I. — P. 1735–1740.

[5] Duch W. Visualization of Hidden Node Activity in Neural Networks: I. Visualization Methods. — Zakopane, Poland, 2004.

[6] Principal Manifolds for Data Visualization and Dimension Reduction / A.N. Gorban, B. Kegl, D.C. Wunsch, A. Zinovyev, eds. — Springer, 2008.

[7] Nabney I., Bishop C. NETLAB software. — Birmingham, UK: Aston University, 1997. — http://www.ncrg.aston.ac.uk/netlab.