Algorithms and methods for extracting knowledge about objects defined by arrays of empirical data using ANN models

A A Arzamastsev¹, N A Zenkova² and N A Kazakov¹

¹ Voronezh State University, Department of Mathematical and Applied Analysis, 394018, Voronezh, University Square 1, Russia.
² Tambov State University, Department of Mathematical Modeling and Information Technology, 392000, Tambov, International Str., 33, Russia.

E-mail: arz_sci@mail.ru

Abstract. Most Data Mining techniques are based on traditional methods of constructing mathematical models and their parametric identification. The article considers the technology of extracting knowledge about objects, given by empirical data arrays. It is based on detecting some characteristic features in the object generating these data. They can include dynamic and static characteristics; delays in various channels; inertia of channels or inertia combined with delay; different channel sensitivity; bypass and certain classes of nonlinearities. We also considered the objects that are characterized by a constant flow of new data, which makes it necessary to retrain the model. As a tool for obtaining knowledge in this work, the apparatus of artificial neural networks (ANNs) is used. It includes a structure builder, which allows you to quickly change the structure of ANN models and machine training algorithms. This enables you to successfully solve parametric identification problems. Algorithms and methods of knowledge extraction have been developed for these tasks. Examples that confirm the possibility of their use are given.

1. Introduction

At present, in various fields of science, there is a significant increase in the number of objects to be described and studied by mathematical methods. This results from the widespread introduction of information technology and digitalization in different areas of activity: industry, transport, communications, a wide range of natural, social and technical sciences. In many cases, the mathematical and computer models of an object must be built relying only on the available empirical data (information about its inputs and outputs).

In natural sciences, similar problems arise when studying fundamentally new objects, whose physical laws and mechanisms of operation are not known. In the objects of the social sphere, a similar situation arises due to the weak formalization of this subject area and a significant amount of empirical data.

The construction of a mathematical model adequate to the object, in its turn, simplifies its study and allows us to obtain new knowledge about its mechanisms, relationships, etc.

In the scientific literature, such a problem is called the problem of structural and parametric identification of the "black box", Structural System Identification and Data Mining ("extraction" of knowledge from data).
When solving such problems, the authors usually proceed from the existing mathematical propositions and methods, for example, the classical theorems of Weierstrass and Kolmogorov [1, 2], use methods of correlation, regression, cluster and factor analysis as well as other technology. In this case, possible properties of an object that generates arrays of empirical data are practically not considered.

The approach proposed in this paper is based primarily on the detection of certain characteristic features in an object based on its empirical data: dynamic and static characteristics; delays in various channels; inertia or inertia combined with delay; different channel sensitivities; bypass and certain classes of nonlinearities. Particular attention is paid to objects that are characterized by a constant flow of new data, which makes it necessary to retrain the model.

As a tool for obtaining knowledge in this work, the apparatus of artificial neural networks (ANNs) is used, which includes a structure builder and machine training algorithms. The presence of these components makes it possible to successfully solve the problems of structural and parametric identification.

Note that the possibility of using artificial neural networks for modeling objects was considered in various works, for example [3], and in publications [4, 5] the schemes using ANN apparatus are also applied for solving Data Mining problems.

We call the mathematical and computer models, obtained using ANN based on empirical data, ANN models. After the structural and parametric identification of such a model is carried out, it can be studied by both mathematical and computer methods, which can reveal new knowledge about the object.

2. Statement of the problem

The representation of an object for the purposes of its modeling is shown in figure 1. The vector of input coordinates $X$ is user-independent input parameters, vector $U$ is the input variable parameters with which the user can affect the object (control it), vector $Y$ are the properties of the object which are of interest to the user. It should be noted that the object itself does not "distinguish" between $X$ and $U$, but such a representation is convenient from the user's point of view.

In our case, the object shown in figure 1 is presented by a rectangular table $A[(n + m + k), z]$, where $z$ is the number of entries in the table. There are the following features: the number of entries $z$ is arbitrary, it can be either excessive or insufficient for solving the problem. Measurements of array elements $a[i, j]$ are made with a known error, which may be different for all inputs and outputs.

It is necessary to find the optimal (in a certain sense) system of relationships that characterizes the object $Y = F(X, U, P)$. Here $F$ is an operator, $P$ is a vector of optimizable parameters. Such a system of relations of an object represents its mathematical model, which can be further studied. While making computational experiments with the model, new knowledge can be extracted.

![Figure 1. Scheme of object representation for modeling.](image-url)
Note that various mathematical tools are used as operator $F$ in mathematical modeling, such as systems of ordinary differential equations, partial differential equations, methods of mathematical statistics, etc. In our work, we use the apparatus of artificial neural networks (ANNs) as a tool for modeling and extracting knowledge about an object given by an array of empirical data. Mathematical (computer) models obtained using ANNs based on empirical data will be called ANN models. In this case, the vector of weighting coefficients of neurons $W$ acts as a vector of optimizable parameters.

The choice of ANN apparatus is due to the following reasons: 1) as a rule, software products that implement ANN models include a structure builder, which allows you to quickly adapt the structural scheme of the model without changing the method of its parametric identification; our programs are as such [6,7]; 2) they contain ready-made and well-debugged machine training algorithms, which makes it possible to successfully solve parametric identification problems; 3) the availability of a significant number of specialized libraries for Python, Mathematica, MATLAB languages (for example, Theano, TensorFlow, Keras for Python); 4) opportunity for parallelization of computations, high adaptability to empirical data and tolerance to their errors.

3. The basic tasks and algorithms
Automatic recognition of the dynamic or static nature of an object based on empirical data alone is generally a difficult problem. However, if the table has a column corresponding to time, then the task is greatly simplified.

In general, the algorithm for operation with dynamic objects looks like this: 1) identify a data column responsible for time characteristics of the object; if there is one, then go to step 2, if not, the algorithm finishes its work (the object is static), we go to the algorithm for analyzing static data; 2) find out whether the dynamic sequence is a time series, if yes, then determine the step of such a series, if not, we use any of the interpolation methods (for example, spline interpolation) to obtain from the original table the one which presents time series, then we turn to the time series analysis algorithm.

The algorithm for operation with static data is as follows: 1) create a simple ANN model with a known number of inputs ($n + m$) and outputs $– k$, as such a model can be an ANN with one neuron in a hidden layer; 2) run one of the machine training algorithms and evaluate the optimization result; several machine training algorithms can be used; the result of optimization is often an error in the operation of the ANN model with the optimal values of the coefficients in comparison with empirical data; if the value of the error has an acceptable level, then it is considered that the ANN model is adequate, it can be used to obtain new knowledge about the object, the operation of the algorithm ends there; if the amount of the error is large, then go to step 3; 3) build up the structure of the network with a possible change in the transfer functions of neurons (algorithms for building up structures and stopping criteria were proposed in our previous works); go to step 2.

It should be noted that this algorithm was the most tested in our previous works; it also allows parallelization, which significantly reduces the time to solve the problem.

The algorithm for operation with time series using ANN models includes several stages: 1) computation of cross-correlation functions for time series and the formation of conclusions about the presence of delays along various channels; 2) checking the possible inertia of the channels by reducing the inertia of high dimension to the inertia of the first and second orders and pure delay, which leads to the transformation of the initial data table; 3) the choice of $g$ – the order of approximation for g-point scheme of the ANN model; 4) the transformation of the initial data table in accordance with the selected value of $g$ and delays for individual channels; 5) launching one or more machine training algorithms and evaluating the optimization result; if necessary, we build up the structure of the ANN model; if the value of the error has an acceptable level, then it means that the ANN model is adequate, it can be used to obtain new knowledge about the object; if the value of the error is great, then we change the value of $g$ and go to step 3.

In many cases, the object is constructed in such a way that it is constantly enlarged by a flow of new data i.e. the lower border of table $A – z$ is movable. This situation occurs, for example, when a server collects data from a large number of sources – terminals. Obviously, in this case, we need: to
estimate the time of the initial accumulation of data; "retrain" the model and determine the optimal time for the "retraining" period; assess the stationarity/non-stationarity of an object and its ANN model – the problems that set up new tasks for data analysis.

In general, the algorithm for constructing an ANN model and data analysis for this case is as follows: 1) estimation of the period of primary data accumulation in the table, preceding the first training of an ANN model; such an estimate can be carried out on the basis of the integral rate of data accumulation and Kolmogorov theorem, which allows calculating the number of degrees of freedom of the required ANN model; 2) approximation of the training process (decreasing the error in the training process) by a first-order differential equation and determining the rate coefficient of this process; 3) the implementation of a trial (on a partial sample) retraining the ANN model after a certain specified period of time; if the value of the speed coefficient changes insignificantly, then the object is stationary and retraining within this time is not required; if the training speed coefficient changes significantly in comparison with the previous value, the object is nonstationary and retraining is required; again go to step 3. This algorithm is infinite and works as long as the object itself is working.

Examples of the application of these algorithms or calculation technology are given in sections 3–5 of this article.

4. Uses of the approach for a static object

Let's consider an example of extracting knowledge from a data array using the above technique. The simulated object is a biotechnological process in which the waste of ethanol production is utilized with generation of bacterial biomass. Real data are taken from the experience of operating the process at JSC "Bio-Chem", Tambov region. This technological process is known to operate in a continuous static mode. We chose this object for testing the methodology due to a significant amount of empirical data at our disposal.

The complete data file is a rectangular matrix containing 1000 rows, each of which characterizes the state of the bioreactor, and 18 columns, the elements of each of which characterize one of the parameters of the state of this process: the consumption of ethanol production waste – G, m³/h, its concentration – S₀, kg/m³, temperatures in the reactor sections – T₁–T₄, °C, the acidity of the medium in the reactor sections – pH₁–pH₄, and output parameters: biomass concentrations in the sections – x₁–x₄, kg/m³ and substrate concentrations in the reactor sections – s₁–s₄, kg/m³.

It is necessary, on the basis of these data, to find a system of relationships in the form of a mathematical model \( \mathbf{Y} = \mathbf{F}(\mathbf{X}, \mathbf{U}, \mathbf{P}) \).

Since this work deals with the development of a model that has an ANN model as an intellectual core, the previous equation can be rewritten as \( \mathbf{Y} = \mathbf{F}_s(\mathbf{X}, \mathbf{U}, \mathbf{W}) \), where, \( \mathbf{X} \) is a vector of input coordinates of the object, \( \mathbf{U} \) is a vector of variable parameters (control actions), \( \mathbf{Y} \) is a vector of output coordinates, numerically expressing the object properties of interest; \( \mathbf{W} \) is a vector of weight coefficients of the ANN model, the dimension of which also determines the number of its degrees of freedom for its training according to empirical data; \( \mathbf{F}_s \) is an operator that depends on the structure of the ANN model – \( \mathbf{S} \).

In our case, \( \mathbf{X} \) is a vector with two coordinates \( G \) and \( S₀ \), \( \mathbf{U} \) is a vector with eight coordinates \( T₁–T₄ \) and \( pH₁–pH₄ \), \( \mathbf{Y} \) is a vector with eight coordinates \( x₁–x₄ \) and \( s₁–s₄ \).

The desired ANN model should be nonlinear, since its input values can be presented, for example, by pH (acidity index) of the medium and the temperature. pH is a logarithmic function of the concentration of hydrogen ions. The dependence of the main process parameters on the temperature are expressed by the exponential Arrhenius equation and the kinetics of such a process usually corresponds to the Monod hyperbolic model.

On the basis of empirical data, we developed an ANN model using software presented in [6, 7].

Figure 2 shows the process of training an ANN model having a certain configuration. Here we consider the training of ANN with ten input neurons and three neurons in a hidden layer (linear, quadratic and cubic functions) using the gradient method.
It can be seen from these figures that the root-mean-square error of the model is rapidly decreasing, i.e. a network with this configuration learns quickly based on empirical data. In logarithmic coordinates, the training process looks linear (the correlation coefficient with linear dependence is 0.98), which makes it possible to predict the number of training cycles required to achieve a given root-mean-square error.

Figure 2. The process of training the ANN model (linear, quadratic and cubic functions of neurons) by the gradient method, where δ is a root-mean-square error, m is the number of training cycles: a) in natural values, b) in logarithmic coordinates.

Figure 3 shows the correlations of empirical data and \( x_1 \) values calculated using the ANN model for various ANN structures. From these figures it follows that the best structure of the ANN model, corresponding to empirical data and determined by the mean-square-error, includes ten input neurons, three neurons with linear, quadratic and cubic transfer functions in the hidden layer, and one output neuron. At the same time, the maximum error of the ANN model does not exceed 1%. A scheme of such a model is shown in figure 4.

Thus, we have shown that the use of ANN models makes it possible to obtain an adequate representation of static objects specified by large arrays of empirical data. In the future, you can analyze such a model and get new missing knowledge about the object.

5. Using the approach for analyzing objects given by time series

The object that generates the time series is schematically presented in Figure 1. Let’s assume that the dimension of the vector of input coordinates of such an object is given and equal to \( n \). The output value of the object is only one coordinate – \( y \). Note that in this way we represent a wide class of objects found in various fields of knowledge.

Let’s assume that time variations \( x_1(t), x_2(t), ..., x_n(t) \) on interval \( t \in [t_0, t_k] \) are given. The mathematical model of such an object in operator form usually looks like that:

\[
    y = f(x) = f(x_1, x_2, ..., x_n)
\]  

(1)

In the simplest case, the properties of the channels of such an object are ordinary proportional links with transfer functions \( W_1(p) = k_1, W_2(p) = k_2, ..., W_n(p) = k_n \). In other cases, the channel transfer functions can be more complex. The most common are first and second order aperiodic links with transfer functions \( W(p) = \frac{k}{T_p p + 1} \) and \( W(p) = \frac{k}{(T_1 p + 1)(T_2 p + 1)} \) as well as a pure delay link with a transfer function \( W(p) = e^{-\tau p} \). Here, \( T, T_1, T_2, k, \tau \) are some coefficients dependent on the nature of the object.
Figure 3. Correlation of real data for $x_1$ and calculations of $x_1$ according to the ANN model for a neural network: a) with one linear neuron, b) with one linear and one quadratic neuron, c) with one linear, one quadratic and one cubic neuron, c) with one sigmoidal neuron.

Figure 4. The structure of an ANN model that provides an adequate representation of empirical information for a technological process.
Therefore, in a more general case, equation (1) can be written as:

\[ y = f(x, t) = f(x_1, x_2, ..., x_n, t) \]  

(2)

Since usually it is such a system that generates time series, it is necessary to take into account its properties when analyzing and predicting it. The main of the properties that complicate predicting the values of the series based on its prehistory are: inertia, delay and the presence of a stochastic component associated with both lack of information and measurement errors.

A time series \( \{y(t)\} \) means a set of values of a certain quantity (corresponding to the output of the generating object) at successive points in time, so that:

\[ \{y(t)\} \equiv \{y(t_0), y(t_1), ..., y(t_{i-1}), y(t_i), y(t_{i+1}), ...\} \]

(3)

Considering that \( y \) values are usually measured with a constant step \( h \), i.e.

\[ t_i = t_0 + h \cdot i, i = 1,2, ..., \] row (3) can be written as:

\[ \{y\} \equiv \{y_0, y_1, ..., y_{i-1}, y_i, y_{i+1}, ...\} \]

(4)

In practically important cases, the number of series terms is known – \( k \), i.e.

\[ \{y\} \equiv \{y_0, y_2, ..., y_k\} \]

(5)

When predicting the time series generated by the above object, the following situations may occur:

– only row \( \{y\} \) is observed; time series \( \{x_1\}, \{x_2\}, ..., \{x_n\} \) characterizing the inputs of the object are not observable; in this case, it is necessary to assume that all the information from vectors \( \{x_1\}, \{x_2\}, ..., \{x_n\} \) transformed by the object, is contained in vector \( \{y\} \); let us call the problem of optimal choice of the structure and ANN settings for modeling and predicting a given time series as problem 1;

– vector \( \{y\} \) and all the vectors \( \{x_1\}, \{x_2\}, ..., \{x_n\} \) are observable; let us call the problem of optimal choice of the structure and ANN settings for modeling and predicting a given time series as problem 2;

– vector \( \{y\} \) and part of vectors \( \{x_1\}, \{x_2\}, ..., \{x_n\} \) are observable; in this case, it is necessary to assume that all the information from unobservable input vectors is also contained in vector \( \{y\} \); let us call the problem of optimal choice of the structure and ANN settings for modeling and predicting a given time series as problem 3.

In accordance with Kolmogorov theorems on the representability of functions of several variables by means of superpositions and sums of functions of one variable [1, 2], it can be argued that each continuous function of \( n \) variables defined on the unit cube of \( n \)-dimensional space can be presented as:

\[ f(x_1, x_2, ..., x_n) = \sum_{q=1}^{2n+1} h_q \left[ \sum_{p=1}^{n} \phi_q^p(x_p) \right] \]

(6)

where functions \( h_q(u) \) are continuous, and functions \( \phi_q^p(x_p) \), are both continuous and standard, i.e. they do not depend on the choice of the function \( f \). Thus, for example, the function of two variables \( x_1 \) and \( x_2 \) can be presented as:

\[ f(x_1, x_2) = \sum_{q=1}^{5} h_q \left[ \phi_q(x_1) + \psi_q(x_2) \right] \]

(7)

From the point of view of ANN circuit technique, equation (7) can be presented using the following ANN model (Figure 5).

The number of degrees of freedom of such a model is equal to the number of synaptic relationships in the network, i.e. in this case it is 25.

For function \( n \) variables, the number of synaptic relationships corresponding to formula (6) and the general principle of ANN circuit technique is:

\[ P_1 = n \cdot (2n + 1) + n \cdot (2n + 1) + 2n + 1 = 4n^2 + 4n + 1 \]

(8)
Figure 5. Representation of the function of two variables in the form of an ANN model.

It is obvious that the number of rows in the training set should not be less than the number of degrees of freedom. Considering that the number of rows in the training sample can be expressed as

\[ P_2 = k - n - h + 1 \]

(here \( k \) is the total number of terms of the time series, \( h \) is the prediction range using the ANN model, see Figure 6), we get that the following inequality should be satisfied:

\[ P_1 \leq P_2, \quad 4n^2 + 4n + 1 \leq k - n - h + 1 \]

whose solution will be:

\[ n_1 \leq n \leq n_2, \quad n_1 = \frac{-5 - \sqrt{25 + 16(k-h)}}{8}, \quad n_2 = \frac{-5 + \sqrt{25 + 16(k-h)}}{8} \]

Considering also that \( n \geq 1 \) must be fulfilled, we will finally get the range of acceptable values for the number of inputs of the ANN model:

\[ 1 \leq n \leq \frac{-5 + \sqrt{25 + 16(k-h)}}{8} \]

Figure 6. The scheme interrelating the time series length, the number of the ANN model inputs and the prediction range with the number of rows in the training sample.
In this case, the training sample for an arbitrary, given number of inputs of the ANN model can be defined by matrices:

$$X = \begin{pmatrix} y_1 & y_2 & \cdots & y_n \\ y_2 & y_3 & \cdots & y_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k-n-h} & y_{k-n-h+1} & \cdots & y_{k-h+1} \end{pmatrix}, Y = \begin{pmatrix} y_{n+h+1} \\ y_{n+h+2} \\ \vdots \\ y_k \end{pmatrix} \quad (12)$$

In this case, the initial structure of the network of the ANN model will be as follows: one layer of \( n \) input neurons, a hidden layer of \( n \cdot (2n + 1) \) neurons that implement the transformation in accordance with functions of \( \phi^p_0(x_p) \) neurons, a hidden layer of \( (2n + 1) \) neurons that implement the transformation in accordance with the functions of neurons \( h \), and an output layer consisting of one neuron. This network structure can be further complicated.

To determine the optimal number of inputs of the ANN model – \( g \) for problem 1, it is necessary to determine the minimum of the functional:

$$F[w[S(n)]] = \|\Delta\| = \|Y^{tabl} - Y^{net}(w)\| = \sqrt{\sum_{i=1}^{p} \sum_{f=1}^{N}(Y_{ij}^{tabl} - Y_{ij}^{net}(w))^2} \rightarrow \min_w$$ \quad (13)

Vector \( w \) is determined by the structure of network \( S \), which, in its turn, is related to the number of its inputs \( g \) so that there is a one-to-one correspondence between \( g \) and \( w(S) \). Therefore, vector \( w^* \), depending on the structure of the network and minimizing functional (13), corresponds to the optimal ANN model of a given time series \( \{y\} \) and can be defined as:

$$w^* = \arg\min_{w \in \Omega} F(w)$$ \quad (14)

where domain \( \Omega \) characterizes possible values of the vectors of the weight coefficients corresponding to the structures of the networks for which condition (11) is satisfied. It should be noted that to minimize functional (13) and search for arguments corresponding to its minimum (14), one can use currently well-developed nonlinear programming methods. After the optimal ANN model of the time series \( \{y\} \) is determined, its structure is also defined, which allows us to believe that the optimal number of inputs of such a model – \( g^* \) has been also determined. Figure 6 shows that the number of training vectors will always be \( (n + h) \) less than the length of the original series.

When considering multiple time series (problems 2 and 3), it is necessary to calculate pair correlation coefficients (15) to identify the necessary bias (16) when forming the training sample.

$$r_k = \frac{\sum_{i=1}^{n-k} x_i y_{i+k} - \sum_{i=1}^{n-k} x_i \sum_{i=1}^{n-k} y_{i+k}}{\sqrt{\sum_{i=1}^{n-k} x_i^2 - (\sum_{i=1}^{n-k} x_i)^2/(n-k)} \sqrt{\sum_{i=1}^{n-k} y_i^2 - (\sum_{i=1}^{n-k} y_i)^2/(n-k)}} \quad (15)$$

Next, let us define the required shift:

$$\lambda = \max\{\tau_1, \tau_2, \ldots, \tau_n\} \quad (16)$$

Then, with regard to time shifts, the training sample can be defined by matrices:

$$X = \begin{pmatrix} x_{1,\lambda-\tau_1} & x_{2,\lambda-\tau_2} & \cdots & x_{n,\lambda-\tau_n} \\ x_{1,\lambda-\tau_1+1} & x_{2,\lambda-\tau_2+1} & \cdots & x_{n,\lambda-\tau_n+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,k-\tau_1} & x_{2,k-\tau_2} & \cdots & x_{n,k-\tau_n} \end{pmatrix}, Y = \begin{pmatrix} y_\lambda \\ y_{\lambda+1} \\ \vdots \\ y_k \end{pmatrix} \quad (17)$$

The proposed approach was implemented by us for the short-term forecast of air temperature in [8].
6. Using the approach for an object with a constant flow of new data

Let there be a distributed system for obtaining initial information, including $N$ terminals, each of which is designed to input $n$ independent parameters, each having $m$ levels. The operation of the terminals is organized in such a way that each of them works only part of the time, so that the intervals of their downtime are random variables given by distribution densities $p_1(\tau), p_2(\tau), \ldots, p_N(\tau)$. The time to input information does not depend on the terminal number, but linearly depends on $n$, i.e. $\tau_{\text{input}} = k_1 n$, where $k_1$ is some coefficient of proportionality. Let us assume that the next piece of information is added to the database in the form of one entry, when on the $i$-th terminal ($i = 1, \ldots, N$) the corresponding level $l_b$ ($b = 1, \ldots, m$) is assigned to each independent parameter $x_j$ ($j = 1, \ldots, n$).

It is also taken into account that the transfer of information from any terminal to the database is carried out instantly, because this time is several orders of magnitude less than the time spent by the user to enter information.

Then the average time of one operation cycle of the $i$-th terminal (input + waiting for the next input of information), corresponding to the addition of one entry to the database, can be calculated as:

$$\overline{\tau}_i = k_1 n + \int_0^{\tau_{\text{max}}} \tau p_i(\tau) \, d\tau.$$ 

Therefore, in a period of time $T$ (for example, $T = 24$ h), $r_i$ input cycles can be performed on the $i$-th terminal, corresponding to the addition of $r_i$ entries to the database:

$$r_i = \frac{T}{k_1 n + \int_0^{\tau_{\text{max}}} \tau p_i(\tau) \, d\tau}.$$ 

It should be noted that $r_i$ is an integer resulting from rounding up the value obtained from the equation. Accordingly, the total number of entries of information on the database received from all the terminals in period $T$ will be:

$$R = \sum_{i=1}^{N} r_i = T \sum_{i=1}^{N} \frac{1}{k_1 n + \int_0^{\tau_{\text{max}}} \tau p_i(\tau) \, d\tau}.$$ 

Let us evaluate the period of primary data accumulation preceding the first ANN-model training.

In accordance with Kolmogorov theorems [1, 2], it can be argued that each continuous function of $n$ variables (in ES $n$ corresponds to the number of input independent parameters), given on the unit cube of $n$-dimensional space, can be represented in the form (6), where functions $h_q(u)$ are continuous, and functions $\phi_q^p(x_p)$, are both continuous and standard, i.e. they do not depend on the choice of function $f$.

In fact, this means that the “minimal” structure of the ANN model, that allows approximating the function of $n$ variables, should have the following form: an input layer consisting of $n$ (corresponds to the function variables) neurons, the first hidden layer including $(2n + 1) \cdot n$ functional neurons, the second hidden layer, consisting of $(2n + 1)$ functional neurons, and the output layer, summing the neurons. The neural network above is not fully connected and has $(2n + 1) \cdot n + (2n + 1) \cdot n + 2n + 1 = (2n + 1)^2$ relationships (degrees of freedom when training the ANN model).

It should be noted that Kolmogorov theorems do not have any information about the type of nonlinearity of functions $\varphi$ and $h$. Thus, if in the ANN model, an activation function $f(x) = \frac{1}{1+e^{-ax}}$ is used and a fully-connected network is a multilayer perceptron, then for this case the number of degrees of freedom will be $(2n + 1) \cdot (3n^2 + n + 1)$. Considering that the number of entries in the database should be no less than the number of degrees of freedom, we get an estimate for the primary period of data accumulation:
where $K$ is a parameter characterizing the number of degrees of freedom: $K = (2n + 1)^2$ for an ANN model constructed in accordance with Kolmogorov theorem and $K = (2n + 1)(3n^2 + n + 1)$ for a fully-connected model.

The training process of the ANN model, as in the previous section, is carried out in accordance with formulas (13) and (14).

Functional (13) can be minimized by the following methods: the gradient one, so that the subsequent values of vector $w$ are calculated by formula $w^{t+1} = w^t - h(t)grad(E(w^t))$ and by scanning method. For these methods, the training time estimates can be obtained as follows: $\theta_{\text{grad}} = k_2nR$ and $\theta_{\text{skan}} = k_3R\alpha^N$. Here $k_2$ and $k_3$ are the proportionality coefficients depending on the technical characteristics of the equipment used, and $\alpha$ is the number of partitions of the variable range.

The goal of training an ANN model based on empirical data is to find a suitable structure and coefficients that minimize the deficiency (13). The specified structure summarizes the knowledge obtained on the basis of the data received from the terminals into the database.

The data used for training the ANN model can differ both quantitatively (the number of entries in the database at the time of beginning the training) and qualitatively (dispersion, errors, etc.). Therefore, an equation would be needed to predict the training process of the ANN model. In our works, we used the following differential equation:

$$\frac{dE(t)}{dt} = k_4[E^* - E(t)],$$

with the initial condition: $E(0) = E_0$ (19)

having a solution:

$$E(t) = E^*(1 - e^{-k_4t}) + E_0e^{-k_4t}$$ (20)

where $E(t)$, $E_0$ and $E^*$ are the values of the reduced root-mean-square error, its initial value and the level at which it will be fixed at the end of the training cycle of the ANN model; $k_4$ is the parameter of the specific training rate, which depends on $n$, the volume of the training sample and the method used to minimize the deficiency (13); $t$ – dimensionless time of the training process, obtained by referring real time to the time of the first training cycle $t \in [t_1, t_2]$. Equation (20) reflects the phenomenology of the training process: at the initial moment of time, the deficiency is equal to $E_0$, and tends to $E^*$ at the end of the training cycle.

The proposed analytical models (18), (20) in practice make it possible to solve the following problems: to estimate the primary period of accumulation of information necessary for constructing the ANN model of the object under study; determine the initial structure of the ANN model based on the input characteristics of the object under research; estimate the number of training cycles of the ANN model and the time of its optimal retraining.

This approach was implemented by us to develop expert systems for medical purposes in [9].

7. Conclusion

The main tasks are formulated, the solution of which will make it possible to implement the technology of extracting knowledge about an object given by an array of empirical data using ANN models. Examples and basic equations are given that allow constructing an ANN model for the most important classes of objects: a static one, an object of the time series type and an object with a constant supply of new data. Further analysis of the obtained models will make it possible to extract new knowledge about objects, i.e. solve the main task of Data Mining. If the object has certain classes of nonlinearities listed at the beginning of the article, then the problem can be easily solved by adding functional neurons with appropriate transfer functions. The examples given above testify to the possibility of using these approaches.
References

[1] Kolmogorov A N 1956 Representation of continuous functions of several variables by superposition of continuous functions of fewer variables Dokl. Academy of Sciences of the USSR 108 pp 179–82 (in Russian)

[2] Kolmogorov A N 1957 On the representation of continuous functions of many variables by superposition of continuous functions of one variable Dokl. Academy of Sciences of the USSR 114 pp 953–56.

[3] Rabunal J R and Dorrado J 2006 Artificial Neural Networks in Real-Life Applications (Idea Group Publishing)

[4] Kamruzzaman S M and Jehad Sarkar A M 2011 A new data mining scheme using artificial neural networks Sensors 11 pp 4622–47

[5] Arockiaraj M C 2013 Applications of neural networks in data mining International Journal of Engineering and Science 3 pp 8–11

[6] Arzamastsev A A and Zenkova N A 2017 A software package for modeling neural networks with parametrically varying activation functions of neurons Certificate of registration of the computer program 2017619671 Russian Federation Applicant and copyright holder Federal State Budgetary Educational Institution of Higher Education "Tambov State University named after G.R. Derzhavin" 2017616527 (in Russian)

[7] Arzamastsev A A, Rikov V P and Kruchin O V 2012 Artificial neural network simulator with the implementation of modular training principle Certificate of registration of the computer program 2012618141 Russian Federation Applicant and copyright holder Federal State Budgetary Educational Institution of Higher Education "Tambov State University named after G.R. Derzhavin" 2012615778 (in Russian)

[8] Arzamastsev A A and Kozadaev A S 2006 Forecasting time series using artificial neural networks. Short-term forecast of air temperature Tambov State University Reports 11 pp 299–304 (in Russian)

[9] Arzamastsev A A, Zenkova N A and Neudahin A V 2009 Technology for constructing a medical expert system based on artificial neural networks Information Technology 8 pp 60–3 (in Russian)