ATLANTIS PRESS

Atlantis Highlights in Computer Sciences, volume 2

Fourth Workshop on Computer Modelling in Decision Making (CMDM 2019)

Neuroevolution Forecasting of the Living Standards of the Population

Liudmila P. Bilgaeva*, Erzhena Ts. Sadykova†, Victor A. Filippov*‡

*‡East Siberia State University of Technology and Management, Russian Federation

*bilgaeval@mail.ru

†Baikal Institute of rational nature management SB RAS, Russian Federation

***Abstract*—The article is devoted to the problem of forecasting indicators of living standards using neuroevolutionary approach. The NEAT method is considered that allows generating the neural network topology automatically. The paper proposes modification of this method and its implementation based on the activation functions mutation, which enabled to expand the set of possible solutions. A comparative analysis of the training, testing, and forecasting accuracy is carried out. Target indicator forecasting was performed with preliminary forecasting of factor signs that increased forecasting accuracy in comparison to the Windows method used to forecast target indicators directly.**

***Index Terms*—data mining, neuroevolution, genetic algorithm, mutation of activation functions, window method, feature selection method, visualization of neural network topology, living standards of the population**

I. Introduction

At the present stage of society development the state economy is primarily characterized by the level and quality of life of the population, their improvement being an important strategic task. Russian regions are characterized by high differentiation of living standards, increasing social division of society and property stratification. The living standard gap depends on various socio-economic factors, regional and geographical features of the territories influencing directly on the life quality of the population [3]. Unemployment level, per capita income and the Gini coefficient are among the main indicators of the life quality of the population [16], [17]. The article attempts to build a forecast of these indicators based on factor assessment of socio-economic and environmental parameters on the example of the Republic of Buryatia, with the neuroevolution forecasting method being used. The forecasting task itself is not new. Methods of regression analysis were traditionally used to solve it [2].The rapid development of computer and information technology gave impetus to the creation and development of new methods relating to the technology of data mining and machine training [1], [8]. Since the manual construction of neural network topologies requires a lot of knowledge in the field of artificial intelligence, mathematical calculations, and also takes a lot of time to rebuild a new model, if the experiment was unsuccessful, we suggest using the method of neural network with automatic topology generating to solve the problem of forecasting. The neuroevolution method – Neuroevolution through Augmenting Topologies (NEAT) is considered in this regard. It belongs to the so-called methods of topology and weights evolution (Topology and Weight Evolving Artificial Neural Network, TWEANN) [14]. The NEAT method provides three types of mutations:

- mutation of node addition;
- mutation of adding link;
- mutation of the connection weights.

In this paper we introduce the concept of activation functions mutation, which allows us expanding the space of possible solutions. It is proposed to forecast the target indicator based on preliminary forecasting of factor features to improve the forecasting accuracy. Visualization of the obtained neural network topologies enables not only to demonstrate mutations of activation functions during the topology evolution, but also to evaluate the factor attributes having the greatest impact on the target indicators.

II. Theoretical Foundations of Neuroevolution

Neuroevolution is a process of artificial evolution of neural networks, where evolutionary computational algorithms (EC) are used [15]. Evolutionary algorithms deal with a variety of genotypes, where each genotype is a neural network. An artificial neural network is a mathematical model represented by a sequence of processors (neurons) connected with each other by synapses through which signals pass. Neurons in a given network are processors that perform a number of simple mathematic signal operations. Signals are information flowing from one neuron to another through connections (synapses). The paper uses the most well-known type of a neural network – a fully connected multilayer perceptron [10]. It consists of a number of layers with neurons where each neuron of the current layer is connected to each neuron of the next one, and receives input values from each previous neural layer. One of the features of a neural network is that it can be trained. In the training process, the neural network is able to identify the relationship between input and output data by calculating the coefficients of connections between neurons. There are the following paradigms of neural networks training: with a teacher, without a teacher, and with reinforcement. Training with a teacher is used in this work. It is based on the comparison of the real and desired output of the neural network. There is a set of pairs $(x, y), x \in X, y \in Y$, and the goal is to find the function $f : X \to Y$ in an admissible class of functions, which corresponds to the available examples. The architecture of the neural network can be described as

a directed graph, where each $i-th$ node performs the f_i function transferring in the following way:

$$y_i = f_i * (\sum_{j=1}^{n} w_{ij} x_j + \theta_i), \tag{1}$$

where y_i is an output value of the node i, x_j is $j-th$ node input, $w_{i,j}$ – weight of the link between i and j nodes, θ_i – node bias (threshold), f_i – activation function. The most commonly used activation functions are sigmoid, hyperbolic tangent, ReLU. Sigmoid is a smooth monotonic nonlinear S-shaped function, having $(0, 1)$ definition domain and the following form:

$$f(x) = \frac{1}{1 + e^{-ax}}, \tag{2}$$

where α – function slope. Hyperbolic tangent is a smooth monotonic nonlinear S-shaped function. In contrast to the sigmoid function, it has $(-1; 1)$ definition domain and is described by the formula:

$$f(x) = \frac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}}, \tag{3}$$

ReLU is a linear function with $[0, \infty)$ definition domain and the following representation:

$$f(x) = max(0; x), \tag{4}$$

where $max(0, x)$ is a function returning its maximum value.

III. Validation and Choice of Neuroevolution Method

Neuroevolutionary algorithms are divided into two large groups:

- Weight Evolving Artificial Neural Networks.
- Topology and Weight Evolving Artificial Neural Networks.

This paper examines the second group algorithms. These include a large number of methods. The effectiveness of these methods and the classes of networks that can be built with their help are determined by the method of encoding and decoding their genotypes and phenotypes. Most researchers have identified two areas of encoding: direct encoding [9], [14] and indirect encoding [5]–[7]. Table I represents these methods and their authors.

TABLE I
The Authors of the Neuroevolution Methods

Methods	
Direct Encoding	Indirect Encoding
Miller, Todd, Hedge Stanley, Miikkulainen	Kitano Nolfi, Parisi Cangelosi, Nolfi, Parisi Cangelosi, Elman O'Neil, Brabazon Moriarty, Miikkulainen Gomez, Miikkulainen

In a direct coding scheme the genotype is equal to the phenotype, that is, the neurons and connections are directly indicated in the genotype. When using an indirect coding scheme, a genotype contains only rules and structures for creating a neural network. Comparative analysis of neuroevolutionary methods is a complex process, as there are no universal criteria for their evaluation. A list of criteria inherent to encoding methods is considered in this regard. Table II gives a comparative analysis of the methods according to these indicators.

As a result of the analysis, the method of direct genes encoding was chosen due to easy interpretation of the resulting network structure and high performance.

IV. NEAT Method and its Modification

A. Classical NEAT

The research work employs the NEAT (Neuroevolution through Augmenting Topologies) direct encoding method proposed by Stanley and Miikkulainen. It refers to the methods of topology and weights evolution (Topology and Weight Evolving Artificial Neural Network, TWEANN) [14]. The NEAT method starts with the minimal neural network topology when all inputs are directly connected to the outputs. Then it makes an evolution of the weights and gradually complicates the topology creating solutions closest to the minimal size. Minimality gives a performance advantage as compared to other neuroevolutionary methods. This method successfully overcomes the disadvantages of direct coding methods while retaining all their main advantages:

- Competing Conventions. The same phenotype of an artificial neural network can be represented in the genotype in different ways.
- Unprotected innovations. In the process of neuroevolution, innovations (changes in the neural network structure) that appear when neurons are added or removed reduce the overall fitness of the neural network, that is, the value of the fitness function decreases until the weight of the added neuron is optimized. This is due to the fact that adding or removing a neuron or connection introduces a new non-linearity into the linear process.
- Initial size and topological innovation. In almost all neuroevolutionary methods the generation of the initial population is a neural network with random topology. This results in a loss of time to eliminate non-viable topologies, as well as premature convergence to solutions due to non-optimal topologies, i.e. the network is too large. The author of the NEAT method solved this problem by using minimal initial topology, historical markings, and speciation. Thus interbreeding occurs within species, which include only homologous genotypes, and individuals with a similar structure are eliminated leaving the possibility for the development of other individuals, or fall into other species bringing genetic diversity.

TABLE II
COMPARATIVE ANALYSIS OF THE ENCODING METHODS

Indicators	Methods	
	Direct Encoding	Indirect Encoding
Ease of implementation	+	-
Modularity	+	-
Scalability	-	+
Small search space	-	+
Reflection the search space on the subject area	+	-
Simple choice of genetic operators	+	-
The ability to determine the causes of key changes	+	-

In addition to high search efficiency, the NEAT method has a high calculating speed due to the minimum network topology.

B. Modified NEAT

In the classical NEAT algorithm mutation of weights is possible. This paper proposes to perform a mutation of activation functions, which is the modernization subject. It includes a number of possible neuron gene mutations:

- Change one activation function to another.
- Shift the function slope coefficient to a random number.
- Set a new value for the slope factor.

The introduction of these possibilities requires the following information to be added to each neuron gene:

- Type of activation function (sigmoid, linear, tangent).
- Function slope coefficient.

It is proposed to apply formulas 5 and 6 not only to the link genes as in the classical NEAT method, but also to neuron genes, when classifying an individual to any species, which will insignificantly increase the complexity of the calculations.

$$\delta = c_1 \frac{E}{N} + c_2 \frac{D}{N} + c_3 \bar{W}, \tag{5}$$

where δ is the distance compatibility of various structures; E – the number of excess genes; D – the number of non-disjoint genes; $\bar{W}$ – the average weight of differences of compatible genes. c_1, c_2, c_3 coefficients can be used to adjust the values of these three factors. N is the amount of genes in the largest genome.

$$f_i{}' = \frac{f'}{E_{j=1}^{n} sh(\delta(i,j))}, \tag{6}$$

where $f_i{}'$ – the fitness for an individual i is calculated in accordance with its distance δ from any other genome j inside the population; where the separating function $sh()$ is set to 0, if the distance $sh(\delta(i,j)$ is above the threshold δ_t, otherwise, $sh(\delta(i,j)$ is set to 1.

In order to assign a new activation function to a neuron, it is necessary to create a container that will contain different types of activation functions and if necessary, extract a random function from it and assign it to the neuron. It should be noted that the mutation of the activation function, according to the NEAT rules, assigns a new innovation number to the neuron. However, it may happen that the container returns the same activation function that the neuron has. In this case, the neuron has not changed, but the innovation has occurred. Such a situation can lead to gene crossing errors and further to the "loss of course" of the evolutionary method, therefore, a solution can never be found. Accordingly, if a mutation of the activation function occurs, it is necessary to ensure that a different activation function is assigned to the neuron.

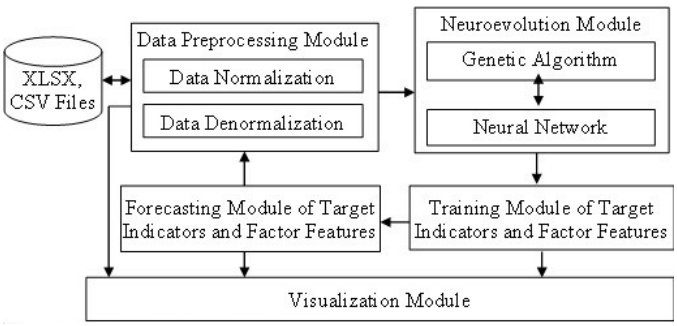

Fig. 1. Software Architecture

V. NEUROEVOLUTIONARY FORECASTING SOFTWARE

The developed program is a desktop application with a graphical user interface, the architecture of which is shown in Fig. 1.

The program includes five modules: a data preprocessing module, a target values and factor signs training module, a neuroevolution module, a forecasting module for target indicators and factor signs, and a visualization module. To start working with the program you need to load the initial data samples in the XLSX format (MS Excel format) and CSV (text format, separated by the symbol ";"). The data preprocessing module is designed to normalize the source data and denormalize them. This is due to the fact that normalized sets of initial data are fed to the input of the neural network, and the final results should be presented in absolute values of the target indicators. Rationing is a linear transformation by the formula 7.

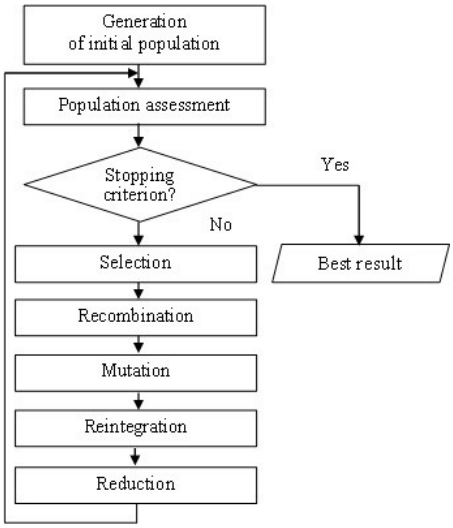

Fig. 2. Genetic Algorithm Scheme

$$\tilde{x_i} = \frac{x_i - min}{max - min}(b - a) + a, \tag{7}$$

where x_i – value to be normalized; min – minimum value in the set X; max – maximum value in the set X; a, b – interval to which the x_i value will be reduced.

The neuroevolution module is the core of the program and consists of two components that together implement the NEAT method: a neural network, and a genetic algorithm. As a modification of the classical NEAT method was proposed, the NEAT4J library taken as its basis was rewritten by 50–70%. It enabled to solve such tasks as to optimize a code, change the logic of data exchange between class objects, and introduce the possibility of neuron mutation. A neural network receives a vector of values with dimension equal to the number of neurons in the input layer, passes it through the neural network, and writes the result into a new vector. The genetic algorithm is designed to build and train neural networks [4]. The scheme of the genetic algorithm is shown in Fig. 2.

The neural networks selected by the genetic algorithm are further used in other modules as predictive elements. For the genetic algorithm to work, it is necessary to give a number of settings after which the main evolution cycle consisting of many epochs begins. Every epoch triggers the existing genotypes evolution, and then checks the interruption conditions. At the same time additional data processing is performed in this cycle: the completion percentage is calculated, the best genotype for iteration is saved in a file. The genotype evolution in the current epoch is implemented by a special *runEpoch()* method which uses the principles of multithreaded programming to provide parallel computing. The method divides all genotypes available in the current epoch into five equal parts. Next, the algorithm considers the fitness function value for each genotype in parallel. Two genotypes are necessary to complete crossing and create a descendant. Therefore, the use of different genetic operators requires that calculations must be completed in each stream. The parallelization process is possible due to the fact that the calculation of the fitness function value for each genotype is an independent operation. The fitness function converts an available genotype into a neural network, passes the training data set through the received network and calculates the mean square error of the training sample. A test sample is then taken and passed through the neural network in the same way; the root mean square error for the testing sample is calculated at the same time. The value of the fitness function of the genotype is formed as the sum of errors in the training and testing samples. The sum of errors allows determining, if the genotype is fit to all parts of the data available. If only one of the errors is used as the fitness value, it will lead to a strong discrepancy in the values in the area where the error was not taken into account. After calculating the fitness value for each individual, the selection process begins. Depending on the user-defined parameters, the genotypes are divided into subgroups (niching is applied); selection, crossover, mutation, reintegration, and reduction take place in these subgroups. After these operations, a new epoch is launched using newly formed individuals.

The training module consists of two parts. The first part is focused on the training of target indicators using a genetic algorithm, which is performed in two stages: preparation for training and training itself. At the stage of preparation for training, normalized initial data are loaded, it is divided into training and testing samples, and the parameters of the genetic algorithm are adjusted. Important settings include the probability of occurrence of the genetic algorithm events (the probability of a new link, the probability to change the activation function, etc.), the choice of activation functions, the size of populations and the number of epochs. These parameters affect the duration of training, as well as the load of RAM. The second part is focused on the factor features training by the window method. Using the *WindowPrediction* class methods a new set for a factor attribute is generated. This method employs an algorithm of splitting data into windows. 75% of the obtained set is used for training, the rest amount – for testing. The window method training is designed to obtain a neural network prediction of factor signs. For each factor sign, a neuroevolutionary trainer is created, which is a neuroevolutionary module described above. At the input of the method that starts the learning process of the factor sign, parameters such as the serial number of the column with the factor sign in the predicted sample and settings for the neuroevolutionary module are received. Since all factor signs are independent of each other, and it is necessary to have data on all factor signs involved in the forecasting process for further forecasting of the target indicator, the program uses parallel computing. Therefore, the *train()* method returns a link to the stream on which parallel calculations will be performed. To speed up the training process, we moved the sampling method for the window method inside the *train()* method. Therefore, for each factor sign, a trainer is created, for which the data preparation module starts the process of

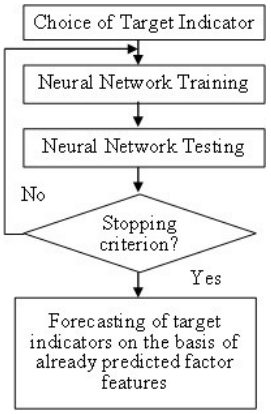

Fig. 3. Target Indicator Forecasting Scheme

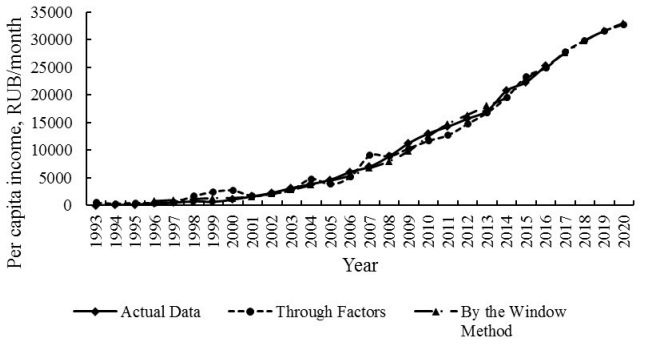

Fig. 4. Forecasting Per Capita Income

Fig. 5. Training, Testing and Forecasting Errors

generating a training sample and begins the processes of training and testing the neural network to predict the factor sign independently of other trainers.

The forecasting module consists of two parts; one part performs the prediction of factor signs, and the other – the prediction of target indicators. The process of predicting factor signs begins after training factor signs using the *predictFactorSign()* method. Depending on the chosen forecast period, an iterative process is started: the input data for the neural network made by the trainer for the factor sign is being created. The neural network produces a value that is used to create new input data that are again fed to the input of the neural network. The target indicators prediction, the scheme of which is shown in Fig. 3, is implemented by the *predict()* method, which receives the neural network as a parameter obtained during the neural network training stage to predict the target indicator. The *predict()* method passes a sample dataset through itself, which is formed from the results produced by neural networks for factor signs. The results are added to the collection, forming a set of predicted data. If there are actual values for the generated sample dataset for the forecast year, then the mean square forecast error is calculated. When the calculations are done, the forecasting module completes its work and transfers the results to the visualization module.

The visualization module allows you to display the source and normalized data, the processes of training and forecasting factor signs and targets with the simultaneous display of the neural network topology, in which you can see factor signs that have the greatest impact on the predicted target. In addition, the visualization module allows you to display graphs of training and forecasting errors. Denormalization results of predicted targets are saved in a docx file that can be used for further analysis.

VI. Experiment Results

Forecasting of indicators of the population living standards was carried out on the example of the Republic of Buryatia based on the indicators system proposed in the source [13]. Such indicators as “Average Per Capita Income”, “Unemployment Rate” and “Gini Coefficient” were used as target ones, which, in our opinion, most fully reflect the current trends in the social life of a modern society. Nineteen factors were used to predict the targets, which included both socio-economic and environmental parameters [12]. We conducted a number of experiments to show how the accuracy of training, testing and forecasting changes with the neat neuroevolutionary method being modified, the proposed methodology of forecasting the target indicator based on the factor signs preliminary prediction, as well as with the use of the feature selection method. The article shows in detail the results of experiments on the example of forecasting per capita income. Experiment 1 presents the results of forecasting “Per Capita Income” target indicator using only the Windows method and the proposed method with a preliminary prediction of the factor characteristics (Fig. 4).

The initial data for this indicator cover the period from 1993 to 2016. The graph shows the forecast for four years from 2017 to 2020. In Fig. 5 you can see the values of training, testing and forecasting errors. Training and testing errors are almost the same.

The forecasting error in the preliminary forecasting of factor features is 38% less than if we had only used the Windows method to predict the target. Since preliminary forecasting of factor signs shows the best results in the target indicator forecasting, we will employ this particular technique in our

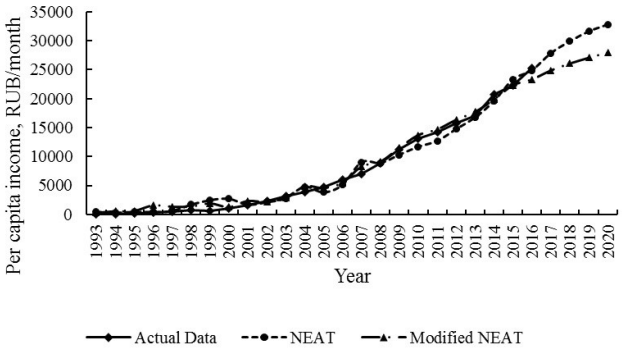

Fig. 6. Forecasting Per Capita Income

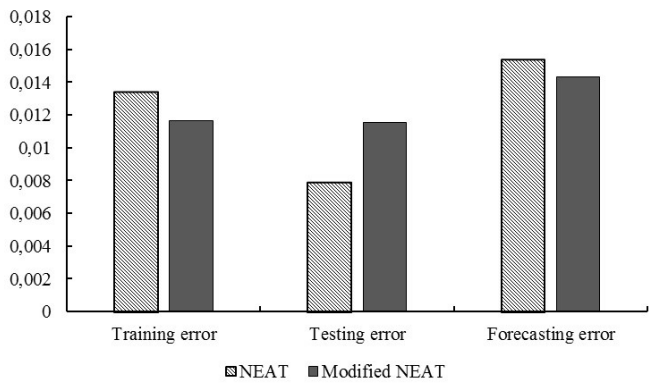

Fig. 7. Training, Testing and Forecasting Errors for NEAT and Modified NEAT Methods

further experiments. Experiment 2 demonstrates application of the modified NEAT method, which consists in the automatic selection of activation functions for the input, hidden, and output layers of a neural network. The classical NEAT method defines activation functions for each layer in advance. Fig. 6 shows the forecasting result. Fig. 7 shows a diagram of training, testing, and forecasting errors.

Fig. 7 shows that the training error is less by 15% when using the modified NEAT method. However, testing results are 45% more accurate when using the classic NEAT method. In general, the forecasting result with the modified NEAT method is 8% more accurate. Experiment 3 shows how factor features affect the target forecasting. We have determined the following conditions of the experiment:

1) All the factor features were submitted to the neural network input and were used in the target indicator forecasting, and there were 19 of them (the traditional method).
2) All factor features were fed to the neural network input, and those factor features that were automatically selected by the method of features selection were involved in forecasting [11].

Fig. 8 shows the result of this experiment, in which the following settings are applied: forecasting of the target

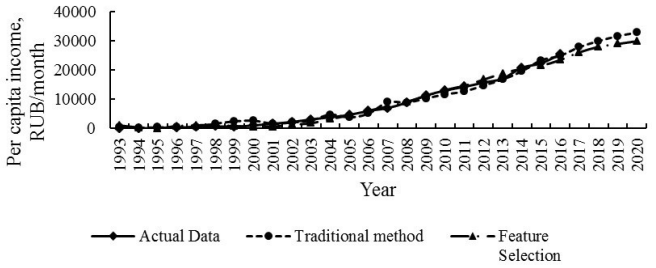

Fig. 8. Forecasting Using the Traditional Method and Feature Selection.

Fig. 9. Training, Testing and Forecasting Errors for Traditional Method and Feature Selection

indicator is carried out on the basis of preliminary forecasting of factor signs; automatic selection of factor signs by feature selection method.

The following factors influence the growth of per capita cash income in the Republic of Buryatia. Geographically Buryatia occupies a peripheral position in Russia; due to harsh climatic conditions, it is assigned to the territories of the Far North or equivalent to them. Compared with the central regions, current salary increases do not compensate for the high cost of necessary goods and services. Besides the price differentiation for consumer goods and services results in the high cost of living and has a significant impact on the living standards of the population in the Siberian regions. The obtained results of forecasting this indicator showed that a sharp refraction of the existing dynamics would not occur over the next years, since the existing trends will have an impact over the analytical period. A more accurate forecast for the growth of per capita cash income was shown by the Feature Selection method. The analysis of training, testing and forecasting errors presented in Fig. 9 shows that the training error is 44% less when using the Feature Selection method than the traditional one.

When we tested the neural network, the error of the Feature Selection method increased by 87%. However, when forecasting, this method shows the best result by 25% compared with the traditional method. It gives reason to assert that it is impossible to predict the accuracy of future forecasts at the stage of training and testing a neural network in the developed system. It is worth noting that this pattern was observed in 80% of cases of different data sets.

As it can be seen from the Figures 10–12, the neural

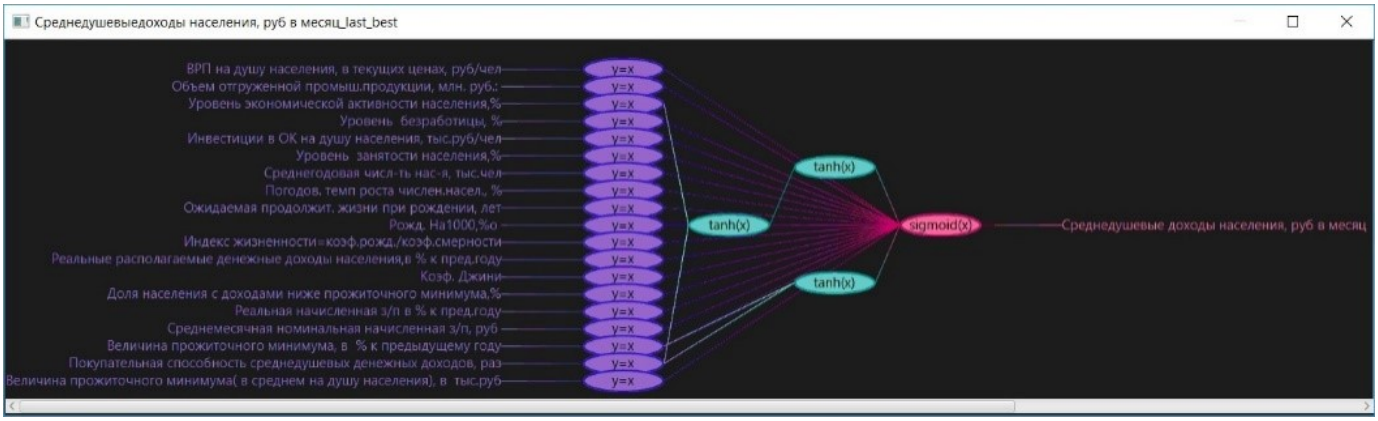

Fig. 10. Topology Generated in the First Experiment

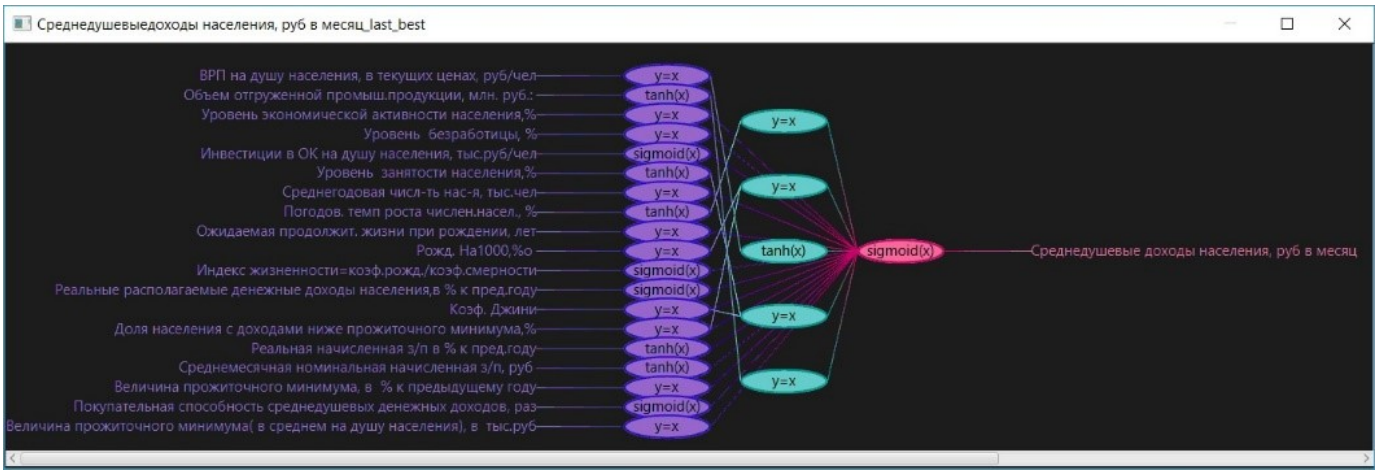

Fig. 11. Topology Generated in the Second Experiment

network topology visualization turns out to be different for each of the three carried experiments. Fig. 10 shows the neural network topology for the first experiment. There are 19 factor signs, i.e. 19 neurons, and all of them are involved in the target indicator forecasting. It should be noted that there are linear activation functions on the input layer, output layer has sigmoid functions, and the hyperbolic tangent functions are applied to two hidden layers. In the last case, the first hidden layer contains one neuron, while the second one contains two neurons.

In the second experiment the neural network architecture has one hidden layer with five neurons, four of which have a linear activation function, and one neuron has a hyperbolic tangent. 10 neurons have a linear activation function, and four neurons apply a sigmoid function in the input layer. The output layer neuron has a sigmoidal activation function (Fig. 11).

In the third experiment the neural network topology consists of input, output, and three hidden layers. The input layer has three neurons, which are the following three factors: “The Average Annual Population”, “Birth Rate Per 1000 People”, “The Average Per Capita Subsistence Minimum”, which are automatically selected by the feature selection method. It should be noted that these factors have the greatest impact on the forecast target “Per Capita Income”.

As it can be seen from the Fig. 12, the input layer has a linear activation function, the hidden layers have a hyperbolic tangent, and the output layer has a sigmoid. In addition to the “Per Capita Income” target, experiments were carried out on such indicators as “unemployment rate” and “Gini coefficient”, the forecasting of which is presented in Fig. 13 and 14, respectively.

The graph (Fig. 13) clearly tracks the dynamic changes in the “Unemployment Rate” target indicator in the Republic of Buryatia for the analyzed period. Unemployment, as measured by the methodology of the International Labor Organization at the UN, and its regional differentiation depend on many factors, such as the economic development level, remoteness from the Russian European center, peculiar natural resources (e.g. lake Baikal), demographic characteristics, employment, etc. The obtained forecast results of this indicator point that the most accurate forecast version was obtained by selecting factor signs using the feature selection method.

An important indicator characterizing the level and quality of life in the country is the Gini coefficient showing the

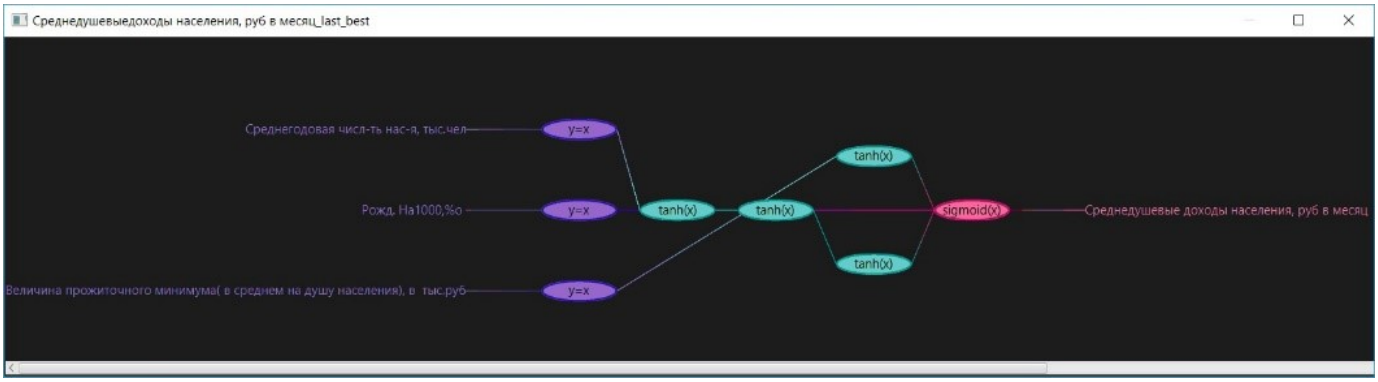

Fig. 12. Topology Generated in the Third Experiment

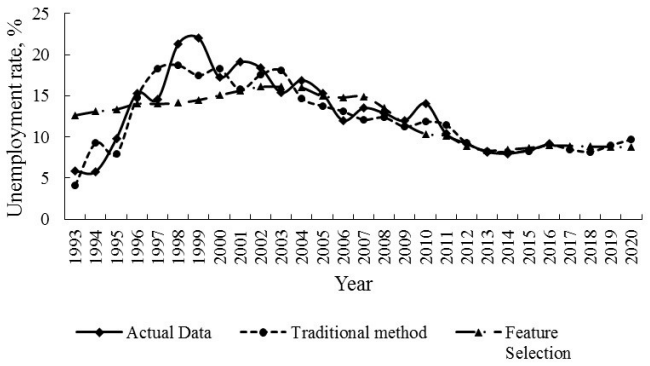

Fig. 13. Unemployment Forecasting

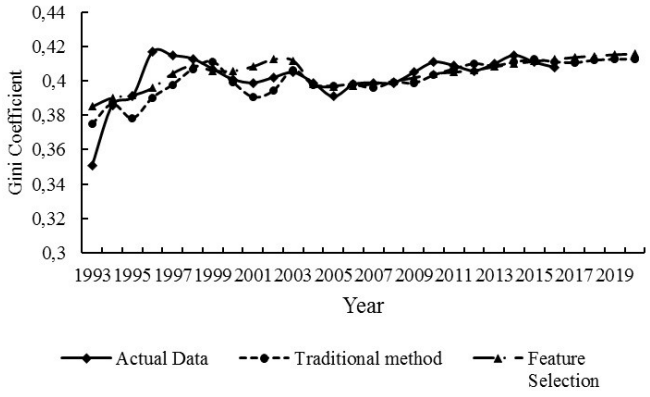

Fig. 14. Gini Coefficient Forecasting

deviation degree of the line of the actual distribution of the total income of the population from the line of their equal distribution. The Gini coefficient estimated value generally ranges from 0 to 1, and the higher the value, the more unevenly the society income is distributed. The calculations showed (Fig. 14) that during the analyzed period there was an increase in this indicator in the Republic of Buryatia, despite a slight decrease in 2016, i.e., the existing negative trend characteristic of the past period is going to continue in the forecast period in the Republic of Buryatia. The forecast accuracy results for the target indicators "Unemployment Rate" and "Gini Coefficient" are presented in the Table III.

TABLE III
TRAINING, TESTING AND FORECASTING ERRORS FOR "UNEMPLOYMENT RATE", AND "GINI COEFFICIENT" TARGET INDICATORS

Error	Through Factors	By the Window Method	Modified NEAT	Feature Selection
Target Indicator "Unemployment Rate"				
Training	0,05804	0,05804	0,05177	0,09387
Testing	0,00698	0,00698	0,01303	0,01147
Forecasting	0,04968	0,08022	0,04805	0,08281
Target Indicator "Gini Coefficient"				
Training	0,05816	0,05816	0,06404	0,06404
Testing	0,00423	0,00423	0,01979	0,01979
Forecasting	0,057	0,05356	0,0577	0,0577

The error values given in the table mean that it is not possible to choose a universal method for any data set, and it is necessary to carry out numerous experiments based on which the most suitable method is to be found. Thus, according to the statistics of the developed program use, in 90% of all cases, where a neural network trained by the Windows method was used, the greatest prediction error was got in comparison to other methods. However, in the case of "Gini Coefficient" forecasting this method ranks second in accuracy, giving way to the method of forecasting with automatic selection of activation functions (Modified NEAT). It enables to suggest that different data sets require different approaches to neural network training for forecasting performance. The list of indicators that affect the "Unemployment Rate" and "Gini Coefficient" forecasting is presented in the Table IV.

One can observe in the Table IV the most informative features for forecasting the "Unemployment Rate" and "Gini Coefficient" target indicators, obtained as a result of the factor signs selection by the feature selection method. They indicate their sufficient importance in terms of assessing the current social situation in the Republic of Buryatia. All experiments were carried out on a personal computer with the following characteristics: processor – Intel Core i7 8700k 4.90 GHz, 6

TABLE IV
SELECTED INFOMATIVE FEATURES

Indicators	Selected informative features			
	1	2	3	4
Unemployment Rate	Fixed Investment Per Capita	Average Annual Population		
Gini Coefficient	GRP Per Capita at Current Prices	Birth Rate Per 1000 Population	Real Payroll as % of the Previous Year	Purchasing Power of Per Capita Cash Income

core / 12 threads; RAM – 16GB; external memory – SSD Samsung 970 EVO 250GB; operating system – Windows 10.

VII. CONCLUSION

Using the developed software computational experiments were carried out, which made it possible to obtain predicted values of the target indicators, with the help of which one can assess the living standard of the population. The experiments results are as follows.

1) The proposed methodology for predicting the target indicator using factor signs preliminary forecasting shows a more accurate forecast compared to using only the target indicator employing the window method. However, there are exceptions to the rules;
2) The offered neuroevolutionary NEAT method modification based on the activation function mutation allows automatically building of the neural network topology to solve the forecasting problem and get a more accurate forecast than in the classic NEAT, when activation functions are set manually;
3) Selection of informative factors using the feature selection method allows obtaining the smallest prediction error of the target indicator in comparison to a selection where the genetic algorithm is applied;
4) The proposed methodology of forecasting target indicators based on preliminary forecasting of factor features made it possible to build a forecast of such indicators as “average per capita income”, “unemployment rate”, and “Gini coefficient”. The results showed that the factor signs selection with the feature selection method enabled to get a more realistic forecast of target indicators;
5) The developed methodology made it possible to clearly demonstrate the current trends in the social component of the region development for the analyzed period and these indicators changing in the future. It will allow us using them as initial data for solving the problems of analysing and forecasting the social processes development indicators on the regional level. Such an approach will make it possible to correctly and reasonably develop strategic socio-economic programs with the aim of social processes further development, as well as adjusting the current situation on the part of government authorities;
6) Visualization of the automatically generated neural network topology enables you to visualize the informative factors involved in predicting the target indicator and the activation functions distribution among the neural network layers.

REFERENCES

[1] Geron Aurelien: “Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems”. Alfa-kniga, Sankt-Peterburg, 2018
[2] J. Box, G. Jenkins: “Time series Analysis, forecast and management”. Mir, Moscow, 1974
[3] I. I. Eliseeva: “Poverty measurement in Russia: opportunities and limitations”. J.Statistics Issues (8), pp. 70–88, 2017
[4] L. A. Gladkov, V. V. Kureychik, V. M. Kureychik: “Genetic algorithms”. Physmatlit, Moscow, 2006
[5] F. Gomez, R. Miikkulainen: “Incremental Evolution of Complex General Behavior”. In: Adaptive Behavior (5), pp. 317–342, 1997
[6] M. Gronroos: “Evolutionary Design of Neural Networks”. University of Turku, Turku, 1998
[7] H. Kitano: “Designing neural network using genetic algorithm with graph generation system”. J. Complex Systems, vol. 4, pp. 461—476, 1990
[8] M. S. Kupriyanov, A. A. Barseghyan, V. V. Stepanenko, I. I. Kholod: “Data Analysis Technologies: Data Mining, Visual Mining, Text Mining, OLAP”. BHV-Petersburg, Sankt-Peterburg, 2007
[9] G. Miller, P. M. Todd, Sh. U. Hegde: “Designing neural networks using genetic algorithms”. In Scharer, JD (Ed.), Proceeding of the Third International Conference on Genetic Algorithms, pp. 379–384. Virginia, USA, 1989
[10] S. Osovskiy: “Neural networks for information processing”. Hotline-Telecom, Moscow, 2016
[11] H. C. Peng, F. Long, C. Ding: “Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy”. J. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27(8), pp. 1226–1238, 2005
[12] (2019, May 19) Russian Federal State Statistics Service (RFSSS) Homepage. [Online]. Available: http://www.gks.ru/RFSSS
[13] V. E. Saktoev, E. Ts. Sadykova: “Sustainable Development of Regional Economic Systems with Environmental Regulations”. ZAO ”Economy”, Moscow, 2011
[14] K. O. Stanley, R. Miikkulainen: “Evolving Neural Networks Through Augmenting Topologies”. Evolutionary Computation. MIT Press, Vol.10(2), Cambridge, USA, pp. 99–127, 2002
[15] Yu. P. Tsoy, V. G. Spitsyn: “An evolutionary approach to customization and learning artificial neural network”. J. Neuroinformatics, vol. 1(1), pp. 34–61, 2006
[16] N. N. Yashalova: “Development of indicators of a green economy at the regional level”. J. National interests: priorities and security, vol. 40(277), pp. 26–34, 2014
[17] N. V. Yurtikova: “Application of social indicators for poverty assessment in modern Russia”. J. Living standards of the population of Russian regions, vol. (2), pp. 95–99, 2018