Application of Artificial Neural Networks in the Problems of the Patient's Condition Diagnosis in Medical Monitoring Systems

Viktoriia Strilets[1](✉), Nina Bakumenko[1], Serhii Chernysh[2], Mykhaylo Ugryumov[1], and Volodymyr Donets[1]

[1] V. N. Karazin Kharkiv National University, Kharkiv, Ukraine
striletsvictoria@gmail.com, n.bakumenko@karazin.ua, ugryumov.mykhaylo52@gmail.com, vovan.s.marsa@gmail.com

[2] National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine
91sergey@gmail.com

Abstract. The problem of accurate medical diagnosis is always urgent for any person. Existing methods for solving the problem of classification of the state of a complex system are considered. The paper proposes a method of classification of patients' status in medical monitoring systems using artificial neural networks. The artificial neural networks training method uses bee colonies to simulate less training error. The research purpose is to determine the patient's belonging to a particular class according to the variables of his condition, which are recorded. Examples of using the method to determine the status of patients with urological diseases and liver disease are given. The classification accuracy was more than 80%.

Keywords: Classification problem · Artificial neural networks · ROC-curve · Medicine monitoring

1 Introduction

The defects appearance and defects development in complex systems is a complex dynamic process. Experts cannot always predict their appearance and behavior of the system in this case. It is not always possible to come to one thought at what stage of development is the defect and in what state of the system, what methods must be applied to eliminate the defect and normalize the state of the system. Controlling and predicting the dynamic process of a complex system helps experts make decisions that lead to better performance.

As a complex system was considered medical and biological system, which includes a doctor, patients and a system for diagnosing the condition of patients. Each treatment cycle is characterized by a set of final patient's states. The number of states under consideration is determined by the expert based on the results of the cluster analysis. The method for solving the problem of patient clustering is presented in [1].

We accept the hypothesis that the patient's condition is uniquely determined by the system of variables. The problem of recognizing the patient's current state is reduced to the problem of classifying the values of the patient's state variables.

Much attention is paid to the classification problems of complex systems. Many papers have been published today that describe methods for solving the classification problem for technical and biomedical systems.

A naive Bayes classifier is a simple probabilistic classifier based on the Bayes theorem with a "naive" assumption of independence. Bayesian classifiers can be very effective [2] or minor modifications of algorithms can greatly increase its efficiency [3].

Despite the apparent simplicity, the naive Bayesian classifier is capable of producing excellent results even in complex life problems [2].

The advantage of this classifier is a small amount of data necessary for training, parameter evaluation and classification [4]. Also, the naive Bayesian method is often in demand in the classification of text because of the implementation simplicity and the very high efficiency of the classification of text or entities, the nature of which is tied to a static difference [3, 4].

K-Nearest Neighbor Classifier is a metric algorithm for automatic classification of objects or regression, it is also one of the most commonly used and well-known methods for performing recognition tasks and solving data mining problems [4]. Among the advantages of this method, one can single out the simplicity of implementation and high performance and not a high probability of error [5]. Among the shortcomings are high requirements for the computer memory, intolerance to noise and, as a consequence, poor efficiency, the solution to these problems is considered in [6].

Random Forest Classifier is a machine learning algorithm proposed by Braiman [12], the essence of which is to use a certain set of decision trees. This algorithm combines the Breiman bagging method and the random space method proposed by Ho [7]. Due to the fact that the decision tree is the basis of the algorithm, the algorithm provides a comparison of the importance of parameters and helps to identify extremely important properties for determining the final class or state [8–10]. The works in which this algorithm was used to determine the condition of patients, speech recognition and handwriting are given [8]. Modifications of this algorithm help determine the sensitivity to drugs and select candidates for medical research [9]. Random Forest Classifier also helps determine the type of cancer, highlighting important parameters that doctors can pay attention to on their own classification, which can increase the likelihood of diagnosing cancer even in the absence of access to this system [10]. Another area of use of modified Random Forest algorithms is the prediction of molecular interactions [11].

Logistic Regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although there are many more complex extensions. The binary model of logistic regression has extensions to more than two levels of the dependent variable: categorical output data with more than two values are modeled using polynomial logistic regression, and if several categories are ordered using ordinal logistic regression, for example, an ordinal proportional coefficient logistic model. The model itself simply simulates the probability of an output in terms of input data and does not perform statistical classification (this is not a classifier), although it can be used to create a classifier, for example, by choosing a limit value and classifying the input data with a higher probability than a cutoff as one class, lower cutoffs like the other; This is a common way to make a binary classifier. Coefficients are usually not computed by a closed-form expression, in contrast to linear least squares. Also, due to the nature of this model, it is possible to obtain a graph with the importance of properties [12, 13].

Decision Tree Classifier is a decision support tool used in machine learning, data analysis and statistics. This tool uses a tree-like model of decisions and their possible consequences, including random outcomes of events, resource costs, and utility. This is one way to display an algorithm consisting only of conditional statements. Decision trees are commonly used in operations research, especially in decision analysis, to determine a strategy with the highest probability of achieving a goal [14, 15].

AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm formulated by Yoav Freind and Robert Shapira [20]. This algorithm can be used in combination with several classification algorithms to improve their efficiency. The algorithm strengthens the classifiers by uniting them into a "committee". AdaBoost is adaptive in the sense that each subsequent committee of classifiers is built on objects incorrectly classified by previous committees. AdaBoost is sensitive to data noise and outliers. However, it is less prone to retraining compared to other machine learning algorithms [17, 18].

AdaBoost calls weak classifiers in the cycle t = 1, …, T. After each call, the distribution of weights Dt is updated, which correspond to the importance of each of the objects in the training set for classification. At each iteration, the weights of each incorrectly classified object increase, thus the new committee of classifiers "focuses its attention" on these objects. Among the advantages, it is possible to single out the simplicity of implementation and high work efficiency, as the algorithm tends to increase the margin [17–19].

Publications [20, 21] detail the general principles of the theory of artificial neural networks that are being widely used to construct diagnostic models in the form of regression equations.

The research purpose is to determine the patient's belonging to a particular class according to the variables of his condition, which are recorded, using artificial neural networks.

2 Research Problem Statement

On the basis of the systematic analysis of the process of diagnosing the patient's condition, a hierarchy of stages of diagnosis was determined: laboratory diagnosis (blood tests, etc.), visual diagnosis (ultrasound, MRI, etc.) and doctor's examination. At each stage, the corresponding patient status variables are logged. An experimental sample of the variables being recorded was formed, characterizing the condition of the patients being monitored.

Suppose there is a multidimensional state matrix $X = \{x_{i,j}\}$ (i = 1 … I, j = 1 … J), where I is the number of patients in the sample, J is the number of measured state variables. Traditionally, the rows of this matrix are called precedents. Centering and normalization of data is performed by the formula

$$\overset{\circ}{x}_{ij} = \left(x_{ij} - \langle X_j \rangle\right)/\sigma_j, \qquad (1)$$

where $\langle X_j \rangle$ is the mean value of the jth state variable, σ_j is its mean square deviation.

The problem of a diagnostic model building: a vector function is given by the set of training pairs $\left(\vec{X}^{(0)}, \vec{d}\right)_p$, $p = 1 \ldots P$, where $\vec{X}^{(0)}, \vec{d}$ are the input vectors with dimensions H_0 and the output with dimensions H_{k+1}. This sample should be approximated. The result of the solution of the problem must be some mathematical mechanism that would obtain any value of the vector function $\vec{Y}^{(K+1)}\left(\vec{X}^{(0)}\right)$, which is given in the form of a training sample, by a given input vector in the range that is limited by the input.

The classification problem statement is formulated. Let $\vec{X}^*$ is the vector of variables that describe the state of precedents, and let M is the set of class numbers (scenarios). The number of possible scenarios is known, as well as for each scenario (class), subsets of observed state variables (symptoms) are formed. According to the values of the projections of the vector $\vec{X}^*$, the precedent is assigned to one of the possible sets R_m, where $m = 0 \ldots M - 1$. It is necessary to find such an m-th scenario for which the maximum density of the conditional probability distribution of the occurrence of $\vec{X}^*$ in the precedent under the m-th scenario is

$$\exists!\ m^* \in C_m\left(\rho\left(\vec{X}_m^*|R_m\right)\right)(m = 0 \ldots M - 1) : \rho\left(\vec{X}_m^*|R_m\right) \rightarrow max, \qquad (2)$$

here $C_m\left(\rho\left(\vec{X}_m^*|R_m\right)\right)$ – the set of m-th indices of the density distribution of the conditional probability of the occurrence of $\vec{X}^*$ in the precedent in the m-th scenario.

The logistic function argument was used as the target function values. The use of logistic regression makes it possible to predict the likelihood of an event occurring by the values of a certain number of attributes. To do this, we introduce the so-called dependent variable y, taking only one of the two values - as a rule, numbers 0 (event did not occur) and 1 (event occurred), and independent variables (signs) - $\vec{X}^*$, based on the values of which you need to calculate the probability of accepting one or the other value of the dependent variable.

It is assumed that the probability of an event $y = 1$ is:

$$Pr\left\{y = 1|\vec{X}^*\right\} = f(z),$$

here $z = \theta^T x = \theta_1 x_1 + \ldots + \theta_n x_n$, $\theta_i x$ – he vectors of the independent variables values $x_1, \ldots, x_n$ i and the parameters (regression coefficients) are real numbers $\theta_1, \ldots, \theta_n$, respectively, and $f(z)$ is the so-called logistic function (sometimes also called sigmoid or logit function):

$$f(z) = \frac{1}{1 + e^{-z}}.$$

Since y takes only values 0 and 1, then the probability of the second possible value is:

$$\Pr\{y = 0|x\} = 1 - f(z) = 1 - f(\theta^T x).$$

For brevity, the distribution function y for a given x can be written as follows:

$$\Pr\{y|x\} = f(\theta^T x)^y \left(1 - f(\theta^T x)\right)^{1-y}, y \in \{0, 1\}.$$

To solve the object state classification problem, we used a unidirectional multilayer artificial neural network (UMN) to study and a radial basis artificial neural network (RBN). Let's look at their architecture and the specificities of learning methods to solve the problem under consideration.

3 Structure and Training Method of Unidirectional Multilayer Artificial Neural Networks

The simplest UMN with one hidden layer (K = *1*) is shown in Fig. 1. Here $\left\{Y_{ph}^{(0)}\right\}$ – is the set of input data, $\left\{Y_i^{(k)}\right\}$ – is the set of output data of the k-th layer; k is the layer number, k = 1 … (K + 1), K is the number of hidden layers; p = 1 … P, P is the number of analogues; $\left\{w_{ij}^{(k)}\right\}$ is the set of weights of the k-th layer; i is an element of the k-th layer; j is the element of the (k–1)-th layer; H_0 is number of network inputs; H_1 is number of hidden layer neurons; H_2 is the number of network outputs.

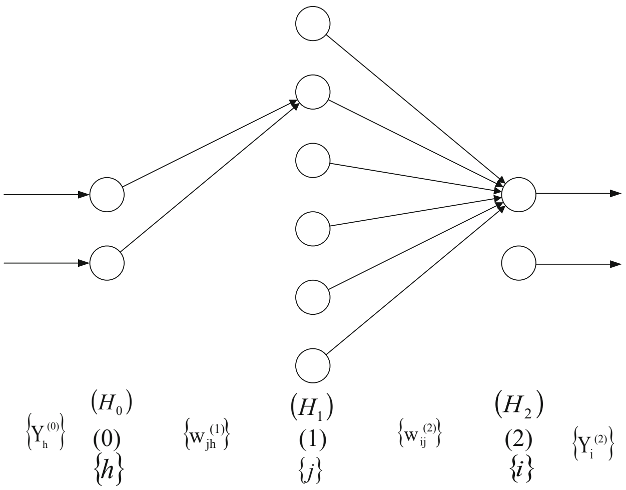


Fig. 1. UMN structure

Analytical representation of the required functions for a unidirectional multilayer network, according to Fig. 1, has the following structure:

$$Y_i^{(2)} = f\left(s_i^{(2)}\right), s_i^{(2)} = w_{i0}^{(2)} + \sum_{j=1}^{H_1} w_{ij}^{(2)} Y_j^{(1)}, i = 1 \ldots H_2,\ j = 1 \ldots H_1$$

$$Y_j^{(1)} = f\left(s_j^{(1)}\right), s_j^{(1)} = w_{j0}^{(1)} + \sum_{h=1}^{H_0} w_{ih}^{(1)} Y_h^{(0)}, h = 1 \ldots H_0,$$

here $f(s) = th(\beta s) = \frac{e^{\beta s} - 1/e^{\beta s}}{e^{\beta s} + 1/e^{\beta s}}$ is the chosen activation function,

$f_s^{'} = \beta[1 - f^2(s)]$ is the derivative of the activation function.

Stable (robust) statistical methods of UMN parameters estimation were used as methods of UMN training, detailed description of which was published in the monograph [20]. These methods provide robustness and informative parameters of systems (processes) statistical models with a priori uncertainty of the input data.

The model parameters were denoted by the vector M – the vector of random numbers of dimension m, and the data of measurements by the vector D – the vector of random numbers of dimension H0 + HK + 1. If p measurements of vector D were observed, then the posterior probability density distribution of vector M after P (p = 1 … P) measurements would be equal to

$$\rho(M|D_P) = \frac{\rho(D_P|M)\rho(M)}{\rho(D_P)},$$

here $\rho(D_P|M) = \prod_{p=1}^{P} \rho(D_p|M)$, $\rho(M) = \prod_{n=1}^{m} \rho(M_n)$, $\rho(D_P) = \prod_{p=1}^{P} \rho(D_p)$ is constant, $m = \sum_{k=1}^{K+1} (H_{k-1} + 1)H_k$ – for UMN ($m = H_0 H_1 + (H_1 + 1)H_2$ – for radial basis neural networks), $m \le \mathrm{P} H_{k+1}$.

The vector $\widehat{M}$ was found as a solution to the problem

$$\widehat{M}_{t+1} = \arg\ \inf_{M \in D_M} E(M|D_P),$$

where function E is a scalar convolution of objective functions by $f_i \equiv Y_i^{(2)}$, $x_h = Y_h^{(0)}$. has the form

$$E = \frac{1}{2PI} \sum_{p=1}^{P} \gamma^{P-p} \sum_{i=1}^{I} \left\{ f_{fit} \left[4 \left(\frac{\Delta_{f,p}}{f_i^*} \right)^2 \left(1 + \sigma_{f,p}^0 \right)^{-2} \right] + \beta_{t+1} f_{fit} \left[\left(\sigma_{f,p}^0 \right)^2 - 1 \right] \right\},$$

for which $\beta_{t+1} = \frac{\sigma_D^2}{\sigma_M^2}, \sigma_D^2, \sigma_M^2$ – the variance of random numbers, t = 1 … T – the number of the epoch of training, $I = H_k$, γ – the level of significance (γ = [0.95, 0.99]), $f_{fit}(d_i) = 1 - exp\left[\left(-\frac{L_{fit}}{4}\right) d_i\right]$, $L_{fit} \ge 4 (d_i > 0)$, $\left(\overset{\circ}{\sigma}_{f_i}\right)^2 = \frac{\left(\sigma_{f_i}\right)^2}{\left(\sigma_{f_i}^*\right)^2}$, $\sigma_{f_i}^2 = \beta^4 \left[1 - f^2\left(s_i^{(2)}\right)\right]^2$

$\sum_{j=1}^{H_1}\left\{\left(w_{ij}^{(2)}\right)^2\left[1-f^2 s_j^{(1)}\right]^2\sum_{h=1}^{H_0}\left[\left(w_{ih}^{(1)}\right)^2\left(\sigma^*_{F_h^{(0)}}\right)^2\right]\right\}$, $\left(\sigma^*_{f_i}\right)^2=\left(\frac{2l_f}{f_{i,max}-f_{i,min}}\right)^2\left[\frac{\overset{\circ}{\Delta}_{f_i}}{300}f_{i,max}\right]^2 n_\alpha$ $(n_\alpha=1)$, $\overset{\circ}{\Delta}_{f_i}=\frac{\Delta_{f_i}}{f_{i,max}}100\%$.

The solution – approximate functions of the form $Y_i^{(k+1)}\left(\vec{Y}^{(0)}\right)$ – is found by stochastic approximation on the basis of the tiered gradient conjunction method [20]. The training used a regularization algorithm that implements interrupts in the iterative process in cases of accumulation of calculation errors [20].

Training and moment coefficients were dependent on the following:

$$\eta_{ij}^{(k)}=\rho_{ij}^{(k)}(t)\eta_{ij}^{(k)}(t-1),\eta_{ij}^{(k)}(0)=\eta_{max},$$

here $\rho_{ij}^{(k)}(t)=-\frac{1}{\left.\frac{\partial S_{ij}^{(k)}}{\partial w_{ij}^{(k)}}\right|_t}\left(1-exp\left(h\left.\frac{\partial S_{ij}^{(k)}}{\partial w_{ij}^{(k)}}\right|_t\right)\right)$, $\alpha_{ij}^{(k)}(t)=-\frac{S_{ij}^{(k)}(t)}{S_{ij}^{(k)}(t)-S_{ij}^{(k)}(t-1)}$, the implementation of the condition $E_{\tau+1}>E_\tau, E_\tau<E_{\tau-1}$ is given in the form:

(a) if $\overset{\circ}{E}_\tau\le K_w$ and $S_{ij}^{(k)}(t)S_{ij}^{(k)}(t-1)>0$ then $\alpha_{ij}^{(k)}(t)=\alpha_{mid}$;
(b) if $\overset{\circ}{E}_\tau\le K_w$ and $S_{ij}^{(k)}(t)\left(w_{ij}^{(k)}(t)-w_{ij}^{(k)}(t-1)\right)\alpha_{ij}^{(k)}(t)<0$ then

$$w_{ij}^{(k)}(t+1)=w_{ij}^{(k)}(t)+\mu(t)\left\{\eta_{ij}^{(k)}(t)r_{ij}^{(k)}(t)-\upsilon(t)\alpha_{ij}^{(k)}(t)\left[w_{ij}^{(k)}(t)-\widehat{w}_{ij}^{(k)}\right]\right\}+\widetilde{w}_{ij}^{(k)}(t+1),$$

where the quantities $\widehat{w}_{ij}^{(k)}$ were determined according to the method of simulating the movement of bee colonies by the formula

$$\widehat{W}=\arg\inf_{\substack{W\in D_w,\\ \tau\in[1,t-1]}} E(W,\tau), W=\left\{w_{i,j}^{(k)}\right\}$$

The following parameter values were accepted h = 0.95, K$_w$ = 1.04, $\rho_{ij}^{(k)}(t)=[0.7,1.05]$, $\eta_{ij}^{(k)}(t)=[0.01,0.4]$, $\alpha_{ij}^{(k)}(t)=[-0.8,0.8]$, α_{mid} = 0.005, α_{opt} = 0.5.

When pairwise comparison of formal mathematical models, the variance of the signal variance, which characterizes the robustness of a particular model, is estimated:

$$D_{Y_i,dB}=10\lg\left(\frac{D_{Y_i}^{(\lambda)}}{D_{Y_i}^{(0)}}\right), \text{ decibel; } \lambda=1,2.$$

The residual variance values for each of the compared formal mathematical models were then used as estimates of the signal dispersions.

4 Structure and Training Method of Radial Basis Artificial Neural Networks

The structure of the simplest RBN with one hidden layer ($K = 1$) is similar to that shown in Fig. 1. We have the following designations:

$\vec{Y}^{(k)} = \left[Y_1^{(k)}, \ldots, Y_{H_1}^{(k)}\right]^T$, $k = 0, 1, 2$, is vector of the input data of the k-th layer;

$\vec{c}_j = \left[c_{j1}, c_{j2}, \ldots, c_{jH_0}\right]^T, j = 1 \ldots H$, is vector of coordinates of centers of activation function for hidden layer neurons;

$\vec{\sigma}_j = \left[\sigma_{j1}, \sigma_{j2}, \ldots, \sigma_{jH_0}\right]^T$, $j = 1 \ldots H_1$, is vector that specifies the width of the window of the activation function of the j-th neuron of the hidden layer;

$\varphi_j = \left(\vec{Y}_p^{(0)}, \vec{c}_j, \vec{\sigma}_j\right) = exp\left(-\frac{1}{2}\sum_{h=1}^{H_0} Z_{pjh}^2\right) \equiv \varphi_{pj}$ is radial-base activation function of the hidden layer neuron, $Z_{pjh} = \frac{Y_{ph}^{(0)} - c_{jh}}{\sigma_{jh}}$;

w_{ij} is the weight of the connection between the i-th neuron of the source layer and the j-th neuron of the hidden layer (here, according to the notation in Fig. 1, it is implied that $w_{jh}^{(1)} = e_{jh} = 1, w_{ij}^{(2)} \equiv w_{ij}$).

The analytical representation of the desired functions for the radial basis network in accordance with Fig. 1 has the following structure:

$$Y_i^{(2)} = f\left(s_i^{(2)}\right), s_i^{(2)} = w_{i0}^{(2)} + \sum\nolimits_{j=1}^{H_1} w_{ij}^{(2)} Y_j^{(1)}, i = 1\ldots H_2,\ j = 1\ldots H_1;$$

$$Y_j^{(1)} = \varphi\left(s_j^{(1)}\right), s_j^{(1)} = \frac{1}{2}\sum\nolimits_{h=1}^{H_0} \left(z_{jh}\right)^2, h = 1\ldots H_0,$$

where $\varphi(s) = \exp(-s)$ – is the activation function selected, $\varphi_s^{'}(s) = -\varphi(s)$ – s a derivative of the activation function.

A hybrid algorithm was used to train RBNs when the number of training pairs significantly exceeds the number of neurons in the hidden layer $P \gg H_1$. The learning process is divided into two stages:

- selection of linear network parameters (weights of the output layer) based on the use of the pseudo inversion method;
- adaptation of nonlinear parameters of activation functions (centers $\vec{c}_j$ i width $\vec{\sigma}_j$ for these functions).

Both stages are closely intertwined. In the presence of P training pairs $\left(\vec{Y}_p^{(0)}, \vec{d}_p\right)$, $p = 1\ldots P$, and fixation of specific values of centers and width of activation functions (in the first moment these will be initial values) we get a system of equations ($P \gg H_1$):

$$\Phi\vec{w}_i = \vec{d}_i, i = 1\,..\,H_2,$$

where $\Phi = \left[\varphi_{pj}\right], p = 1\ldots P, j = 0\ldots H_1$;

$$\varphi_{p0} = 1, \vec{w}_i = \left[w_{i0}, w_{i1}, \ldots, w_{iH_1}\right]^T, \ \vec{d}_i = \left[d_{0i}, d_{1i}, \ldots, d_{pi}\right]^T.$$

Weight vector $\vec{w}_i$ can be defined in one step by a pseudo inversion of the matrix Φ, i.e. $\vec{w}_i = \Phi^+ \vec{d}_i$, where Φ^+ – pseudo-inverted to Φ matrix. In practice, the pseudo inversion is calculated using eigenvalue decomposition. (SVD - decomposition), according to which $\Phi = USV^T$ [17].

In the second stage of the algorithm, when the values of the output weights are fixed, the excitation signals are passed over the network to the output layer, which allows to calculate the magnitude of the error for the sequence of vectors $\left\{\vec{Y}_p^{(0)}\right\}$. There is a return to the hidden layer (back propagation) after that. The gradient vector of the choice function with respect to specific centers $\vec{c}_j$ i ширини $\vec{\sigma}_j$ is determined by the error value $\left\|\vec{Y}^{(2)} - \vec{d}\right\|_{L_2}$.

The algorithm of forming the "coverage area" by radial basis functions with consideration of K-neighbors $\sigma_{jh}^2 = \sum_j = \frac{1}{K}\sum_{k=1}^{K}\sum_{h=1}^{H_0}\left(c_{jh} - c_{kh}\right)^2, k = 1\ldots K \quad K \in [3, 5]$ was applied to determine the values of $\sigma = \left\{\sigma_{jh}\right\}$, which will reduce the duration of training RBN [17].

Refining the values of the covariance matrix $\sigma = \left\{\sigma_{jh}\right\}$ and the coordinates of the centers $C = \left\{c_{jh}\right\}$ completes the next training cycle. Correction of the elements of the covariance matrix is carried out according to the formula (a recurrent learning algorithm is used, which corresponds to the stochastic approximation method based on the ravine conjugate gradient method and ensures convergence $\sigma_{jh}(t) \underset{t\to\infty}{\longrightarrow} \widehat{\sigma}_{jh}$ with probability P = 1):

$$\sigma_{jh}(t+1) = \sigma_{jh}(t) + \mu(t)\left\{\eta_{jh}(t)r_{jh}(t) + v(t)\alpha_{jh}(t)\left[\sigma_{jh}(t) - \widehat{\sigma}_{jh}\right]\right\} + \widetilde{\sigma}_{jh}(t+1),$$

where the projections of the search direction vector $r_{jh}(t)$ were in accordance with the conjugate gradient method.

Repeated repetition of both steps leads to complete and fast learning of the network, especially if the initial values of the parameters of the radial basis functions are close to optimal values.

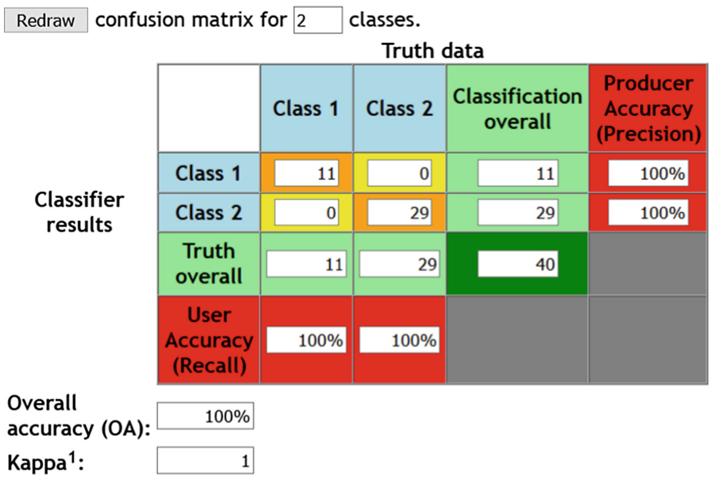
In practice, the selected steps affect the adaptation of parameters to varying degrees. As a rule, the SVD algorithm functions faster (it finds the local minimum of the function in one step). To balance this disproportion, one refinement of linear parameters is usually accompanied by several cycles of nonlinear parameter adaptation.

5 An Example of the Use of Artificial Neural Networks for the Classification of Patients' Conditions in Medical Monitoring Systems

Two samples were considered, which represent the laboratory data of patients on two types of diseases: liver disease and urological diseases.

The Urological Data Sample contains 47 state variables, whose values were of three types: real numbers, booleans, and enumerators. The training sample consisted of data on 30 patients, the test sample contained data on 10 patients. The result of the classification is to determine the patient's condition (healthy or ill). For classification, the RBN with logistic function was used. RBN structure contains 47 neurons of the input layer corresponding to the number of state variables, 26 neurons of the hidden layer, and 2 neurons of the output layer corresponding to the two classified states. As a result of training, the error was 0.122 and the standard deviation of 0.0024. The recognition result on the control sample is 100%.

As a result of training and network testing of the Urological Data Sample, the confusion matrix and ROC curves were constructed to analyze the quality of the classification (Figs. 2 and 3).



Redraw confusion matrix for 2 classes.

Truth data

Classifier results

	Class 1	Class 2	Classification overall	Producer Accuracy (Precision)
Class 1	11	0	11	100%
Class 2	0	29	29	100%
Truth overall	11	29	40	
User Accuracy (Recall)	100%	100%		

Overall accuracy (OA): 100%

Kappa[1]: 1

Fig. 2. Confusion matrix of the Urological Data Sample classification problem

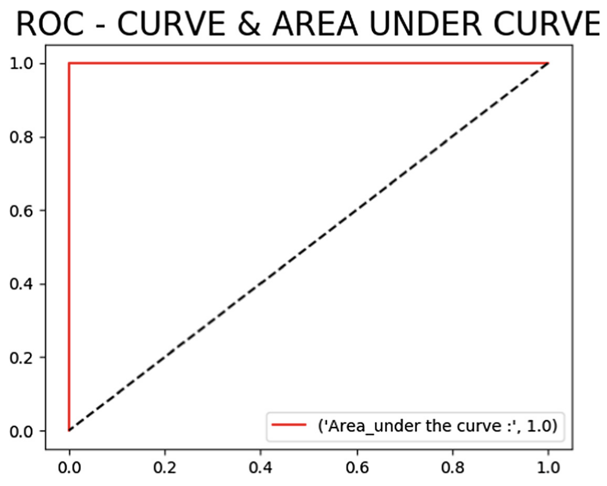
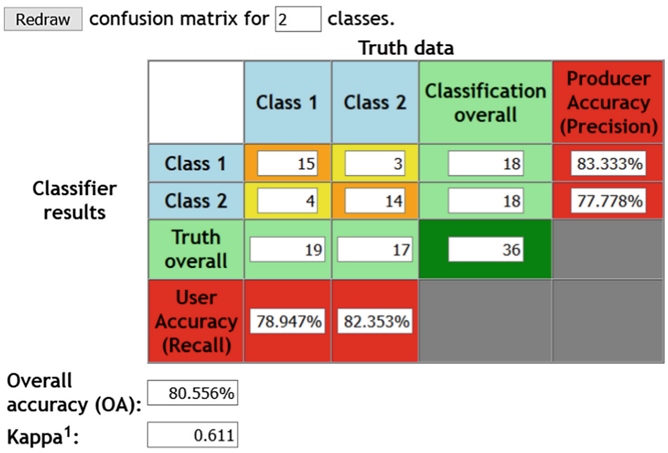


Fig. 3. ROC-curve of the Urological Data Sample classification problem

The liver disease sample contains 10 state variables, the values of which were of three types: real numbers, booleans, and enumerators. The study sample consisted of data on 550 patients, and the test sample contained data on 40 patients. RBN structure contains 10 neurons of the input layer corresponding to the number of state variables, 540 neurons of the hidden layer, and 2 neurons of the output layer corresponding to the two classified states. As a result of training, the error was 0.0107 and the standard deviation of 0.0185. The recognition result on the control sample is 80%.

As a result of training and network testing of the liver disease sample, the confusion matrix and ROC curves were constructed to analyze the quality of the classification (Figs. 4 and 5).



	Class 1	Class 2	Classification overall	Producer Accuracy (Precision)
Class 1	15	3	18	83.333%
Class 2	4	14	18	77.778%
Truth overall	19	17	36	
User Accuracy (Recall)	78.947%	82.353%		

Overall accuracy (OA): 80.556%

Kappa^1: 0.611

Fig. 4. Confusion matrix of the liver disease sample classification problem

A confusion matrix, also known as an error matrix, [25] is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). ROC-curve (receiver operating characteristic) – a graph that allows you to evaluate the quality of binary classification displays the relationship between the share of objects from the total number of characteristic carriers correctly classified as bearing the attribute (called the sensitivity of the classification algorithm) and the share of objects from the total number of objects that do not carry the attribute, erroneously classified as bearing a trait [22–24].

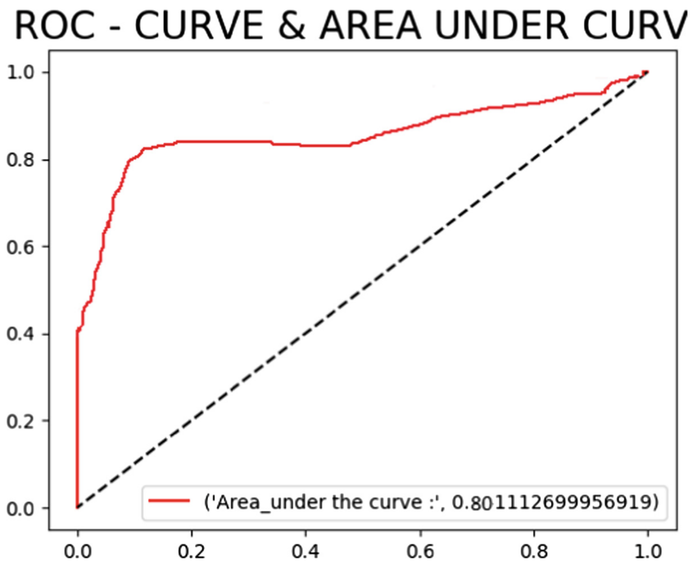


Fig. 5. ROC-curve of the liver disease sample classification problem

References

1. Bakumenko, N.: Application of the c-means fuzzy clustering method for the patient's state recognition problems in the medicine monitoring systems. In: Bakumenko, N., Strilets, V., Ugriumov, M. (eds.) CEUR Workshop Proceedings of 3rd International Conference on Computational Linguistics and Intelligent Systems, COLINS 2019, vol. 2362, pp. 218–227 (2019)
2. Zhang, H.: The Optimality of Naive Bayes. American Association for Artificial Intelligence (2004)
3. Metsis, V., Androutsopoulos, I., Paliouras, G.: Spam filtering with Naive Bayes—which Naive Bayes? In: Third Conference on Email and Anti-Spam, 27–28 July 2006, Mountain View, California USA (2006)
4. Rennie, J., Shih, L., Teevan, J., Karger, D.: Tackling the poor assumptions of Naive Bayes classifiers. In: Proceedings of the Twentieth International Conference on Machine Learning, Washington D.C. (2003)
5. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **IT-13**(1), 21–27 (1967)
6. Garcia, S., Derrac, J., Cano, J., Herrera, F.: Prototype selection for nearest neighbor classification: taxonomy and empirical study. IEEE Trans. Pattern Anal. Mach. Intell. **34**(3), 417–435 (2012)
7. Ho, T.K.: Random decision forests. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, pp. 278–282, 14–16 August 1995
8. Denisko, D., Hoffman, M.M.: Classification and interaction in random forests. Proc. Natl. Acad. Sci. U.S.A. **115**(8), 1690–1692 (2018)
9. Riddick, G., Song, H., Ahn, S., Walling, J., Borges-Rivera, D., Zhang, W., Fine, H.A.: Predicting in vitro drug sensitivity using Random Forests. Bioinformatics (Oxford, England) **27**(2), 220–224 (2011)

10. Touw, W.G., Bayjanov, J.R., Overmars, L., Backus, L., Boekhorst, J., Wels, M., van Hijum, S.A.: Data mining in the Life Sciences with Random Forest: a walk in the park or lost in the jungle? Brief Bioinform. **14**(3), 315–326 (2013)
11. Basu, S., Kumbier, K., Brown, J.B., Yu, B.: Iterative random forests to discover predictive and stable high-order interactions. Proc. Natl. Acad. Sci. U.S.A. **115**(8), 1943–1948 (2018)
12. Breiman, L.: Random forest, machine learning. In: Proceedings of the Thirteenth International Conference, vol. 45, pp. 5–32 (2001)
13. Strano, M., Colosimo, B.M.: Logistic regression analysis for experimental determination of forming limit diagrams. Int. J. Mach. Tools Manuf **46**(6), 673–682 (2006)
14. Cramer, J.S.: The origins of logistic regression. Tinbergen Inst. **119**, 167–178 (2002)
15. Kaminski, B., Jakubczyk, M., Szufel, P.: A framework for sensitivity analysis of decision trees. Central Eur. J. Oper. Res. **26**(1), 135–159 (2017)
16. Karimi, K., Hamilton, H.J.: Generation and interpretation of temporal decision rules. Int. J. Comput. Inf. Syst. Ind. Manag. Appl. **3** (2011)
17. Gao, W., Zhou, Z.-H.: On the doubt about margin explanation of boosting. Artif. Intell. J. **203**, 1–18 (2013)
18. Freund, Y., Schapire, R.E.: A short introduction to boosting. J. Jpn. Soc. Artif. Intell. **14**(5), 771–780 (1999)
19. Kegl, B.: The return of AdaBoost. MH: multi-class Hamming trees. In International Conference on Learning Representations, 2014
20. Системное совершенствование элементов сложных технических систем на основе концепции обратных задач [Текст]: монография/ В. Е. Стрелец, А. А. Трончук, Е. М. Угрюмова и др.; под общ. ред. М. Л. Угрюмова. – Х. : Нац. аэрокосм. ун-т им. Н. Е. Жуковского « Харьк. авиац. ин-т » , 148 с (2013)
21. Угрюмова Е.М. Обучаемые искусственные нейронные сети в построении формальных математических моделей систем при априорной неопределенности данных/Е.М. Угрюмова// Вісник Харківського національного університету: зб. наук. пр. Сер. Мате матичне моделювання. Інформаційні технології. Автоматизовані системи управління. – 2010. – Випуск 13 (№890). – С. 237–253 (2010)
22. Powers, D.M.W.: Evaluation: from precision, recall and f-measure to ROC, informedness, markedness & correlation. J. Mach. Learn. Technol. **2**(1), 37–63 (2011)
23. Flach, P., Hernandez-Orallo, J., Ferri, C.: A coherent interpretation of AUC as a measure of aggregated classification performance. In: Appearing in Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA (2011)
24. Bi, J., Bennett, K.P.: Regression error characteristic curves. In: Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC (2003)
25. Stehman, S.V.: Selecting and interpreting measures of thematic classification accuracy. Remote Sens. Environ. **62**(1), 77–89 (1997)