

DOI: <https://doi.org/10.15407/kvt192.02.005>

UDC 004.021

**L.S. FAINZILBERG**<sup>1</sup>, D.Sc. (Engineering), Professor,  
Chief Researcher of the Department of Intelligent Automatic Systems  
e-mail: [fainzilberg@gmail.com](mailto:fainzilberg@gmail.com)

**N.A. MATUSHEVYCH**<sup>2</sup>, Master student  
Faculty of Biomedical Engineering  
e-mail: [natalie.matushevych@gmail.com](mailto:natalie.matushevych@gmail.com)

<sup>1</sup> International Research and Training Center for Information Technologies  
and Systems of the National Academy of Sciences of Ukraine  
and Ministry of Education and Science of Ukraine,  
Acad. Glushkova av., 40, Kiev, 03187, Ukraine

<sup>2</sup> The National Technical University of Ukraine  
«Igor Sikorsky Kyiv Polytechnic Institute», Peremohy av., 37, Kiev, 03056, Ukraine

## COMPARATIVE EVALUATION OF CONVERGENCE'S SPEED OF LEARNING ALGORITHMS FOR LINEAR CLASSIFIERS BY STATISTICAL EXPERIMENTS METHOD

---

**Introduction.** One of the main tasks of artificial intelligence is pattern recognition, which is often reduced to determining the discriminant function parameters in the multidimensional feature space. When recognizable objects can be completely separated by a linear discriminant function, the task is reduced to the linear classifier learning. There are many algorithms for linear classifiers learning, two of which are the Rosenblatt learning algorithm and the Kozinets algorithm.

**The purpose of the article** is to investigate the properties of the Rosenblatt and Kozinets learning algorithms on the basis of statistical experiment by the Monte Carlo method.

**Methods.** Two algorithms for linear classifiers learning have been studied: Rosenblatt and Kozinets. A number of researches have been performed to compare the convergence rate of algorithms for a different number of points and for their different location. Variation of the iterations number of algorithms spent on samples of different sizes was analyzed.

**Results.** Statistical experiments have shown that for a small sample size in approximately 20% of cases the convergence rates of the Rosenblatt and Kozinets algorithms are the same, but with the increase of observations number, the Kozinets learning algorithm proved to be the absolute leader. Also, the convergence rate of the Kozinets learning algorithm is less sensitive to the location of points in the learning sample.

© L.S. FAINZILBERG, N.A. MATUSHEVYCH, 2018

ISSN 2519-2205 (Online), ISSN 0454-9910 (Print). Киб. и выч. техн. 2018. № 2 (192)

5

**Conclusions.** *The higher convergence rate of the Kozinets algorithm compared to the Rosenblatt algorithm, confirmed by a series of statistical experiments, allows formulating a promising research line on the evolution of neural networks where the Kozinets algorithm will be used to adjust the basic elements — perceptrons.*

**Keywords:** *Linear classifier, Rosenblatt algorithm, Kozinets algorithm.*

## INTRODUCTION

The task of learning objects recognition of different physical nature (Machine Learning) — one of the main tasks of artificial intelligence [1–6]. Quite often it is regarded as a problem of determining the parameters of the discriminant function (functions) in the multidimensional feature space [7].

Linear discriminant functions deserve special attention, which, according to [8], any Bayesian recognition strategy comes down to in probability space. It should be borne in mind, that linear discriminant function assumes that an increase in the values of one feature can be compensated by a decrease in the value of another feature, which is not always true [9]. Nevertheless, linear classifiers are widely used in solving many practical problems [9, 10].

In those cases when Nature goes to meet the designer of the application system and in the original or transformed (straightening) feature space the recognizable objects can be completely separated by a linear discriminant function, the problem is reduced to the learning of a linear classifier on a finite number of observations [8]. There are variety of linear classifiers learning algorithms, two of which — the perceptron learning algorithm proposed by Frank Rosenblatt [11] and the algorithm of B.N. Kozinets [12].

In a well-known theorem of Novikoff it is proved that the perceptron algorithm converges for a finite number of iterations under the condition of the objects linear separability of the training sample [13]. This theorem is much more clearly and convincingly proved in [14]. It is this proof that is regarded as the canonical proof of perceptron convergence. An analogous theorem on finite convergence is proved for the Kozinets algorithm [12].

In the same time formal conditions that give an estimate of the maximum number of iterations of these algorithms are rather rough [8]. Therefore, these estimates do not allow an unambiguous answer to an important question: which of the algorithms and when provides the fastest rate of convergence in the learning process for the final sample of observations. A range of other properties of these algorithms, which are important in solving specific practical problems are also unknown.

The purpose of the article is to investigate the properties of the Rosenblatt and Kozinets learning algorithms on the basis of statistical experiment by the Monte Carlo method.

## LEARNING ALGORITHMS FOR LINEAR CLASSIFIERS

Before describing the proposed technology for performing a statistical experiment, let's consider basic principles of the Rosenblatt and Kozinets learning algorithms on the example of recognizing two classes  $V_1$  and  $V_2$  [11–13].

Let the observation of the sample with the known belonging to classes is set in  $N$ -dimensional feature space:

$$X = \{(x_1^{(N)}, c_1), (x_2^{(N)}, c_2), \dots, (x_n^{(N)}, c_n)\}, \quad (1)$$

where  $n$  — number of elements in the sample,  $x_j^{(N)} \triangleq (x_{j1}, \dots, x_{jN})$  — points ( $N$ -dimensional vectors), and  $c_j$  — an indicator variable such that:

$$c_j = \begin{cases} +1, & \text{if } x_j^{(N)} \in V_1, \\ -1, & \text{if } x_j^{(N)} \in V_2. \end{cases} \quad j = 1, \dots, n. \quad (2)$$

It is assumed that the observations of the classes  $V_1$  and  $V_2$  can be separated by a linear discriminant function

$$D(x) = \langle w, x \rangle \triangleq \sum_{i=0}^N w_i x_i, \quad (3)$$

in which, for convenience, the notation  $\langle w, x \rangle$  denotes the scalar product ( $N+1$ )-dimensional vectors  $w = (w_0, w_1, \dots, w_N)$  - parameters (weights) of the discriminant function and extended vectors  $x = (1, x_1, \dots, x_N)$ .

The problem is to determine the parameters vector  $w = (w_0, w_1, \dots, w_N)$  of the discriminant function (3) for the final training sample (1) with known values of the indicator variable (2), which will allow us to separate the observations of the sample according to the scheme:

$$\text{decision in favor of } V_1, \text{ if } \langle w, x \rangle > 0, \quad (4)$$

$$\text{decision in favor of } V_2, \text{ if } \langle w, x \rangle < 0. \quad (5)$$

The idea of both algorithms is to implement iterative procedures which allow to adjust some initial value of the vector  $w = (w_0, w_1, \dots, w_N)$ , based on sequential viewing of points in the training sample (1). As a result of such correction after a certain number of iterations, the discriminant function will ensure an error-free separation of the sample elements according to the scheme (4), (5).

The difference between learning algorithms is in the correction mechanism.

The F. Rosenblatt algorithm [11] is reduced to the implementation of such steps (Fig. 1):

1. Arbitrarily set the initial values of the vector  $w^{(0)}$ . For example, for the two-dimensional case ( $N = 2$ ) can set  $w^{(0)} = (0, 0, 1)$ .

2. The observations  $x_\alpha^{(t)} = x^{(N)}$ ,  $\alpha = 1, \dots, n$ , from the training sample (1), is selected sequentially and in accordance with (3) the values of the discriminant function  $D(w^{(t-1)}, x_\alpha^{(t)})$  are defined at the current value of the vector  $w^{(t-1)}$ ,  $t = 1, 2, \dots$ .

3. An error is calculated :

$$\delta_{\alpha}^{(t)} = D(w^{(t-1)}, x_{\alpha}^{(t)}) - c(x_{\alpha}^{(t)}), \tag{6}$$

which is the difference between the value of the discriminant function (3) and the known value of the indicator variable (2), which corresponds to the selected observation  $x_{\alpha}^{(t)}$ .

4. If the current observation  $x_{\alpha}^{(t)}$  is not properly classified (fig. 1, b) the weights vector is modified as follows:

$$w^{(t)} = w^{(t-1)} + \gamma (\delta_{\alpha}^{(t)})^T x_{\alpha}^{(t)}, \tag{7}$$

where  $0 < \gamma < 1$  — preset correction rate (fig. 1, b).

5. Steps 2–4 are repeated until all sampling points (1) have been classified correctly (fig. 1, c).

A theorem was proved in [13], according to which for a finite linearly separable sample the iterations number of the Rosenblatt algorithm is limited:

$$t^0 \leq \frac{Q^2}{\varepsilon^2}, \tag{8}$$

where  $Q = \max_{i \in [1, n]} |x_i^{(N)}|$ ,  $\varepsilon = \min_{x \in \text{Co}(X)} |x^{(N)}| > 0$ , here  $\text{Co}(X)$  — convex hull of the set  $X$ .

In the work [12] B.N. Kozinets proposed different iterative learning algorithm, which was later called the Kozinets algorithm.

The main idea of the algorithm is that at each step of the iteration  $t = 1, 2, \dots$  is searched for such an observation  $x_{\alpha}^{(t)} = x^{(N)}$ ,  $\alpha = 1, \dots, n$ , from the training sample (1), which is incorrectly classified at the current value of the parameter vector  $w^{(t-1)}$  of the discriminant separating function. If there are no such vectors for all points of the training sample, then the algorithm completes its work.

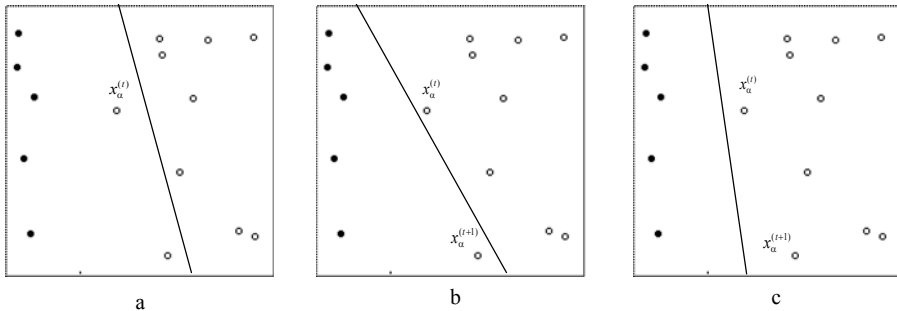


Fig. 1. Graphic interpretation of the Rosenblatt learning algorithm

If a vector  $x_\alpha^{(t)}$  is found that is incorrectly classified, then the parameter vector is corrected as follows [8]:

$$w^{(t)} = (1 - \gamma^{(t)}) \cdot w^{(t-1)} + \gamma^{(t)} \cdot x_\alpha^{(t)}, \quad (9)$$

where

$$\gamma = \operatorname{argmin} \left| (1 - \gamma^{(t)}) \cdot w^{(t-1)} + \gamma^{(t)} \cdot x_\alpha^{(t)} \right|. \quad (10)$$

Let's give a graphic interpretation of the Kozinets algorithm for the case  $N = 2$ , when using a straight line it is necessary to divide two sets of points on the plane. The algorithm is reduced to performing such steps (Fig. 2, 3):

1. Two points of the training sample, belonging to different classes, are randomly selected, and a straight line is drawn between them  $AB$  (Fig. 2, a).

2. The parameters of the middle perpendicular  $W_0$  to the segment  $AB$  determine the initial approximation of the parameters vector  $w^{(0)}$  of the unknown discriminant function.

3. An arbitrary point  $M$  is chosen and the sign of the indicator variable (2) determines its belonging to one of the classes. The point  $M$  connects with a point  $B$  of the same class by line, and the perpendicular  $AP$  is gone down to the indicated line from the point  $A$  of the opposite class (Fig. 2, b).

4, a. If the base of the perpendicular  $AP$  extends beyond the straight line  $BM$  (Fig. 2, b), then the point  $M$  determines the new position of the straight line  $AB$  (Fig. 1, c), with the help of which the parameters  $w^{(1)}$  of the corrected discriminant function  $W_1$  are found.

4, b. If the base of the perpendicular  $AP$  lies within the segment  $BM$  (Fig. 3, b), then the new position of the straight line  $AB$  (and hence the parameters  $w^{(1)}$  of the corrected discriminant function  $W_1$ ) is determined by point  $P$ .

Steps 1–4 are repeated until all points of the training sample (1) are properly classified.

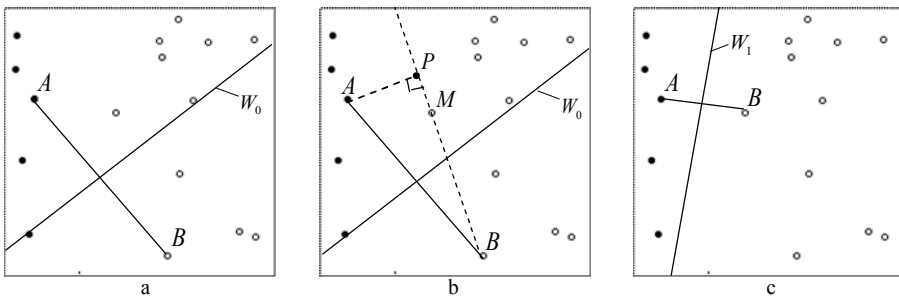


Fig. 2. Graphical interpretation of the Kozinets algorithm on the plane (1st case)

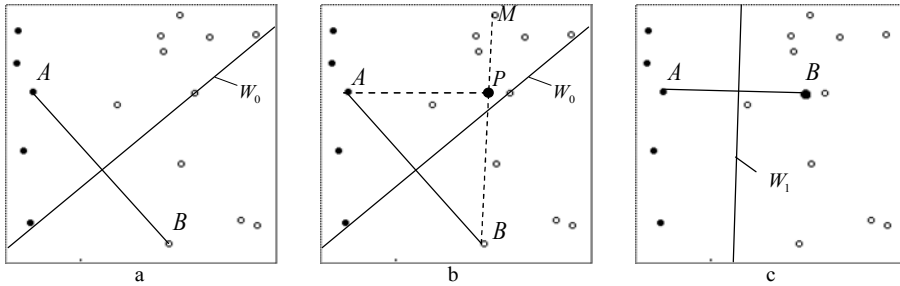


Fig. 3. Graphical interpretation of the Kozinets algorithm on the plane (2nd case)

In accordance with the proved theorem [12, 14], the Kozinets algorithm converges in a finite number of iterations

$$t_0 \leq \frac{Q^2}{\varepsilon^2} \ln \frac{Q^2}{\varepsilon^2}. \quad (11)$$

At first glance, comparing the estimates (8) and (11) it may seem that the Rosenblatt algorithm always converges with a smaller number of iterations. However, as noted in [8], the estimates (8) and (11) are rather rough, and hence the conclusion about the superiority of the Rosenblatt algorithm on the basis of these estimates is not valid. Therefore, it is proposed to compare the rate of algorithms convergence based on the statistical experiment.

## TECHNOLOGY OF STATISTICAL EXPERIMENT IMPLEMENTATION

Following the statistical experiment methodology [15–17], a software tool system was developed. It makes it possible to carry out experiments to estimate the rate of convergence of the Rosenblatt and Kozinets algorithms. Such experiments were performed on the random data samples generated by the Monte Carlo method.

When the program starts (Fig. 4), the user is able to generate data for the experiment (*generate data*) in three different ways:

- automatically (*automatically with preferences*);
- manually by specifying the sampling points corresponding to the different classes on the plane (*manually on the canvas*);
- by loading from a file (*download from file*).

With a single start of the program (*launching unit*), sets of points of two classes are generated, which can obviously be separated by a straight line, and the Rosenblatt and Kozinets learning algorithms are started parallel (Fig. 5).

At the end of the experiment, the dividing lines and the histogram of the iterations numbers spent by each algorithm are displayed (*drawing the unit iteration chart*).

For multiple experiments (*launching multiple*), a number of additional functions are available, in particular, a histogram display of the percentage of the iterations number spent by each algorithm in learning (*drawing chart with percent of the multiple iterations*). This makes it possible to evaluate and compare the probabilities of successful learning completion and thereby determine the leader with specific system settings.

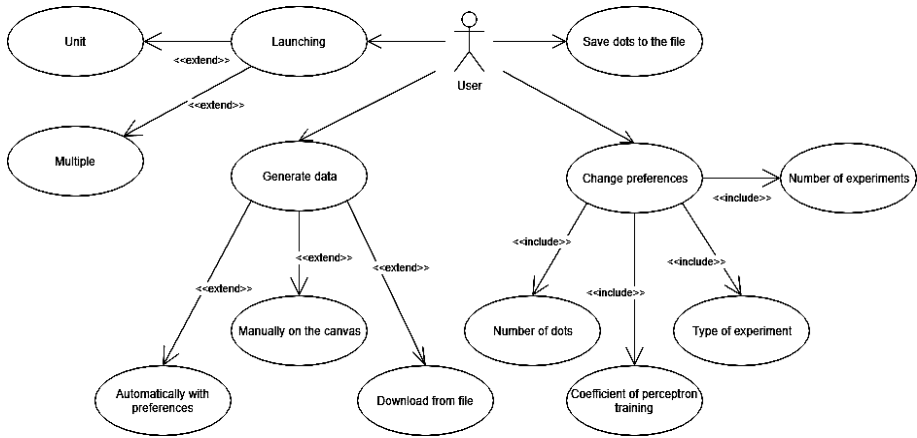


Fig. 4. Use Case diagram in UML notations

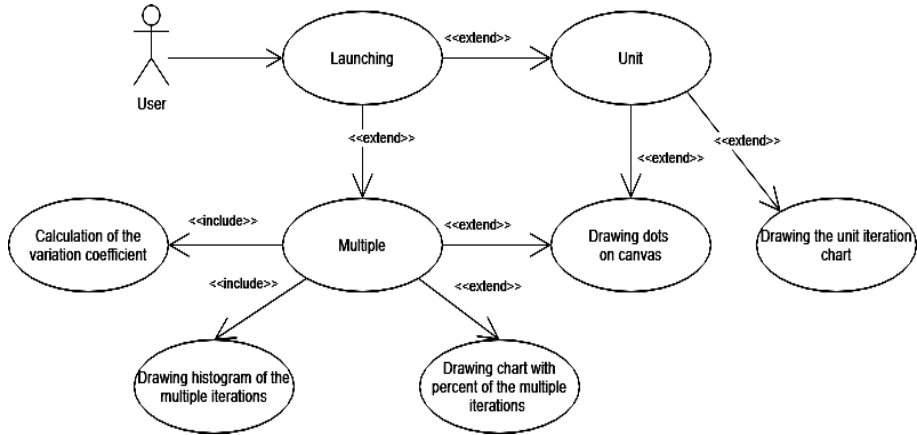


Fig. 5. Use Case diagram in UML notations

It is also possible to calculate the coefficient of variation of the iterations number spent by algorithms in multiple experiments (*calculation of the variation coefficient*).

For illustration on Fig. 6 it is presented a sequence diagram, which explaining the details of the relationships between the main units of the program: the interface (*desktop application*), the point generation unit (*points generation*), the drawing unit (*drawing algorithm*) and the algorithms of Rosenblatt (*Perceptron algorithm*) and Kozinets (*Kozinets algorithm*).

Each time when the procedures that implement the algorithms of Rosenblatt (*Perceptron algorithm*) and Kozinets (*Kozinets algorithm*) run, the iteration numbers  $U_1$  and  $U_2$  are calculated before the work stoppage the Rosenblatt and Kozinets algorithms, respectively. The stopping moments determine the condition

$$\langle w, x_j \rangle \cdot c_j > 0, \quad j = 1, \dots, n, \quad (12)$$

which in accordance with (2), (4), (5) indicates that at the current value of the parameter vector  $w$  all points of the training sample corresponding to the classes  $V_1$  and  $V_2$ , are completely separated by a linear discriminant function.

Based on the comparison of numbers  $U_1$  and  $U_2$ , the leader in the current experiment is determined:

The leader is the Rosenblatt algorithm, if  $U_1 < U_2$ ,

The leader is the Kozinets algorithm, if  $U_2 < U_1$ .

Results of comparison (*result of comparison*) are visualized by the corresponding bar chart (*charts drawing*).

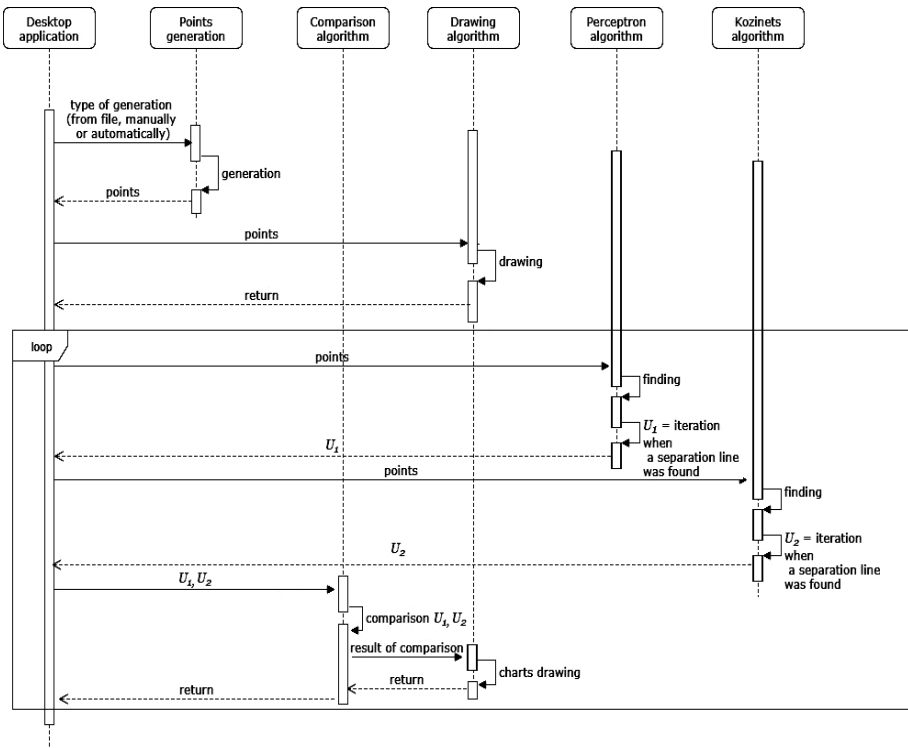


Fig. 6. Sequence diagram



## RESULTS OF STATISTICAL EXPERIMENTS

Let's consider some results obtained during the experimental study of the properties of the Rosenblatt and Kozinets learning algorithms on samples of random observations linearly separable on the plane.

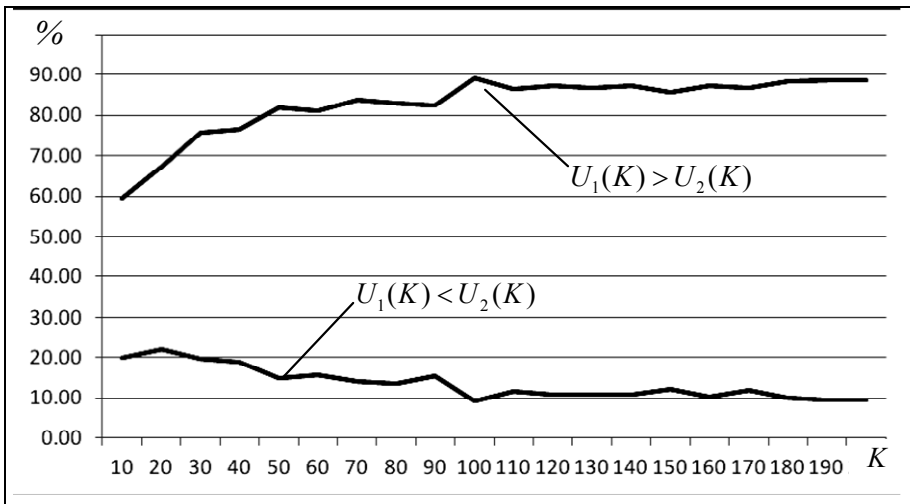
The first series of experiments was aimed at estimating the rate of algorithms convergence for a different number of points  $K = 10, 20, \dots, 200$ , randomly located on the plane. In each individual experiment, the iteration numbers  $U_1(K)$  and  $U_2(K)$  were determined until the Rosenblatt and Kozinets algorithms were stopped, respectively, in accordance with condition (12).

Comparison of numbers  $U_1(K)$  and  $U_2(K)$  made it possible to determine the leader of a separate experiment, which divided the points for a smaller number of corrections of the discriminant function parameters. Further, the leadership percentage in the series of 5000 experiments was determined (Fig. 7).

The experiments showed that in approximately 20% of cases, with  $K < 40$  of observations, both algorithms required the same number of iterations. With the increase in the number of points, Kozinets learning algorithm was an absolute leader (Fig. 7).

At the same time, even with more points in 10% of cases (in one of ten), Rosenblatt algorithm was learned faster than the Kozinets algorithm.

The experiments also showed that the convergence rate of the Rosenblatt algorithm depends not only on the number of points being processed, but also on their location to each other on the plane.



**Fig. 7.** The graph of the percent dependence of the algorithms leadership on the number of points:  $U_1(K) < U_2(K)$  — the leader is the Rosenblatt algorithm;  $U_1(K) > U_2(K)$  — the leader is the Kozinets algorithm

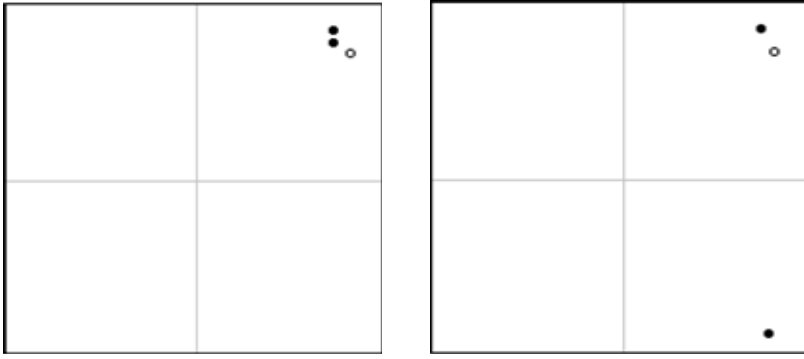


Fig. 8. «Simple» (left) and «difficult» (right) training samples

To illustrate this fact, let's consider the results of multiple learning experiments on two samples, consisting of only three points — two points of the class  $V_1$  and one point of the class  $V_2$  (Fig. 8).

The sample shown in the Fig. 8, left, turned out to be «simple» for both algorithms: for the classes separation the Rosenblatt algorithm required 12 iterations, and the Kozinets algorithm — only one iteration.

At the same time, when the algorithms were learned on a sample shown in the Fig. 8, right, the convergence speeds of the algorithms differed significantly. Kozinets algorithm coped with the problem in just two iterations, at the same time for the Rosenblatt algorithm this sample turned out to be «difficult»: more than 500 iterations were required to separate the points.

For this effect explanation, let's consider the dynamics of the change in the discriminant function position in the Rosenblatt algorithm learning process for the two samples.

In accordance with expression (7) for case with  $N = 2$  the correction of the linear discriminant function's position

$$w_2x_2 + w_1x_1 + w_0, \tag{13}$$

which occurs when the point is incorrectly classified, is reduced to three operations:

$$w_2^{(t)} = w_2^{(t-1)} + \gamma\delta_\alpha^{(t)}x_2^{(t)}, \tag{14}$$

$$w_1^{(t)} = w_1^{(t-1)} + \gamma\delta_\alpha^{(t)}x_1^{(t)}, \tag{15}$$

$$w_0^{(t)} = w_0^{(t-1)} + \gamma\delta_\alpha^{(t)}, \tag{16}$$

where  $\delta_\alpha^{(t)}$  — classification error.

In the process of learning on a «simple» sample, the dividing line (13) from the initial position (Fig. 9, 1), corresponding to vector  $w^{(0)} = (0, 0, 1)$ , gradually changes its direction (Fig. 9, 2–5) and with a relatively small number of itera-

tions takes the final position (Fig. 9, 6), in which the points of the training sample are correctly classified. At the same time, changes in the position of the straight line are mainly due to a change in its slope, determined by the ratio  $w_1^{(t)} / w_2^{(t)}$ , since the separation of «simple» sampling points practically does not require the displacement of a straight line relative to the origin of coordinates, i.e. correction of attitude  $w_0^{(t)} / w_2^{(t)}$ .

During the learning of Rosenblatt algorithm for the "difficult" sample, the dynamics of the change in the discriminant function position is completely different (Fig. 10). A distinctive feature of the «difficult» sample from «simple» consists in a significant difference in distances along the vertical  $\Delta x_2 = |x_2^A - x_2^C|$  and horizontal  $\Delta x_1 = |x_1^A - x_1^C|$  between the points  $A$  and  $C$  of one class:

$$\Delta x_2 \gg \Delta x_1 . \tag{18}$$

In other words, in this case the points  $A$  and  $C$  of one class differ *substantially* in one of the coordinates and practically coincide in the other. This leads to the fact that during the adjustment it is required to change not only the slope of the discriminant function  $w_1^{(t)} / w_2^{(t)}$  (Fig. 10, 1–3), but also its displacement  $w_0^{(t)} / w_2^{(t)}$  (Fig. 10, 4–9).

Therefore, on a «difficult» sample, the Rosenblatt algorithm, though converging in a finite number of steps (long live the Novikoff theorem!), but the number of such steps is large enough.

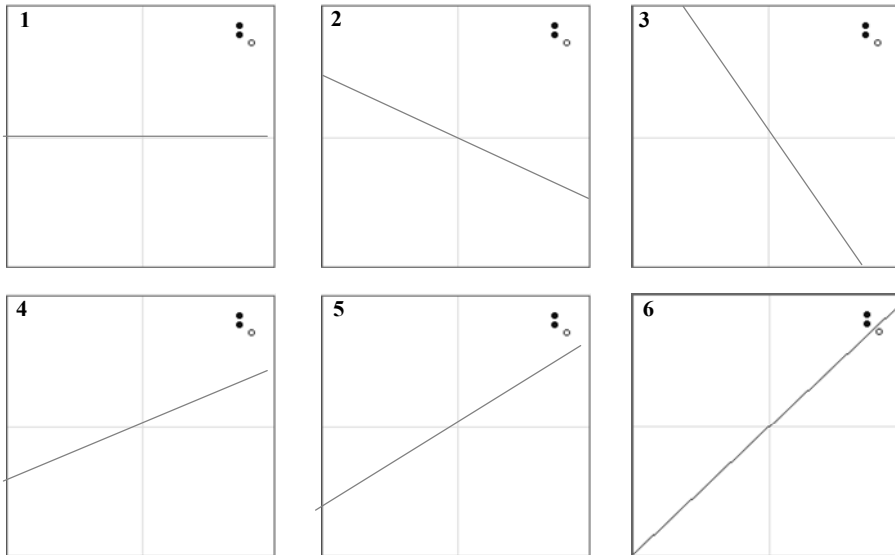


Fig. 9. Stages of the Rosenblatt algorithm learning process on a «simple» sample

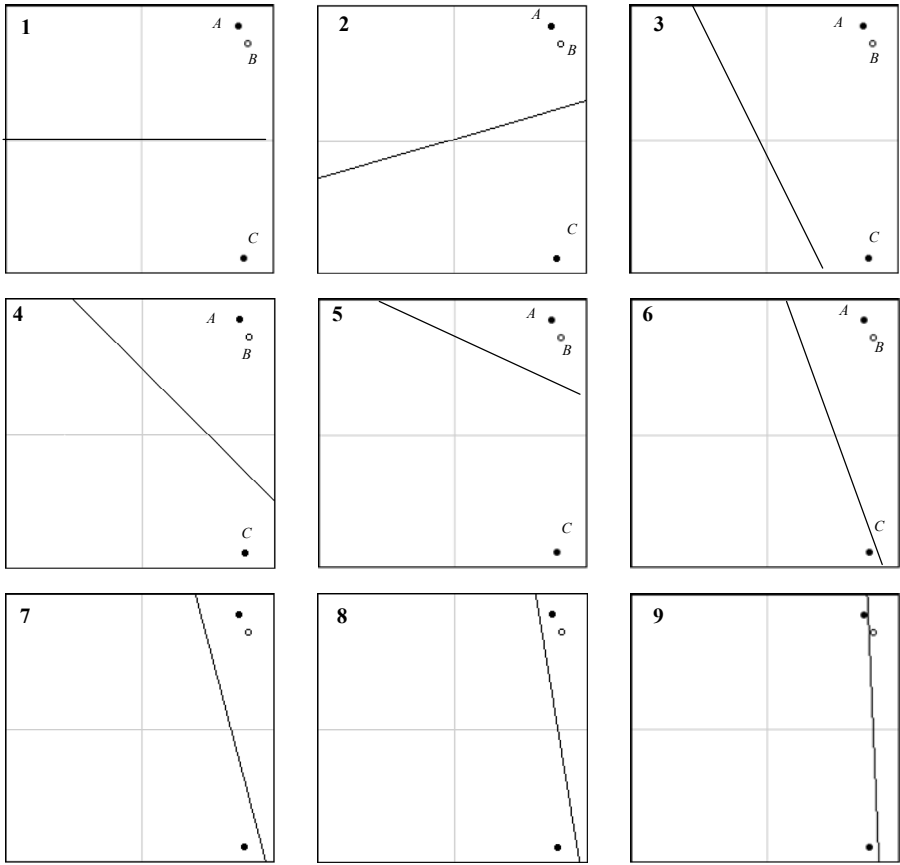


Fig. 10. Stages of the Rosenblatt algorithm learning process on a «difficult» sample

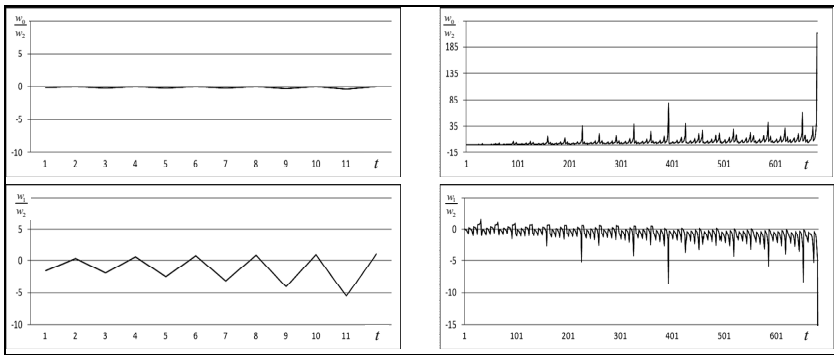


Fig. 11. Dynamics of changes in the slope and displacement of a linear discriminant function on the «simple» (left) and the «difficult» (right) training samples

For illustration, the Fig. 11 shows the dynamics of changing attitudes  $w_1^{(t)} / w_2^{(t)}$  and  $w_0^{(t)} / w_2^{(t)}$  in the learning process on «simple» (left) and «difficult» (right) samples. It is easy to see that on the «simple» sample the slope of

the discriminant function did not practically change, while on the «difficult» one it varied from 0 to 200 cu.

It's worth paying attention to one important difference between the learning procedures for the Rosenblatt and Kozinets algorithms. As already noted, the Kozinets algorithm provides a search for the first incorrectly classified sampling point, which leads to the next correction of the discriminant function's parameter vector.

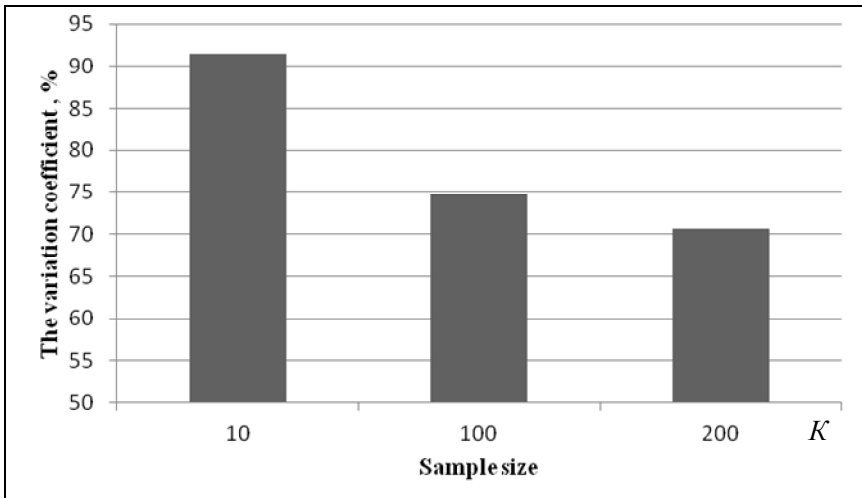
Since the search for such points is carried out randomly, for the acceleration of the learning process, it is advisable at each next stage of viewing the training sample's points to exclude the possibility of *repeatedly* checking the fulfillment of the condition  $\langle w, x_k \rangle \cdot c_k > 0$  of correct classification for the same observations  $x_k$ ,  $k \in [1, n]$ .

For this purpose, an uncomplicated optimization procedure was developed. It ensures the formation of an observations' reduced subset for the next step in the correction of the parameter vector. Of course, after this step is completed, a new incorrectly classified point is searched for the entire sample of observations.

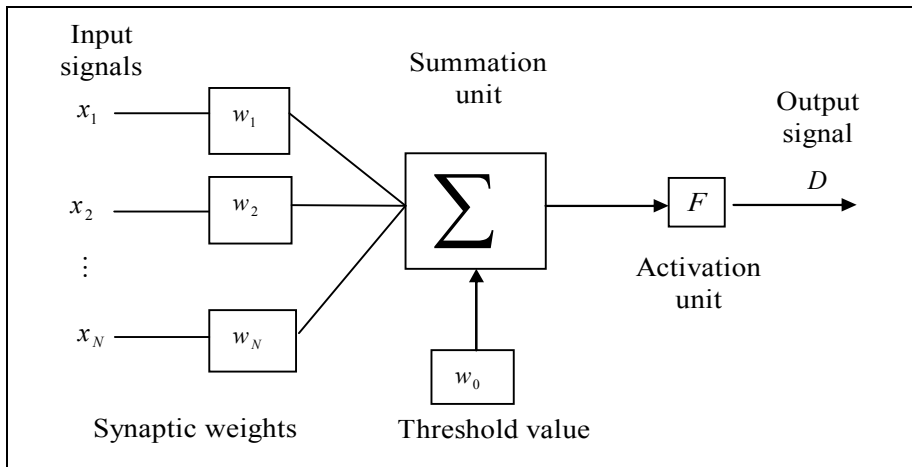
Statistical experiments showed that on average, usage of the optimization procedure allows accelerating the convergence time of the Kozinets algorithm more than eight times.

During the experiments execution, the analysis of number variation of spent iterations was carried out for the Kozinets algorithm on samples of different volumes. For this, during the multiple learning of the algorithm on a specific sample, the Pearson's variation coefficient was calculated [18] — the ratio of the standard deviation of the iterations number to the average value of the iterations number, expressed as a percentage.

The experiments showed that with increasing number  $K$  of points in the sample, the variation coefficient of the iterations number decreases: after increasing the sample size from 10 to 200 points, the variation coefficient of the iterations number decreased by 20% (Fig. 12).



**Fig. 12.** Dependence of the iterations number variation coefficient of the Kozinets algorithm from the sample size



**Fig. 13.** Rosenblatt single-layer perceptron

Recall that the Rosenblatt learning algorithm suggests the principle of adjusting the linear discriminant function parameters (3), which structurally implements the perceptron scheme (Fig. 13) — the basic element of neural networks [19], which are actively used to solve many applied problems [20].

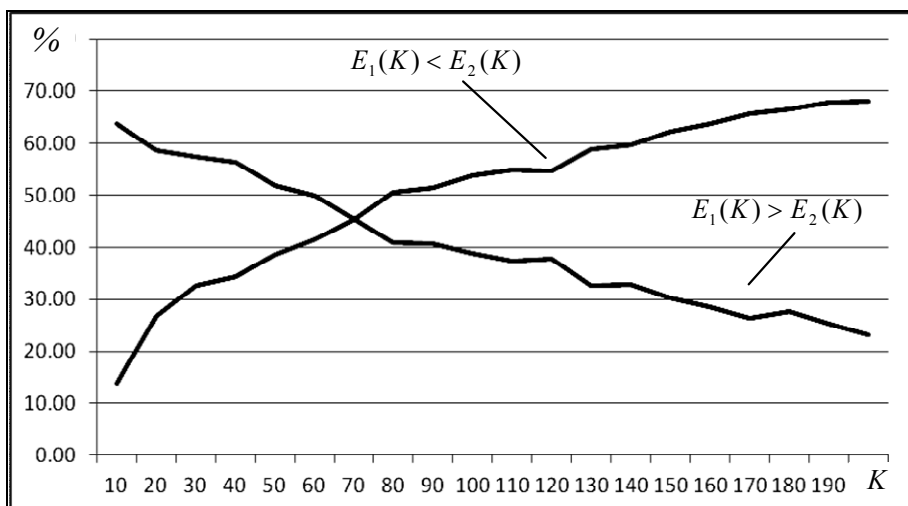
It is clear that the Kozinets algorithm, in fact, offers an alternative approach to learning the same scheme. The experimental researches that were carried out, which confirmed the high rate of convergence of the Kozinets algorithm, allow to hope that using of this algorithm as the learning one for the basic elements of a neural network will increase their effectiveness. At least, the research of this possibility is promising and will be the subject of our further investigations.

In conclusion, recall that for the learning process characteristics of neural networks used a special term — the «learning epoch» [10], which means the stage of the discriminant function parameters correction for a single viewing of *all* points in the training sample.

It is clear that for the Kozinets learning algorithm the «learning epoch» and the «iteration step» are the equivalent concepts. At the same time, the «learning epoch» for the Rosenblatt algorithm consists of the corrections sequence of the separating function parameters for a single scan of the entire sample and the detection of each incorrectly classified observation.

Taking into account the above interpretations of terms, the comparative analysis of the «learning epochs» number  $E_1$ ,  $E_2$  that were spent by one and the other algorithms on a series of randomly generated observations in the experiments was carried out.

Experiments have shown that from this point of view, with the points amount  $K < 70$  the Kozinets algorithm has an advantage, and as the points amount  $K$  increases, the leadership percentage of the Rosenblatt algorithm increases respectively (Fig. 14).



**Fig. 14.** The graph of the percent dependence of the «learning epochs» number on the observations number:  $E_1(K) < E_2(K)$  — the leader is the Rosenblatt algorithm;  $E_1(K) > E_2(K)$  — the leader is the Kozinets algorithm

After all, the rate of the learning algorithms convergence mainly characterizes the corrections number of the discriminant function parameters, and hence, from this point of view, the Kozinets algorithm can be considered an uncontested leader, which is illustrated by the Fig. 7.

Thus, the developed software system made it possible to establish previously unknown properties of the Rosenblatt and Kozinets learning algorithms on the basis of a statistical experiments series, to conduct their comparative analysis and outline the prospects for further research on the improvement of neural networks.

## CONCLUSIONS

Statistical experiments carried out using the developed software system have shown that for a small sample size the convergence rates of the Rosenblatt and Kozinets algorithms are the same in approximately 20% of cases. With the increase in the number of observations, the Kozinets learning algorithm proved to be the absolute leader and, with a  $K > 100$  number of observations, it learned faster than the Rosenblatt algorithm in 90% of the cases.

The convergence rate of the Kozinets learning algorithm is less sensitive to the points location in the training sample and with the increase in the observations number of the variation coefficient, the iterations number decreases.

The higher convergence rate of the Kozinets algorithm compared to the Rosenblatt algorithm, confirmed by a series of statistical experiments, allows formulating a promising line of research on the evolution of neural networks in which the Kozinets algorithm will be used to adjust the basic elements — perceptrons.

REFERENCES

1. Hastie T., Tibshirani R., Friedman J. The elements of statistical learning. New York: Springer; 2014. 739 p.
2. Bishop C.M. Pattern recognition and machine learning. New York: Springer; 2006. 738 p.
3. Merkov A.B. Image recognition: Introduction to statistical learning methods. Moscow: URSS; 2011. 256 p. (In Russian).
4. Vapnik, V. The nature of statistical learning theory. New York: Springer-Verlag; 1995. 188 p.
5. Gori M. Machine Learning: A constraint-based approach. — Waltham: Morgan Kaufmann; 2017. 580 p.
6. Kodratoff Y., Michalski R.S. Machine learning: an artificial intelligence approach, Vol. 3. Elsevier; 2014. 825 p.
7. Camastra F., Vinciarelli A. Machine learning for audio, image and video analysis: Theory and Applications. Madrid: Springer; 2015. 561 p.
8. Schlesinger M, Hlavac V. Ten lectures on statistical and structural pattern recognition. Dordrecht/Boston /London: Kluwer Academic Publishers; 2002. 519 p.
9. Burstein F., Holsapple C.I.W. Handbook on decision support systems 2: variations. New York: Springer; 2008. 798 p.
10. Kung S.Y. Kernel methods and machine learning. Cambridge: Cambridge University Press; 2014. 591 p.
11. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*. 1958. N. 65(6), P. 386–408.
12. Kozinets B.N. A recursive algorithm for dividing the convex hulls of two sets. *Computer Science and Programming*. 1966. N. 4. P. 43–50. (In Russian).
13. Novikoff A.B. On convergence proofs on perceptrons. *Symposium on the Mathematical Theory of Automata*. 1962. N. 12, P. 615–622.
14. Vapnik V.N., Chervonenkis A.J. Theory of pattern recognition. Moscow: Nauka; 1974. 416 p. (In Russian).
15. Buslenko N.P. The method of statistical modeling. Moscow: Statistics; 1970. 113 p. (In Russian).
16. Rubinstein R.Y., Kroese D.P. Simulation and the Monte Carlo method (3 ed.). New York: John Wiley & Sons; 2016. 432 p.
17. Robert C.P., Casella G. Monte Carlo statistical methods (2nd ed.). New York: Springer; 2004. 649 p.
18. Trickey K.A. Structural models of coefficient of variation matrices. Los Angeles: University of California; 2015. 233 p.
19. Kussul E., Baidyk T., Wunsch D.C. Neural networks and micromechanics. New York: Springer Science & Business Media; 2009. 221 p.
20. Misuno I.S., Rachkovskij D.A., Slipchenko S.V. Experimental investigation of handwritten digit classification. *System Technologies*. 2005. Issue 4(39). P. 110–133.

Received 15.03.2018

ЛИТЕРАТУРА

1. Hastie T., Tibshirani R., Friedman J. The elements of statistical learning. New York: Springer; 2014. 739 p.
2. Bishop C.M. Pattern recognition and machine learning. New York: Springer; 2006. 738 p.
3. Мерков А.Б. Распознавание образов. Введение в методы статистического обучения. Москва; 2011. 256 с.
4. Vapnik, V. The nature of statistical learning theory. New York: Springer-Verlag; 1995. 188 p.
5. Gori M. Machine Learning: A constraint-based approach. — Waltham: Morgan Kaufmann; 2017. 580 p.
6. Kodratoff Y., Michalski R.S. Machine learning: an artificial intelligence approach, Vol. 3. Elsevier; 2014. 825 p.
7. Camastra F., Vinciarelli A. Machine learning for audio, image and video analysis: Theory and Applications. Madrid: Springer; 2015. 561 p.
8. Schlesinger M, Hlavac V. Ten lectures on statistical and structural pattern recognition. Dordrecht/Boston /London: Kluwer Academic Publishers; 2002. 519 p.



9. Burstein F., Holsapple C.I.W. Handbook on decision support systems 2: variations. New York: Springer; 2008. 798 p.
10. Kung S.Y. Kernel methods and machine learning. Cambridge: Cambridge University Press; 2014. 591 p.
11. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain, *Psychological Review*. 1958. N. 65(6), P. 386–408.
12. Козинец Б.Н. Рекуррентний алгоритм заделення випуклих оболочек двух множеств. *Вычислительная техника и вопросы программирования*. 1966. №. 4. С. 43–50.
13. Novikoff A.B. On convergence proofs on perceptrons. *Symposium on the Mathematical Theory of Automata*. 1962. N. 12, P. 615–622.
14. Вапник В.Н., Червоненкис А.Я. Теория распознавания образов. Москва: Наука; 1974. 416 с.
15. Бусленко Н.П. Метод статистического моделирования. Москва: Статистика; 1970. 113 с.
16. Rubinstein R.Y., Kroese D.P. Simulation and the Monte Carlo method (3 ed.). New York: John Wiley & Sons; 2016. 432 p.
17. Robert C.P., Casella G. Monte Carlo statistical methods (2nd ed.). New York: Springer; 2004. 649 p.
18. Trickey K.A. Structural models of coefficient of variation matrices. Los Angeles: University of California; 2015. 233 p.
19. Kussul E., Baidyk T., Wunsch D.C. Neural networks and micromechanics. New York: Springer Science & Business Media; 2009. 221 p.
20. Misuno I.S., Rachkovskij D.A., Slipchenko S.V. Experimental investigation of handwritten digit classification. *System Technologies*. 2005. Issue 4(39). P. 110–133.

Получено 15.03.2018

*Л.С. Файнзильберг*<sup>1</sup>, д-р техн. наук, проф.,  
голов. наук. співроб. відд. інтелектуальних автоматичних систем  
e-mail: fainzilberg@gmail.com

*Н.А. Матушевич*<sup>2</sup>, магістрант  
факультет біомедичної інженерії  
e-mail: natalie.matushevych@gmail.com

<sup>1</sup> Міжнародний науково-навчальний центр інформаційних  
технологій та систем НАН України та МОН України,  
пр. Академіка Глушкова, 40, м. Київ, 03187, Україна

<sup>2</sup> Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»,  
пр. Перемоги, 37, Київ, 03056, Україна

#### ПОРІВНЯЛЬНА ОЦІНКА ШВИДКОСТІ ЗБІЖНОСТІ АЛГОРИТМІВ НАВЧАННЯ ЛІНІЙНИХ КЛАСИФІКАТОРІВ ЗА МЕТОДОМ СТАТИСТИЧНОГО ЕКСПЕРИМЕНТУ

Розглянуто алгоритми лінійної класифікації Ф. Розенблата та Б.Н. Козинця. Проведено експериментальні дослідження збіжності алгоритмів на різних вибірках даних. Наведено результати статистичних експериментів для оцінювання швидкості збіжності алгоритмів Козинця та Розенблата, залежності результатів від розташування елементів в вибірці та варіації кількості ітерацій алгоритмів під час навчання на вибірках різного обсягу.

Більша швидкість збіжності алгоритму Козинця у порівнянні з алгоритмом Розенблата, що підтверджено серіями проведених статистичних експериментів, дозволяє сформулювати перспективний напрямок досліджень з розвитку нейронних мереж, в яких алгоритм Козинця буде використано для настройки базових елементів — перцептронів.

**Ключові слова:** лінійний класифікатор, алгоритм Розенблата, алгоритм Козинця, перцептрон.

Л.С. Файнзилберг<sup>1</sup>, д-р техн. наук, проф.,  
глав. науч. сотр. отд. интеллектуальных автоматических систем  
e-mail: fainzilberg@gmail.com

Н.А. Матушевич<sup>2</sup>, магистрант  
факультет биомедицинской инженерии  
e-mail: natalie.matushevych@gmail.com

<sup>1</sup> Международный научно-учебный центр информационных  
технологий и систем НАН Украины и МОН Украины,  
пр. Академика Глушкова, 40, г. Киев, 03187, Украина

<sup>2</sup> Национальный технический университет Украины  
«Киевский политехнический институт имени Игоря Сикорского»,  
пр. Победы, 37, Киев, 03056, Украина

## СРАВНИТЕЛЬНАЯ ОЦЕНКА СКОРОСТИ СХОДИМОСТИ АЛГОРИТМОВ ОБУЧЕНИЯ ЛИНЕЙНЫХ КЛАССИФИКАТОРОВ МЕТОДОМ СТАТИСТИЧЕСКОГО ЭКСПЕРИМЕНТА

**Введение.** Одной из главных задач искусственного интеллекта является распознавание образов, которое довольно часто сводится к определению параметров дискриминантной функции в многомерном пространстве признаков. Когда распознаваемые объекты могут быть полностью разделены линейной дискриминантной функцией, задача сводится к обучению линейного классификатора. Существует множество алгоритмов обучения линейных классификаторов, два из которых — алгоритм обучения Розенблатта и алгоритм Козинца.

**Цель статьи** — исследовать свойства алгоритмов обучения Розенблатта и Козинца на основе проведения статистического эксперимента методом Монте-Карло.

**Методы.** Исследованы два алгоритма обучения линейных классификаторов: Розенблатта и Козинца. Проведен ряд исследований для сравнения скорости сходимости алгоритмов при различном числе точек и их расположении. Проанализирована вариация количества затраченных итераций алгоритмами на выборках разного объема.

**Результаты.** Экспериментальные исследования позволили определить, что при малом объеме выборки приблизительно в 20 % случаев скорости сходимости алгоритмов Розенблатта и Козинца одинаковы, но с увеличением количества наблюдений алгоритм обучения Козинца оказывался абсолютным лидером. Также скорость сходимости алгоритма обучения Козинца менее чувствительна к расположению точек в обучающей выборке.

**Выводы.** Более высокая скорость сходимости алгоритма Козинца по сравнению с алгоритмом Розенблатта, подтвержденная сериями проведенных статистических экспериментов, позволяет сформулировать перспективное направление исследований по развитию нейронных сетей, в которых алгоритм Козинца будет использован для настройки базовых элементов — персептронов.

**Ключевые слова:** линейный классификатор, алгоритм Розенблатта, алгоритм Козинца, персептрон.