

The problem of developing universal classifiers of biomedical data, in particular those that characterize the presence of a large number of parameters, inaccuracies and uncertainty, is urgent. Many studies are aimed at developing methods for analyzing these data, among them there are methods based on a neural network (NN) in the form of a multilayer perceptron (MP) using GA.

The question of the application of evolutionary algorithms (EA) for setting up and learning the neural network is considered.

Theories of neural networks, genetic algorithms (GA) and decision trees intersect and penetrate each other, new developed neural networks and their applications constantly appear.

An example of a problem that is solved using EA algorithms is considered. Its goal is to develop and research a classifier for the diagnosis of breast cancer, obtained by combining the capabilities of the multilayer perceptron using the genetic algorithm (GA) and the CART decision tree.

The possibility of improving the classifiers of biomedical data in the form of NN based on GA by applying the process of appropriate preparation of biomedical data using the CART decision tree has been established.

The obtained results of the study indicate that these classifiers show the highest efficiency on the set of learning and with the minimum reduction of Decision Trees; increasing the number of contractions usually degrades the simulation result. On two datasets on the test set, the simulation accuracy was $\approx 83-87\%$.

The experiments carried out have confirmed the effectiveness of the proposed method for the synthesis of neural networks and make it possible to recommend it for practical use in processing data sets for further diagnostics, prediction, or pattern recognition

Keywords: neural network, multilayer perceptron using a genetic algorithm, CART decision tree

UDC 004.8

DOI: 10.15587/1729-4061.2021.242795

DEVELOPMENT OF THE CLASSIFIER BASED ON A MULTILAYER PERCEPTRON USING GENETIC ALGORITHM AND CART DECISION TREE

Liudmyla Dobrovska

Corresponding author

PhD, Associate Professor*

E-mail: luci.dln17@gmail.com

Olena Nosovets

PhD*

*Department of Biomedical Cybernetics

National Technical University of Ukraine

«Igor Sikorsky Kyiv Polytechnic Institute»

Peremohy ave., 37, Kyiv, Ukraine, 03056

Received date 10.08.2021

Accepted date 13.10.2021

Published date 29.10.2021

How to Cite: Dobrovska, L., Nosovets, O. (2021). Development of the classifier based on a multilayer perceptron using genetic algorithm and cart decision tree. *Eastern-European Journal of Enterprise Technologies*, 5 (9 (113)), 82-90.

doi: <http://doi.org/10.15587/1729-4061.2021.242795>

1. Introduction

Researchers have recognized neural networks (NNs) as a tool for solving many problems related to biomedicine and the health care system. Among the areas of NN application in the healthcare system, the following can be distinguished: processing of biomedical signals, diagnostics of diseases, assistance to medical systems in supporting decision-making.

Neural networks “are able to study the relationship between input-output mapping on a given sample of data without any prior knowledge or assumptions about the statistical distribution of the data. This ability to learn from data without any prior knowledge makes the NN suitable for solving practical problems of classification and regression. In many biomedical applications, classification and regression problems form an important and integral part. In addition, NNs are inherently nonlinear, which makes them more practical for accurate modeling of complex data objects (or structures)” [1, 2].

Theories of NN, genetic algorithms (GA), fuzzy logic and decision trees intersect and penetrate each other, new developed NN and their applications are constantly appearing. The research of NNs that evolve makes the development of NNs more acceptable and better reflects the benefits of NNs in large-scale and complex networks.

The integration of GA and NN within a hybrid system designed to solve a specific problem implements a neuro-

evolutionary approach to teaching and tuning NN. It not only makes NNs capable of learning and development, but can solve some of the problems that exist in the design and implementation of NNs, increasing their productivity.

The development and implementation of NN based on GA have become an important area of research and development of the NN theory [3–5].

The study of the problems of integration of GA and NN showed that the relevant theories and methods need to be improved and standardized, and applied research needs to be strengthened.

The problem of developing universal classifiers of biomedical data in general and diagnostic data, in particular those that characterize the presence of a large number of parameters, inaccuracies and uncertainty, is urgent. Research is aimed at developing methods for analyzing these data, among them there are methods based on a network in the form of a multilayer perceptron (MP) using GA.

Using GAs to find the initial values of the NN weights before applying gradient-based methods is beneficial for supervised learning-based classification problems. Genetic algorithms are search procedures based on natural selection and inheritance mechanisms; they use the evolutionary principle of survival of the fittest individuals (chromosomes). Genetic algorithms differ from traditional optimization methods with such basic elements, in particular, GA:

1) not the values of the task parameters are processed, but their encoded form;

2) search for a solution proceeding not from one point, but from a certain population of points;

3) use only the target function (fitness function), and not its derivatives or other additional information;

4) apply probabilistic rather than deterministic selection rules. The listed properties, operations on populations, the use of a minimum of information about problems and the randomization of operations result in the stability of GAs and their advantages over other technologies.

The issue of applying evolutionary algorithms to NNs is not yet fully consolidated, although the first theoretical studies began in the early 90s. At the moment, it is necessary to constantly adapt the models to cope with unstable, variable and dynamic systems. This prompts the search for more promising systems.

It is urgent to solve the issues related to the problem of developing classifiers of biomedical data in the form of NNs based on GA, which characterize the presence of a large number of parameters, inaccuracies and uncertainty. The results of such studies will provide effective systems for their application in the health care system.

2. Literature review and problem statement

Building a NN based on a genetic algorithm increases the accuracy while reducing the time spent on solving problems. The results of one of the first studies in this direction are given in [6], where the results of the creation of such a network for the automation of turning in the manufacturing industry are presented. The proposed approach is universal and is used by researchers. Modern research developments of neural networks based on GA are carried out mainly in the following three aspects: optimization of weight coefficients, optimization of the architecture and rules of learning neural networks.

It is shown in [7] that a large number of algorithms have been developed to determine the values of the NN weight for a fixed topology, most of which usually fall into the local optimum. The results obtained are highly dependent on the learning parameters and initial weights, as well as the network topology. For example, in the case of a MP that learns using the backpropagation algorithm or its modifications, it is necessary to set the learning level, initial weight values, the number of hidden layers and neurons in each of them.

Optimizing compound weights turns into a global adaptive learning method. Learning with GA makes it possible to determine a set of weight coefficient values that approaches the global optimum without the need to calculate a gradient; the individual suitability of the “chromosome” can be defined as the error between the expected and actual output of the network. Usually, the effectiveness of hybrid learning is superior to learning methods that use only MP learning.

There are a large number of publications devoted to this area. For example, in [7], an approach to optimizing the network coefficients by studying the variations of operators is considered, and in [8], a new NN model with optimized initial values of the weight and thresholds that are related to each other is proposed. This paper also proposes other approaches to improve the accuracy and efficiency of forecasting.

The NN architecture includes various information (for example, the number of hidden layers, the number of neu-

rons, the activation functions of the layers, etc.). Researchers view NN as an architecture search problem, using learning fidelity and generalizability as an assessment standard to find the architecture with the best performance in the architectural space.

The main research on optimization of network architecture is carried out in two directions. The first of these is lazy reinforcement learning, which is that each component of the architecture is viewed as an action, and the sequence of such actions determines the overall structure of the network, the accuracy of which is used as a reward. This approach is considered in detail in [9, 10].

As shown in [9], convolutional neural networks and related hyperparameters usually vary depending on the task at hand. The development of architectures for such networks is a complex and time-consuming task, however, it does not guarantee an optimal network structure. In turn, the use of learning with deferred reinforcement makes it possible to automate the selection of neural network models, which significantly improves the quality of classification. One of the examples of the practical application of this method is given in [10]. The authors proposed four variants of GA used to learn a deep neural network model to solve the problem of finding a way out of the maze. Each of them was analyzed for the advantages or disadvantages of certain operators, but they all showed high efficiency in solving the task.

The second approach implements a search through successive mutations and repeated combinations of components, as a result of which the most effective architectures are selected, which are used to continue the evolutionary process. An example of this approach is the work [11, 12].

The authors of [11] presented a metamodeling algorithm that is used for the automatic generation of highly efficient neural network architectures. The learning agent learns to sequentially select the layers of the TM using Q-learning with a greedy exploration strategy and repetition experience. The agent explores a large but limited space of possible architectures and iteratively discovers performance-enhanced constructs in a learning task.

In [12], existing neuroevolution methods were used to create an automated CoDeepNEAT (Computing Deep NeuroEvolution of Augmenting Topologies) method to optimize deep learning architectures through evolution. The benefits of an evolutionary approach are illustrated by building a real-time image captioning application. The web is looking for architectures that are learning to integrate images and text to create signatures that blind users can access using existing screen readers.

Both approaches search in the space of discrete architecture. However, a direct search for the best architecture in discrete space is ineffective due to the exponential expansion of the search space with an increase in the number of architectures.

The number of alternative approaches is growing. So, in the article [13], the evolutionary strategy ELearNT (Evolutionary Learning of Rich Neural Network Topologies) is presented, which forms various topologies of neural networks. This type of topology is not fully understood in the literature. The results of the experiments show that in nonlinear regression problems the ELearNT algorithm can automatically build neural network topologies and are able to surpass the classical models of neural networks developed manually.

In [14], it is proposed to optimize the network architecture by reflecting variants of the NN structures in a continu-

ous vector space and carrying out optimization in this space by the gradient method.

Optimization of learning rules can be considered as a self-adjusting process of automatic search for new learning rules (self-adaptation of learning rules). Algorithm design is mainly dependent on the network architecture, so it is difficult to develop a learning rule when there is not enough prior knowledge of the network architecture. In addition, during learning a neural network and its learning rate largely depends on its learning rate.

The work [15] proposes an improvement of the basic approach to the regulation of NN with feedforward. Bayesian approach is attractive in that it provides automatic determination of regularization parameters. This paper shows that the refined Bayesian approach performs better than the classical NN regularization approach. However, the time spent on classification according to the proposed approach is much longer than for other networks.

GA can be used to develop the rules for learning the NN. The paper [16] proposes a method for choosing non-optimal learning rates by evolutionary adaptation of learning rates for each level at each step of learning.

The study of the problems of integration of GA and NN showed that the relevant theories and methods need to be improved and standardized, and applied research needs to be strengthened. For research enhancement methods, it is proposed to apply preliminary data analysis based on the classification and regression tree (CART). Therefore, it should be considered expedient to study a method for developing a neural network based on GA by combining the optimization of the NN weights and the application of preliminary data analysis based on the CART classification trees. This reduces the number of inputs to the neural network and improves the accuracy of the simulation. This allows such biomedical data classifiers to be improved by applying a process of appropriate data preparation to find the neural network architecture with the best performance in the architectural space.

3. The aim and objectives of research

The aim of research is to develop a classifier, obtained by combining the capabilities of the MP, learns with the help of GA. Consider the capabilities of this classifier of breast cancer data (preliminary analysis based on CART Classification Trees was applied to the data, which reduces the number of neural network inputs to improve modeling accuracy). This will reduce the number of NN parameters.

To achieve this aim, the following objectives were identified:

- develop a method for designing a power supply using GA and CART classification trees;
- check the effectiveness of the method on various databases for diagnosing breast cancer diseases;
- prove the possibility of using such NN in the health care system for diagnosing diseases and helping medical systems (or devices) to support decision-making.

4. Materials and methods of research

4. 1. Problem statement

Problem statement (classification of diagnostic data). Let a learning sample be given in the form of input-target

data pairs $\{x^1, t^1\}, \dots, \{x^Q, t^Q\}$, which is generated by the function $t^i=f(x^i), i=1, \dots, Q$, where $x^o=[x_1^i \dots x_n^i]^T$ – the vector input data with elements $t^i=f(x^i), i=1, \dots, Q$, where $x^o=[x_1^i \dots x_n^i]^T$ – desired feedback. The function $f(x)$ is considered unknown, but a set of its implementation is given $O=\{(\bar{\sigma}_1^o, \dots, \bar{\sigma}_n^o, t), i=1, 2, \dots, Q, n>1\}$. Build a network to determine the function $F(w, x^i)$, which approximates the function $f(x)$, describing the transformation of the input signal into the output, and satisfies the condition $\frac{1}{Q} \sum_{i=1}^Q |F(w, x^i) - t^i| < \epsilon$,

where ϵ – some positive number, which is called the residual.

In this case, the weight coefficients w of the network are determined on the basis of hybrid learning (based on GA and NN). Optimizing these values turns into a global adaptive learning method.

4. 2. Learning samples (review of real medical data

During the research, the issue of using evolutionary algorithms (EA) for setting up and learning neural networks is considered on the example of a specific task. Its goal is to develop and research a classifier for the diagnosis of breast cancer, obtained by combining the capabilities of a multilayer perceptron using GA and CART decision trees.

To accomplish this task, three sets of real medical data were used to conduct studies of patients with breast cancer (Table 1). All datasets were freely available from the Internet from medical schools (University of Wisconsin, Clinical Sciences Center). All datasets have a key attribute – a diagnosis – that will help supervise NN learning.

Medical data number “1” [12]. Medical dataset obtained from the Cancer Institute in Wisconsin via UCI.edu. This sample contains 140 observations for 35 attributes. Each record represents observation data for one breast cancer case. These are Dr. Wahlberg’s patients since 1984. Data include only cases of invasive breast cancer with no evidence of distant metastases at the time of diagnosis.

Medical data number “2”. Medical dataset obtained from the Cancer Institute in Wisconsin via Kaggle.com. This sample contains 569 observations for 32 attributes.

Medical data number “3”. Medical dataset obtained from the Cancer Institute in Wisconsin via Kaggle.com. This sample contains 683 observations for 11 attributes.

Table 1

The main parameters of the examined datasets

Data ID (No.)	Number of observations	Number of attributes	Classification attribute
1	140	35	Relapse/remission
2	569	32	Benign/malignant
3	683	11	Benign/malignant

These three datasets (Table 1) were analyzed and modified for use in this study.

4. 2. Algorithm for learning a multilayer perceptron using a genetic algorithm

Preliminary analysis based on CART Classification Trees is applied to the data, which reduces the number of inputs. Among the various classifier models, Decision Trees classifiers are known. Decision tree is a popular method of data analysis, the basis of which is learning by example and the formation of the corresponding hierarchical structure.

A decision tree splits the input space (known as the attribute space) of a dataset into mutually exclusive areas, each of which is assigned a name.

Based on the learning set, the CART decision tree was formed, which generates the function $c: \mathfrak{R}^n \rightarrow L$, where L – the set of class labels. For example, for a binary classifier, $L = \{0, 1\}$. If the vector $x = x^i = [x_1 \dots x_n]^T$, $i = 1, \dots, Q$ is fed to the input of this Tree, then its output $c(x)$ is equal to the value “1” if the class label is greater than or equal to the value “0.5” and “0” if the class label is less than “0.5”.

Particular attention in this work is paid to the formation of a learning algorithm for a multilayer perceptron, which is learned with the help of GA.

Typical steps of using HA for the evolution of a network in the form of a MP are as follows:

Step 1. Initialization (determination of fitness function, selection of GA parameters and initial population of chromosomes).

Let's use a certain coding strategy to encode the architecture and randomly generate the initial population of N individuals, each chromosome-individual represents an NN.

Step 2. Decode each individual of the current generation into an architecture and construct the corresponding NN.

Step 3. Assessment of the fitness of chromosomes in the population.

Let's learn each NN with a decoded architecture according to a predetermined learning rule. The learning takes place according to the MP learning algorithm. The network is being learned, during which it is used to solve a given problem. During the study, the performance of the investigated configuration of NN is measured – its fitness function (FF).

Let's calculate the FF for each individual (estimate all elements of the current population) in accordance with the above learning result.

Step 4. Select a few of the fittest individuals and reserve them for the next generation.

Step 5. Formation of a new population by using genetic operators.

Let's use genetic crossover and mutation operators to process the current population and create the next new one.

Step 6. Checking the condition for stopping the algorithm. If the condition is not met, repeat steps 2–5.

The number of generations depends on whether an acceptable solution has been reached (after a while, all chromosomes and associated FF values will become the same if there were no mutations). At this point, the algorithm should be stopped.

The learning process of the neural network has the form of a parallel search with the aim of improving the chromosome, which continues until the optimal network with the greatest fitness (the smallest value of the mean square error of the neural network) is found.

Most of the GA tracks population statistics in the form of average and minimum values of the population FF.

The fitness function, which is called the evaluation function, represents a measure of the fitness of a given individual (chromosome) in a population. It allows you to assess the degree of fitness of specific individuals in the population and select the most adapted from them according to the evolutionary principle of survival of the “strongest” (those that have adapted better). In optimization problems, the FF is usually optimized and called the objective function. At each GA iteration, the fitness of each individual of a given

population is assessed using the FF, and on this basis the next population of individuals is created, which form a set of potential solutions to the problem, for example, the optimization problem. The current population in GA is called a generation, and the term “new generation” (or “generation of descendants”) is applied to the newly populated individuals.

Coding schemes for information about the NN, presented in the chromosome.

The evolution of the weight of compounds usually takes place in two stages, during which it is necessary to determine:

- 1) an image of the weight for each individual chromosome (binary strings or ordinary numbers);
- 2) search genetic operators.

These decisions are very important because using different representations and genetic operators leads to different results.

The complete net is fed by concatenating all the weights in a single individual. The weights for one hidden unit are placed together so that the crossover operator swaps the complete hidden units, not just individual weights.

One of the important issues in the development of NN is the choice of genetic operators used in GA, since the accuracy depends on them. A detailed analysis of various approaches to the choice of genetic operators is given in [17]. The most common operators are mutation and crossover.

Mutation. The mutation operator has two main roles:

- 1) solution settings (a small change in the chromosome or movement in the search space, for example, adding a small random number after each epoch of learning the neural network);
- 2) big changes (macromutation operator, which randomly replaces a hidden neuron with another, initialization with random weight values).

Crossover operator. There are various variants of it, one of them works by randomly selecting node a in the first parental chromosome and node b in the second, the operator replaces node a with a copy of the second parental node.

There are various ways to encode information about the NN presented in the chromosome. The choice of presentation of information in genes determines the class of networks. The efficiency of the neuroevolutionary method in all parameters depends on the coding scheme.

There are two ways of encoding: direct and indirect.

In the case of direct coding, the chromosome represents some linear representation of the NN. In such a chromosome, the parameters of the network are clearly indicated: inputs, outputs and hidden neurons, connections between them, weight values of connections, and the like. Thanks to such a representation, it is always possible to build a one-to-one correspondence between the structural elements of the NN (neurons, connections, weight values, etc.) and the corresponding parts of the chromosome. This method of coding NN is the most intuitive and simple. Its main disadvantage is an increase in the size of the chromosome with an increase in the number of inputs, neurons, and NN connections. This leads to low efficiency at the expense of a significant increase in search space.

At each iterations of the cycle, information about the NN in the form of a chromosome is encoded, and the resulting network is tested, during which it is used to solve the problem. During testing, the performance of the studied NN configuration is measured – its FF. After all the elements of the current population have been estimated with the help of genetic operators, a new population is created. The learning

process of the neural network has the form of a parallel search with the aim of improving the chromosome, which continues until the optimal network with the greatest fitness (the smallest value of the mean square error of the neural network) is found.

The indirect approach uses more complex methods and algorithms for coding the parameters of the neural network. Usually, the genetic representation is more compact, thereby reducing the search space for the optimal network structure.

An example of a neural network for which it is necessary to select the optimal weight values (Tables 2, 3 – coding of weight and chromosome values) is shown in Fig. 1.

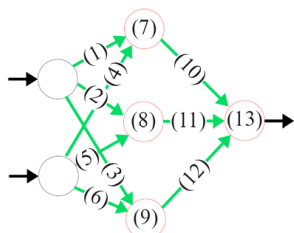


Fig. 1. Neural network – two-layer perceptron with two inputs

Table 2

Example of encoding weight values

Weight	Weight coefficient value
(1)	1 0 1=1×1=1
(2)	1 1 1=1×3=3
(3)	0 1 1=-1×3=-3
(4)	1 1 0=1×2=2
(5)	0 0 0=-1×0=0
(6)	1 0 1=1×1=1
(7)	1 0 0=1×0=0
(8)	0 1 1=-1×3=-3
(9)	0 1 0=-1×2=-2
(10)	0 0 1=-1×1=-1
(11)	1 1 0=1×2=2
(12)	1 0 1=1×1=1
(13)	1 1 0=1×2=2

One of the key issues in designing a multilayer network is determining the number of neurons needed to use it. If the number of neurons is too large, then the network will retrain (this means that the error on the learning data will be small, but on new data it can be large). A network that generalizes well will be successful on new data and on learning data.

The complexity of an NN is determined by the number of its free parameters (the number of weights and biases), which are determined by the number of neurons. If the network is too complex for a given dataset, then this is probably a case of overfitting and poor generalization.

The complexity of the network can be adjusted to match the complexity of the data. This can be done without changing the number of neurons. It is possible to adjust the effective number of free parameters without changing their actual number.

To get a network that is capable of generalization, it is necessary to learn NN. A network learned to generalize will perform the approximation also in new situations, as it does on the data on which it was learned. To get a good generalization, a key strategy is used, the essence of which is finding a simple model, clarifies the data.

Table 3

Chromosome coding

No.	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
Weight code	101	111	011	110	000	101	100	011	010	001	110	101	110

In terms of neural networks, a simple model is a network that contains the smallest number of free parameters (weight coefficients and landslides), or, equivalently, the smallest number of neurons. To get a network that generalizes well, you need to find the simplest network that matches the data.

There are at least five different approaches to creating simple networks: growth, pruning (pruning), global search, regularization, and early stopping.

There are various approaches to solving the problem of improving the ability of NN to generalize – they all try to determine the simplest network that will fit the data. These approaches cover two main groups:

- 1) limiting the number of weight coefficients (or, which is the same, the number of neurons) in the network;
- 2) limiting the value of the magnitude of the weight.

When designing an NN, the first approach was applied (limiting the number of neurons in the hidden layer).

It is assumed that there is a limited amount of data with which to teach the network. If the amount of data is not limited (which means: the number of data points is significantly more than the number of network parameters), then repeating will not be a problem of overfitting.

Generalization error score – multiple tests. To estimate this error for a specific NN, the following was taken into account. Given the limited data available, it is important to keep some test subset out of the way. After the network is learned, the error is calculated by learning the network on this test set. An error on the test case will indicate how the network will perform in the future; it is a measure of the network's generalizability.

To establish that a test is an indicator of a network's ability to generalize, it must be borne in mind that the test set:

- 1) should never be used for learning neural networks (or even select one network from a group of candidate networks); it should only be used after learning has been completed;
- 2) must be representative of all situations for which the network will be used.

This is sometimes difficult to guarantee, especially when the input space is multidimensional or complex in shape.

Let's assume that the test set is removed from the dataset before learning starts, and this set is used at the end of learning to assess generalizability.

NN architecture:

- 1) double layer perceptron,
- 2) the number of neurons:
 - in the hidden layer is equal to the number of inputs (parameters of the reduced database) divided in half;
 - in the original layer – one;
- 3) as the activation function in both layers, the sigmoid activation function $f(u) = \frac{1}{1 + e^{-u}}$, is used, the argument of which is any real number from the interval $(-\infty; +\infty)$, and its values refer to the interval $(0; 1)$.

Selection. A tournament selection method with an elite strategy was chosen: all chromosomes, except for the three most adapted ones, will be divided in pairs and compete to die (every second chromosome will die).

Reproduction (application of genetic operators). At the very beginning of the algorithm, the user can determine the chances of crossover and mutation separately: by default, these values are 100 % and 10 %, respectively.

A conventional single-point crossover was chosen as the operator for the crossover. When it is executed, two descendants appear (the first descendant receives the first part of the genes from the father, the other from the mother, and the second – the first part of the genes from the mother, the second from the father).

The standard single-point mutation is used as the mutation operator.

The number of generations depends on whether an acceptable solution is reached or the specified number of iterations is exceeded. The algorithm tracks the statistics of the population in the form of average and minimum FF values of the elements of the population.

4. 3. Selection of essential features for a neural network based on the CART tree

To determine the CART Decision Trees, the MatLab system was used (the American company MathWorks, which specializes in the development of software for mathematical calculations and simulation; USA). The list of signs after selection for medical data number “1” is given in Table 4.

Table 4

List of signs after screening on medical data 1

No.	There were 35 attributes – now 13
1	$x [0]$
2	$x [4]$
3	$x [1]$
4	$x [9]$
5	$x [10]$
6	$x [6]$
7	$x [5]$
8	$x [7]$
9	$x [12]$
10	$x [11]$
11	$x [2]$
12	$x [8]$
13	$x [3]$

The same lists of signs for other medical data are formed in a similar way (Table 5).

Table 5

List of signs after selection on medical data 2 and 3

No	There were 32 attributes – now 10	There were 11 attributes – now 9
1	$x [7]$	$x [1]$
2	$x [1]$	$x [5]$
3	$x [3]$	$x [0]$
4	$x [4]$	$x [2]$
5	$x [2]$	$x [3]$
6	$x [6]$	$x [7]$
7	$x [5]$	$x [6]$
8	$x [8]$	$x [4]$
9	$x [9]$	$x [8]$
10	$x [0]$	

Based on the CART Decision Trees, a reduction in the number of attributes is performed

5. Results of the study of the network design method based on the genetic algorithm, CART classification trees

5. 1. Method of designing a multilayer perceptron

A MP design method using GA and CART classification trees has been developed.

The complete net is fed by concatenating all the weights in a single individual. The genetic operators (mutation and crossover) used in GA are selected, since the accuracy depends on them. The weights for one hidden unit are placed together so that the crossover operator swaps the complete hidden units, not just individual weights. This method uses a combination of GA-based NN weighting optimization and preliminary data analysis based on CART classification trees, which reduces the number of NN inputs to improve modeling accuracy.

There is the following result of data analysis based on the CART classification trees:

- on the set of medical data number “1” there were 35 attributes after the reduction became 13;
- on the set number “2” there were 32 attributes – it became 10;
- on the set number “3” there were 11 attributes – it became 9.

5. 2. Checking the effectiveness of the method

The NN was constructed twice. On the initial data without reducing the number of incoming attributes (Table 6) and on the data after reducing the number using the CART Classification Trees (Table 7).

To improve the performance of a binary classifier, certain sets S_1 and S_2 (S_1 is a learning set, S_2 is a testing set or control sample).

Table 6

The result of modeling based on the MP-network using the original databases

Data-base No.	Number of attributes	Number of examples (numbers)	Modeling results	Accuracy (%)	
1	35	Learning $S_1=1:70$	22	11	60.21
			10	27	
		Testing $S_2=71:140$	21	14	64.22
			8	27	
2	32	Learning $S_1=1:284$	118	22	75.86
			31	113	
		Testing $S_2=285:569$	117	23	73.36
			25	119	
3	11	Learning $S_1=1:341$	148	11	80.33
			27	155	
		Testing $S_2=342:683$	138	20	78.21
			23	160	

As it is possible to see, the classification accuracy ranged from 60.21 % to 80.33 % in the learning set and from 64.22 % to 78.21 % in the test set.

The result of the functioning (modeling) of a binary classifier on reduced databases is shown in Table 7.

Accuracy shows how many correct results were obtained using a given method (Table 8).

Table 7

The result of modeling based on a MP-network using three reduced databases

Data-base No.	Number of attributes (were – are)	Number of exam-ples (numbers)	Modeling results		Accura-cy (%)
1	35–13	Learning $S_1=1:70$	22	11	70.0
			10	27	
		Testing $S_2=71:140$	21	14	68.57
			8	27	
2	32–10	Learning $S_1=1:284$	118	22	81.34
			31	113	
		Testing $S_2=285:569$	117	23	83.1
			25	119	
3	11–9	Learning $S_1=1:341$	148	11	88.86
			27	155	
		Testing $S_2=342:683$	138	20	87.39
			23	160	

Table 8

Interpretation of experiment results

Class	Modeling result based on RBF network		Accuracy (%)
	Class 1 (True/False)	Class 2 (True/False)	
Class 1=«1»	22 (true «1»)	11 (false «0»)	((22+27)/70)·100 % = 70,0 %
Class 2=«0» or «-1»	10 (false «1»)	27 (true «0»)	

The research results show that the number of attributes and their reductions, which lead to compact networks, affects the simulation result (Table 7):

- small networks generalize well,
- large networks tend to get poor generalizations.

On medical dataset 3 on the test set, the modeling accuracy was ≈87 %.

The efficiency of the algorithms for assessing the current state of objects is one of the main characteristics of computer systems that provide a solution to the problems of medical diagnostics. A convenient means of assessing the effectiveness of a diagnostic algorithm is a method based on the analysis of the so-called Receiver Operating Characteristic curve (ROC). Traditional ROC analysis involves comparing the operational characteristics of an algorithm – sensitivity (S_E) and specificity (S_P). Having these characteristics, it is possible to depict the results of checking the algorithm in a two-dimensional ROC-space, in which S_E values are plotted along the ordinate, and $1-S_P$ is plotted along the abscissa. Thus, a diagnostic test (binary classifier) with fixed operational characteristics is displayed as a point in the ROC space.

Sometimes it is difficult to say which of the classifiers the best is. Therefore, a new indicator of the quality of classifiers was introduced – AUC (Area under curve), which describes the area under the ROC curve on the ROC analysis graph. For a ROC curve that goes through the upper left corner, the AUC will be “1”. If the curve is as close as possible to $f(x)=x$, then the AUC is 0.5. For the data (Fig. 2–4). Graphical dependencies obtained using the ROC package in the R-Studio software tool.

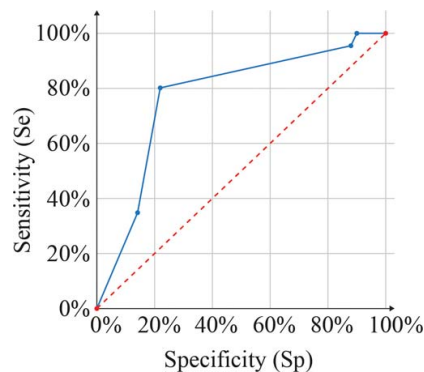


Fig. 2. ROC analysis of medical data from the first set (AUC≈0.69)

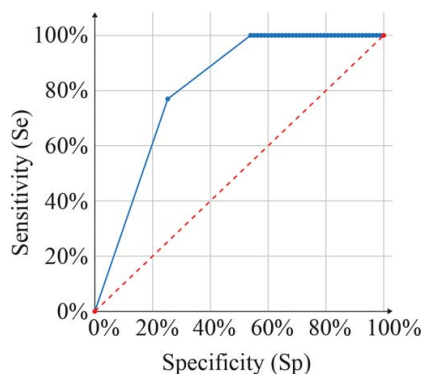


Fig. 3. ROC analysis of medical data from the second set (AUC≈0.82)

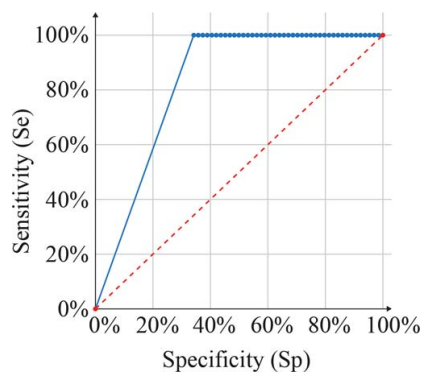


Fig. 4. ROC analysis of medical data from the second set (AUC≈0.88)

So, the possibility of improving the classifiers of biomedical data in the form of NNs based on GA has been established by applying the process of appropriate preparation of biomedical data using the CART Decision Trees.

The effectiveness of the method was tested on three databases for diagnosing breast cancer diseases. There are the following research results:

- on the set of medical data number “1” there were 35 attributes after the reduction became 13; on the set of testing, the modeling accuracy was ≈69 %;
- on the set number “2” there were 32 attributes – it became 10, on the set of testing the modeling accuracy was ≈83 %;
- on the set number “3” there were 11 attributes – it became 9 on the set of testing, the modeling accuracy was ≈87 %.

The sensitivity and specificity of this classifier for various datasets was assessed: the quality of its work on biomedical datasets showed the following result:

- on the set number “1” $AUC \approx 0.69$;
- on set number “2” $AUC \approx 0.82$;
- on set number “3” $AUC \approx 0.88$.

So, classifier networks of small size (with a small number of inputs and, accordingly, a small number of weight values) generalize well.

The obtained results of the study indicate that these classifiers show the highest efficiency on the set of testing and with the minimum reduction in Decision Trees; increasing the number of contractions usually degrades the simulation result.

5. 3. Possibility of using such a multilayer perceptron in the healthcare system

On sets number “2” and “3” on the set of testing, the modeling accuracy was $\approx 83\%$ and $\approx 87\%$, respectively. The obtained results of the study prove the possibility of using such NN in the healthcare system for diagnosing diseases and helping medical systems (or devices) to support decision-making.

6. Discussion of the research results of the network design method based on the genetic algorithm CART classification trees

The problem of developing universal classifiers of biomedical data, in particular those that characterize the presence of a large number of parameters, inaccuracies and uncertainty, is complex and urgent. This research problem has limitations in the form of modeling (classification) accuracy of the corresponding classifiers.

Modern research developments of NN based on GA are mainly carried out in the following three areas: optimization of weight coefficients, optimization of architecture and rules for learning NN.

A method for designing a power supply using GA is proposed by combining the optimization of network weights based on GA and the application of preliminary data analysis based on the CART classification trees. This reduces the number of neural network inputs to improve simulation accuracy.

One of the key issues in designing a multilayer network is determining the number of neurons needed to use it. If the number of neurons is too large, then the network will retrain (this means that the error on the learning data will be small, but on new data it can be large). A network that generalizes well will be successful on new data and on learning data.

To confirm this hypothesis, a study was conducted on three sets of medical data, containing the results of clinical analyzes in patients with breast cancer. The number of initial attributes in these sets ranged from 11 to 35.

The initial data sets were used to construct NN (Table 6):

- the learning accuracy of the system with 35 inputs and 6 neurons is 60.21 %; testing – 64.22 % (first data set);
- the learning accuracy of the system with 32 inputs and 5 neurons is 75.86 %; testing – 73.36 % (second set of data);
- the learning accuracy of the system with 11 inputs and 4 neurons is 80.33 %; testing – 78.21 % (third dataset).

By defining the CART Decision Trees, the list of attributes was reduced in the first dataset from 35 to 13 (Ta-

ble 4), in the second dataset – from 32 to 10 (Table 5), in the third – from 11 to 9 (Table 5). The resulting reduced datasets were divided into learning and test sets. The learning samples were used to learn the neural network, which was used as a binary classifier, while the test samples did not participate in this. The performance of this network was evaluated using learning and test sets.

As a result of counting truly positive and truly negative cases, it was determined (Table 7) that:

- the learning accuracy of the system with 13 inputs and 6 neurons is 70.0 %; testing – 68.57 % (first data set);
- the learning accuracy of the system with 10 inputs and 5 neurons is 81.34 %; testing – 83.1 % (second set of data);
- the learning accuracy of the system with 9 inputs and 4 neurons is 88.86 %; testing – 87.39 % (third dataset).

Additionally, the quality of the classifier data was assessed using the AUC (Fig. 2). ROC analysis of the first set of medical data yielded $AUC \approx 0.69$, the second – $AUC \approx 0.82$; the third – $AUC \approx 0.88$.

To obtain a good generalization, a strategy was used, the essence of which is to find the simplest model that explains the data. In terms of neural networks, a simple model is a network that contains the smallest number of free parameters, or, equivalently, the smallest number of neurons.

The result of the study shows that networks are more compact, which use fewer attributes, allowing for greater accuracy. In addition, the complexity of NN is determined by the number of its free parameters (the number of weights and biases), which are determined by the number of neurons. After reducing the input attributes, the accuracy on the test set for the first dataset increased from 64.22 % to 68.57 %, for the second – from 73.36 % to 83.1 %, for the third – from 78.21 % to 87.39 %.

The results obtained can be explained by the fact that reducing the dimension of the input data allows to eliminate the problem of multicollinear data and reduce the amount of noise by reducing ineffective attributes. It is also worth noting that simplifying the structure of the neural network allows you to avoid overfitting the classifier when the accuracy on the learning set significantly exceeds the accuracy on the test set.

The advantage of this study in comparison with the known analogs is the simplification of the neural network model (in particular, a decrease in the number of inputs) due to the use of CART wooden classifications.

However, the assessment of the impact on the quality of modeling of changes in the network topology, namely, the number of neurons in the hidden layer, is not fully investigated. In addition, it should be mentioned that CART Classification Trees are sensitive to noise in the input data, so an important step is their Preprocessing: removal of outliers and rationing.

The development of this study may be to define optimized initial values for weights and thresholds that are related; exploring other operator variations.

7. Conclusions

1. A method for designing a power supply using GA has been developed by combining the optimization of the weight coefficients of NN based on GA and the application of preliminary data analysis based on the CART classification trees. This reduces the number of inputs to the neural

network and improves the accuracy of the simulation. On its basis, a software application was implemented in the form of a classifier for diagnosing breast cancer diseases, obtained by combining the capabilities of a multilayer perceptron using a genetic algorithm and CART decision trees.got:

- on the set of medical data number “1” there were 35 attributes after the reduction became 13;
- on the set number “2” there were 32 attributes – it became 10;
- on the set number “3” there were 11 attributes – it became 9.

2. Testing the effectiveness of the method on various databases for diagnosing breast cancer diseases showed the following result: small classifier networks (with a small number of inputs and, accordingly, a small number of weight values) generalize well. The obtained results of the study indicate that these classifiers show the highest efficiency on the set of

testing and with the minimum reduction in Decision Trees; increasing the number of contractions usually degrades the simulation result.

There are such indicators of the results of the study of the classifier for various datasets: on the set of testing, the modeling accuracy:

- $\approx 69\%$ in medical data set number “1”;
- $\approx 83\%$ in the set number “2”;
- $\approx 87\%$ in the set number “3”.

The quality of the classifier’s work on different sets of biomedical data showed the following result:

- on the data set number “1” $AUC \approx 0.69$;
- on data set number “2” $AUC \approx 0.82$;
- on data set number “3” $AUC \approx 0.88$.

3. On a variety of tests, the simulation accuracy showed an acceptable result and amounted to:

- $\approx 83\%$ in the set number “2”;
- $\approx 87\%$ in the set number “3”.

References

1. Dobrovska, L., Dobrovska, I. (2015). *Teoriia ta praktyka neironnykh merezh*. Kyiv: NTUU «KPI», 395.
2. Dobrovska, L., Dobrovska, I. (2017). Design of the universal classifier as a RBF network based on the CART solution tree. *Eastern-European Journal of Enterprise Technologies*, 4 (4 (88)), 63–71. doi: <http://doi.org/10.15587/1729-4061.2017.108976>
3. Farizawani, A., Puteh, M., Marina, Y., Rivaie, A. (2020). A review of artificial neural network learning rule based on multiple variant of conjugate gradient approaches. *Journal of Physics: Conference Series*, 1529, 022040. doi: <http://doi.org/10.1088/1742-6596/1529/2/022040>
4. Yao, X. (1993). A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems*, 8 (4), 539–567. doi: <http://doi.org/10.1002/int.4550080406>
5. Dutta, P., Kumar, A. (2018). Modeling and Optimization of a Liquid Flow Process using an Artificial Neural Network-Based Flower Pollination Algorithm. *Journal of Intelligent Systems*, 29 (1), 787–798. doi: <http://doi.org/10.1515/jisys-2018-0206>
6. Venkatesan, D., Kannan, K., Saravanan, R. (2008). A genetic algorithm-based artificial neural network model for the optimization of machining processes. *Neural Computing and Applications*, 18 (2), 135–140. doi: <http://doi.org/10.1007/s00521-007-0166-y>
7. Castillo, P., Arenas, M., Castillo-Valdivieso, J., Merelo, J., Prieto, A., Romero, G. (2003). Artificial Neural Networks Design using Evolutionary Algorithms. *Advances In Soft Computing*. London: Springer, 43–52. doi: http://doi.org/10.1007/978-1-4471-3744-3_5
8. Ding, S., Xu, L., Su, C., Zhu, H. (2010). Using Genetic Algorithms to Optimize Artificial Neural Networks. *Journal Of Convergence Information Technology*, 5 (8), 54–62. doi: <http://doi.org/10.4156/jcit.vol5.issue8.6>
9. Jaafra, Y., Luc Laurent, J., Deruyver, A., Saber Naceur, M. (2019). Reinforcement learning for neural architecture search: A review. *Image and Vision Computing*, 89, 57–66. doi: <http://doi.org/10.1016/j.imavis.2019.06.005>
10. Baker, B., Gupta, O., Naik, N., Raskar, R. (2021). Designing neural network architectures using reinforcement learning. *Int. Conf. Learn. Represent.* San Juan. Available at: <https://arxiv.org/abs/1611.02167>
11. Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O. et al. (2019). Evolving Deep Neural Networks. *Artificial Intelligence In The Age Of Neural Networks And Brain Computing*. Elsevier, 293–312. doi: <http://doi.org/10.1016/b978-0-12-815480-9.00015-3>
12. Real, E., Aggarwal, A., Huang, Y., Le, Q. V. (2019). Regularized Evolution for Image Classifier Architecture Search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 4780–4789. doi: <http://doi.org/10.1609/aaai.v33i01.33014780>
13. Matteucci, M. (2006). ELearnNT: Evolutionary Learning of Rich Neural Network Topologies. *Computer Science*. doi: <http://doi.org/10.21236/ada456062>
14. Luo, R., Tian, F., Qin, T., Chen, E., Liu, T. (2018). Neural Architecture Optimization. *Conference And Workshop On Neural Information Processing Systems*, 7827–7838.
15. Sariev, E., Germano, G. (2019). Bayesian regularized artificial neural networks for the estimation of the probability of default. *Quantitative Finance*, 20 (2), 311–328. doi: <http://doi.org/10.1080/14697688.2019.1633014>
16. Kim, H. B., Jung, S. H., Kim, T. G., Park, K. H. (1996). Fast learning method for back-propagation neural network by evolutionary adaptation of learning rates. *Neurocomputing*, 11 (1), 101–106. doi: [http://doi.org/10.1016/0925-2312\(96\)00009-4](http://doi.org/10.1016/0925-2312(96)00009-4)
17. Michel, D., Navarro, D. (2021). Genetic Operators and Their Impact on the Training of Deep Neural Networks. *Metaheuristics In Machine Learning: Theory And Applications*. Cham: Springer, 97–124. doi: http://doi.org/10.1007/978-3-030-70542-8_5
18. UCI Machine Learning Repository: Data Sets. Available at: <http://archive.ics.uci.edu/ml/datasets.php> Last accessed: 22.09.2021