STUDY OF TRAINING QUALITY OF MULTILAYER ARTIFICIAL NEURAL NETWORKS WITH VARIABLE SIGNAL CONDUCTIVITY IN SCHEDULING PROBLEMS

Alexey Olshansky

Research and Design Institute for Information Technology, Signalling and Telecommunications on Railway Transport (NIIAS), Moscow, Russia, a.olshansky@vniias.ru

Contribution to the State of the Art

https://doi.org/10.7251/JIT2302076O UDC: 004.032.26:556.131

Abstract: The paper studies approaches to railway scheduling problems using artificial neural networks (ANNs). The authors analyze traditional learning algorithms and difficulties for their application. The description of ANN's behavior is provided in the form of a phase portrait. New approaches and techniques are proposed for quality improvement of training of multilayer ANNs with variable signal conductivity.

Keywords: artificial neural networks, scheduling problems, learning algorithms, intelligent adaptive control.

INTRODUCTION

Scheduling tasks are of great importance and well known in railway transportation. The most common scheduling tasks are as follows:

- routing problems (combinatory optimization problems where a set of routes to several consumer points has to be found for a fleet of vehicles located at one or more source points);
- timetabling tasks (preparation of train timetables in such a way that they should meet all available time constraints);
- volume planning tasks (traffic distribution with the required volume of transportation taken into account);
- timetabling and volume planning (preparation of train timetables considering all possible constraints);
- volume planning and routing tasks (construction of train timetables and preparation of routes);
- other optimization tasks.

The main difficulties in solving these problems by strict methods are combinatory complexities, exhaustive searches, computer memory deficiency, and time-consuming computations to reach an optimal solution.

In this case a number of heuristic algorithms are used. For example:

- The Monte Carlo method. In scheduling problems, this method allows obtaining a series of approximated solutions, from which it is required to choose the best one. However, it is obvious that any verified change in the operation of a railway line will lead to necessarily changing the basic parameters of an algorithm, i.e. the type of a random value's distribution function, mathematical expectation, dispersion etc.
- Methods of dynamic programming (Bellman equation). The main idea of this method is the decomposition of a complex problem into a number of simple ones, whose solution is reduced to calculating a single variable. The method allows constructing a schedule using the optimality of any part of a schedule when it is optimal in general.

To sum up on the above methods, it should be noted that to achieve an optimal solution one needs to have numerous tests, and it increases time expenditures, while not giving any guarantee that an obtained solution will be close to optimality.

Approaches to scheduling problems using artificial neural networks

In this context, algorithms based on or constructed using ANN stand out. The reason is that artificial neural networks have a number of qualities inherent in a human brain and not available in classical computer architectures, capable of learning with and without a teacher, and consolidating accumulated knowledge.

Neural network solutions are applied to various types of scheduling tasks. There are various types of solutions.

David R. Martinelli and Hualiang Teng in [1] use a neural network to prepare a train formation plan. A train formation plan is the basis for further construction of freight trains' paths. They state the following task – for a given network of railway stations with existing routes, a table of requests and a given number of wagons one has to construct a train formation plan to satisfy requests.

The authors solve this problem using a multilayer perceptron. The input layer contains as many neurons as there are requests for transportation, the output layer contains as many neurons as there are possible combinations of "request – scheduled train". By training this network on existing training examples using the method of error back propagation, one can have new solutions satisfying these constraints.

The disadvantages of the method are the dependence of a neural network on the available options of a train formation plan and the necessity to retrain it every time the configuration of wagon flows changes.

For instance, papers [2-3] are based on the use of Hopfield networks. The paper [4] considers the issue of scheduling using Hopfield networks applied to the distribution of tasks across several processors. However, the method also has its drawback that is represented by a repeated use of heuristics and necessity to solve complex equations to obtain a schedule, even when some equations are already available.

To avoid problems associated with the use of Hopfield neural networks, such as long working hours and the amount of memory required, the authors of [4] propose to use the competition of neurons. The simulation results show that a competitive neural network with the constraints taken into account in the proposed energy function provides a more suitable approach to solving such a class of planning problems as traveling salesman problems.

Paper [5] considers timetabling classes at university using neural networks. The authors have adapted the Hopfield methodology to the task of timetabling classes: a neural network interpretation of the task is given, specific constraints of the energy function are considered, and the neural network is synthesized. The timetable of a certain number of classes that should be distributed by class rooms with constraints taken into consideration is presented in the form of integer piecewise-constant functions of time. As in the papers described above, the design of a neural network is selected and its energy function is configured. The authors note that they only developed an approach to construction of class timetables. They don't calculate the coefficients of the obtained function, they don't consider the issues of achieving its minimum and, respectively, they don't evaluate the quality of the resulting neural network solution.

A Hopfield neural network is often used in scheduling tasks. However, it is only used with the following restriction – of importance is only the final distribution of resources (free / busy). This network can't be applied in the following cases:

- It matters how the state of these resources changed;
- The state of a resource at a given moment of time should be equal to the state of a resource at the previous moment of time;
- These sequences should be memorized as well as shifted in a two-dimensional coordinate space (e.g. time-distance).

As it is seen, the analyzed papers reflect the implementation of a Hopfield network for processors, class timetables, but not for the systems in which the previous behavior determined the behavior at subsequent moments, and a railway line being that type of a system.

Let us consider some more materials where neural networks are applied to transport problems. In [6], neural networks, as one of the methods, are used to predict congestion of roads in the city.

The application of Hopfield neural networks to optimize airport operations (aircraft landing scheduling) is given in [7].

As a target function for an airport with different aircrafts the paper considers the minimization of a landing time interpreted as a time interval between the arrival of the first and the last aircrafts.

For this purpose, possible modes of aircraft landing – sequences – are encoded as chains. The classical function of Hopfield networks is used as an energy function, where one of the parts implies constraints (such as the impossibility of two planes to land on the same strip at the same time), and the second minimizes the landing time (using pairs of integer binary values of the output signals of Hopfield neurons).

As soon as the Hopfield network has come to a stable state, mutation is applied to the individual outputs and the process of calculation of the energy function is repeated.

This task is more like the task of constructing a schedule for a single-track railway line, however, since not all planes are connected to each other upon arrival, and the behavior of a subsequent train on a single-track line depends on the behavior of a previous one, there are substantiated assumptions that the form of an energy function when constructing a railway schedule can be different from that indicated in [7].

For instance, V. A. Kostenko and A. V. Vinokurov in their paper study the issue of scheduling by using Hopfield networks [4] applied to the distribution of tasks across several processors. A number of problems should be solved to obtain solutions for combinatory optimization problems using neural networks of this class:

- To translate a task into the “language” of neural networks means to find correspondence between the states of neurons and the values of optimized parameters.
- To construct a network energy function given the constraints and the target function. The energy function of a network at the minimum points of the target function should also have minimum points. If the constraints are violated, there should be fines increasing the value of the energy function.

There arise two complex controversial issues:

1. How can we establish correspondence between the members of a network energy function and the members of the general form of network energy?
2. How can we calculate weighting factors for penalty functions?

The author of the paper obtains coefficients for penalty functions using heuristics, noting that their values are subject to future research. Analyzing the influence of the number of processes and processors on the number of correct decisions obtained, he comes to the conclusion that the optimal algorithm for constructing a timetable will be:

a. obtaining a schedule using a heuristic algorithm;
b. obtaining bindings by the Hopfield network, provided that there are no restrictions on the order of the processes using the schedule obtained in paragraph 1, as an initial approximation;
c. obtaining order by a heuristic algorithm of local optimization.

The disadvantage of this method is the repeated use of heuristics and the need to solve complex equations to obtain a schedule, even taking into account the existing restrictions.

In addition to the use of neural networks to solve schedule tasks per se, let us consider such an aspect as modifying the designs of neural networks in terms of rebuilding internal elements and modifying the entire or partial network structure.

Multilayer artificial neural network with variable signal conductivity

One of the attempts to overcome these shortcomings in the subject field of railway transport is the development of a multilayer artificial neural network with variable signal conductivity (abbreviated as MANN VSC) to the issues of scheduling. This was first done in 2015 [8], and is currently the main source for research in the field of improving the quality of education.

MANN VSC is a hybrid neural network that combines the characteristic features of a multilayer perceptron, the Wilshaw-van der Malsburg network with the Hopfield network.

In the paper by Ignatenkov A.V. [9] the following explanations are given: "... from the point of view of the architecture of artificial neural networks, it is important to note one feature the developed neural network must satisfy: the importance of not only the value of the network error function on the output layer, but also the importance of the path from the start neuron through the layers to the final neuron of the network ...". This aspect is considered in the topology below.

The topology of the special neural network with variable signal conductivity described in [10] is given in Fig . 1

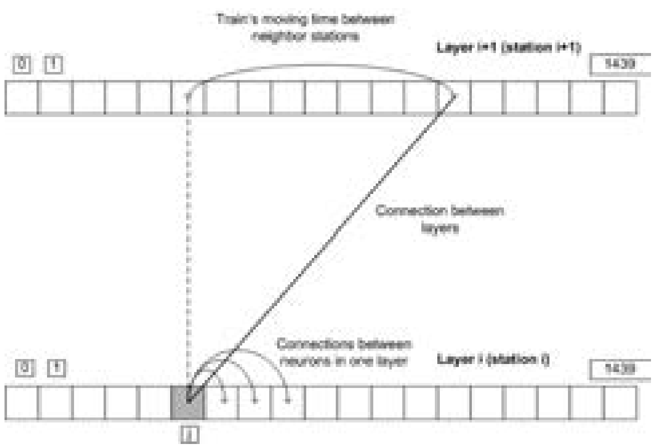


Fig 1. *The topology of MANN VSC*

The number of layers is equal to the number of railway stations. Each layer has 1440 neurons, which is equal to the number of minutes in the period of twenty-four hours.

From each neuron of the *i*-th layer, there are connections to each neuron of the next layer (a total of 1440 links). In addition, each neuron is associated with several neurons on the left (i.e., with neurons with a smaller number) and on the right (with neurons with a larger number).

Each matrix of weights W between two layers with numbers i, i+1 is a square matrix with the number of rows and columns equal to 1440.

$$W_{i,i+1} = \begin{pmatrix} w_{0,1} & \cdots & w_{0,1339} \\ \vdots & \ddots & \vdots \\ w_{1339,0} & \cdots & w_{1339,1339} \end{pmatrix},$$

where W_{ij} is the weight value on the link connecting the neuron with the i-th layer number and the neuron with the j-number of the adjacent layer.

Possible states of the neuron are: "active" – the input signal can be received at the input of the corresponding neuron, "sleep" – the value of the potential of the given neuron is zero, "off" – the neuron cannot receive signals from the previous layer. The states of "sleep" exist for both even and odd directions.

Weights of constraints are initially specified randomly by real numbers from zero to 0.1. Later they change as the neural network is trained. The transit of the signal through the connections between the neurons of neighboring layers displays the process of the train running along the path between the stations. Note that all weights of links from a neuron with number j from 0 to $j+t$ (where t is minimum travel time) are taken equal minus infinity. These weights never change.

Under normal conditions this network is trained according to several algorithms based on error back propagation and featuring some specifics:

1. MANN model for calculating the output vector of the network based on the competition of links for signal transit. Note that the values of the network's output vector and the vector defining the state of the network are different in their physical sense, i.e. the input and the output of the network illustrates the points of entry and exit of a signal, while the values of neurons activation don't have any explicit physical sense.

2. Special learning algorithms for MANN with alternate changing of weights, i.e. the weights which help to reduce MANN's errors increase while the weights preventing the reduction of errors are reduced.

Unlike existing learning algorithms, a fixed initial learning speed is used and an increase in the learning speed of the network is proposed depending on the past number of epochs. In addition, the ratio between the rate of increase of weights and the rate of decrease of weights is regulated by a special function.

When testing and using the resulting network on simulation models of railway lines in 2015-2019, various schedules were obtained with the load level of 185 processes per day. The computational complexity of the neural network is $O(m^2n) + O(mn^2)$ for m neurons in the layer and n layers.

Therefore, the issue of improving the quality of training is relevant. Some main principles of modelling control strategies for such objects were briefly described in [10]. To this end, many attempts have been made, and their results will be shown below.

Training of MANN in terms of digital signal processing

In a particular case when the error function could be described as a sum of sinusoidal harmonics with different frequencies and amplitudes we may use the results obtained in [11]. In a general case we are not sure in this signal error representation.

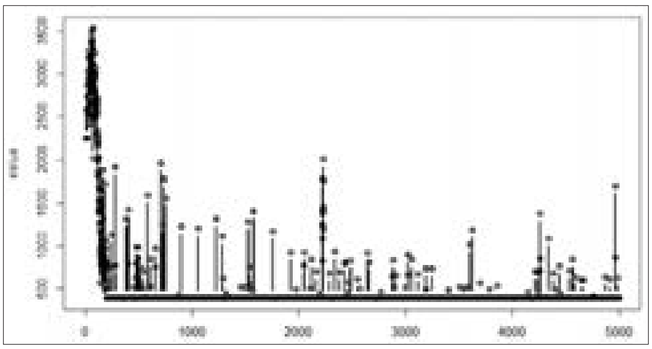


Fig. 2. *One example of error function*

To analyze the behavior of the error function we plot its autocorrelation function (Fig. 3).

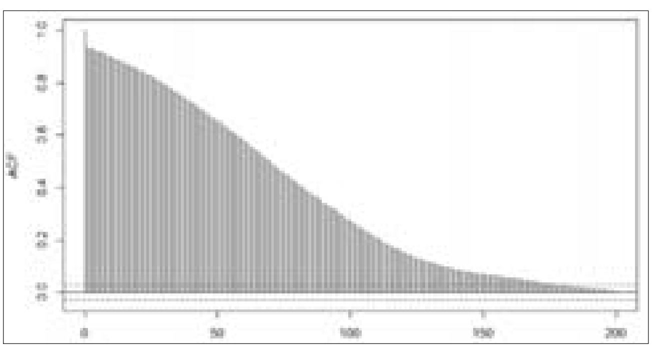


Fig. 3. *ACF of the error function*

It gave us an assumption that it is possible to decompose the signal. The goal of this decomposition is to filter the main components of the error function. After filtering we should try to implicate the decomposition for a rational control scheme to train the network.

According to the useful practice in stochastic market signal processing, we have successfully implicated LOESS techniques [12] to decompose the signal of the neural network error function. It was found that the analyzed signal consists of three perceptible components: a trend part, a periodic signal and irregular components.

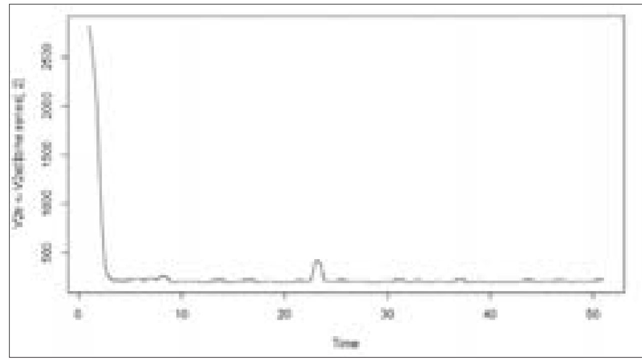


Fig. 4. *An example of STL-decomposition of the error function (trend)*

The trend curve of the neural network error function provides guides for synthesis of the rational control.

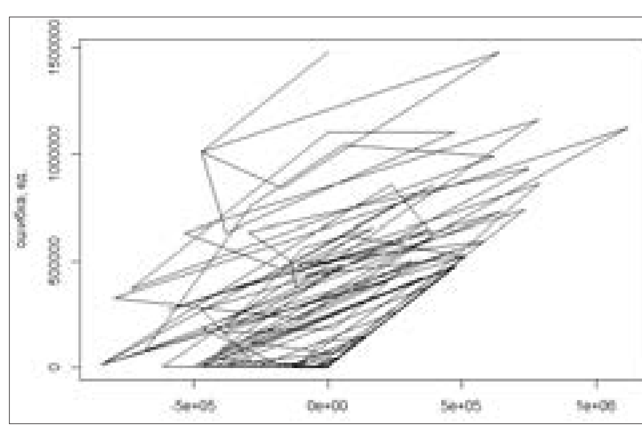
In terms of desired MANN's output this behavior can be considered as acceptable, however, the authors faced the following problem: the need to implicate one control curve using 10^{5-6} weight coefficients on average, which is computationally difficult and requires additional laborious studies in the dependence of the error function on dynamics of each weight coefficient.

Therefore, despite the controllability of such a network, as illustrated in [13], the authors developed a number of techniques for its implementation.

Post-Training Technique

To develop this technique, we carried out a generalized analysis of signal change trajectories in phase coordinates (Fig. 5).

Additionally, in order to study the behavior of the ANN as a system, phase portraits of the error of the artificial neural network in the coordinates (dE(t)/dtE(t)) for a converged and non-converged network were constructed.



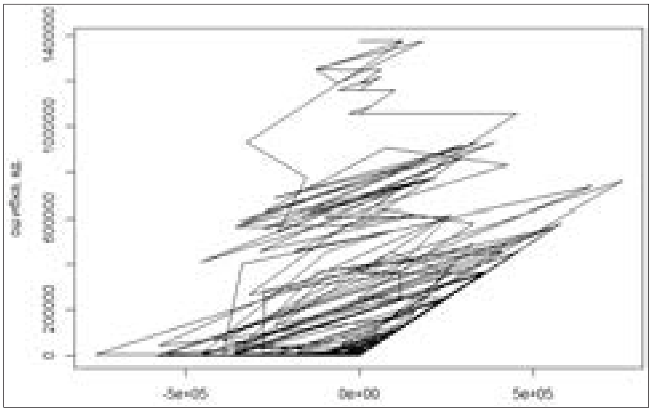


Fig. 5. *Phase portraits (left – converged network)*

There is a certain similarity of portraits, i.e. the presence of certain quasi-stable cycles and points with a low error value, which are not part of the stable cycles. In this case, the rate of change of the error changes, while being on the indicated trajectories with an increase in the total error.

In the behavior of a multilayer artificial neural network [8], sharp jumps are observed both in the error itself and in the rate of change of the error. The reason for this is the network structure and the principle of choosing the maximum connection when calculating the output. When an even signal propagates through the connections between neurons, then at the moment of activation of the neuron, according to the conditions of the problem, a dead zone occurs. The odd signal propagating through the links finds the maximum odd link, which can lead to an "off" neuron. In this case, the signal doesn't go through the given connection but uses the next largest weight connection. And this connection can be far from the point of the desired network output by a significant amount (network connections are initialized when it is created randomly). Thus, a sharp increase in the output error signal occurs.

To mitigate this effect, the model for determining the width of the training links bundle for each neuron was changed according to formula

$$s = \sqrt{E(t) + (E(t) - E(t-1))}, \quad (11)$$

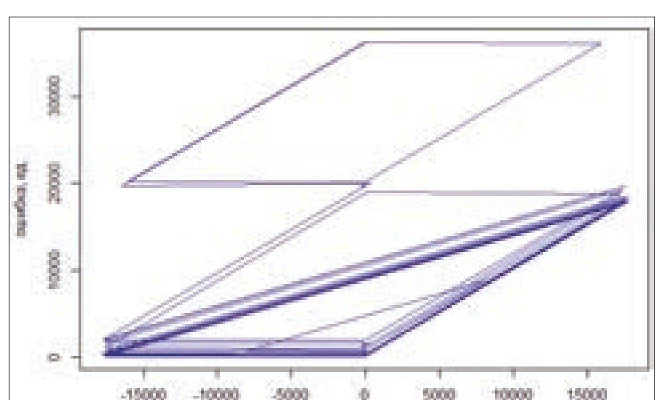
where s is the magnitude of the bundle of neuron connections for training, E (t) is the error, t is the number of the epoch.

Thus, the behavior of the network is controlled not only by an error, but also by its change in the previous step. If the network tries to increase the error, the width of the bundle of connections increases.

The essence of post-training is as follows: during the first epoch of the MANN displays primary errors and writes them as control ones. In subsequent epochs, current errors change, but before output, they are compared with the results of control errors of the previous epoch. In the case when the error of the new epoch turns out to be greater than the control error of the previous epoch, the retraining and lowering of the error value is started, thereby the network tries to reduce the error to the minimum value, simultaneously striving for a quasi-stable position. In the case when the errors of the new epoch become less than the control errors of the previous one, the control errors are rewritten. After a certain number of epochs, the ANN comes into a quasi-stable state, but jumps also occur that on average happen once per 275 epochs. Post-training is carried out 25 times, and afterwards a researcher deals with error results changed during the process.

Despite the fact that it was not possible to completely eliminate the oscillatory behavior of the network, significantly lower error values are observed and the network has been near them for a long time. This allows us to stop training such a network at the right time and get a solution for a smaller number of epochs (1.7-2 times) than it is achieved with a classical network.

The results are presented in the graphs below in Fig. 6.



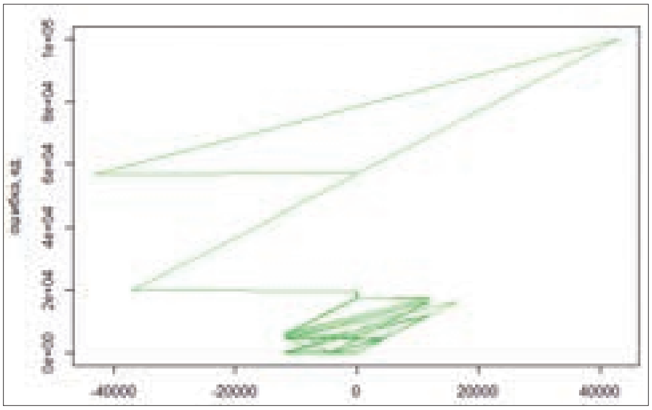


Fig. 6. Phase portrait of a network with post-training (blue – even error, green – odd error)

If phase portraits are compared before and after post-training, one can see that post-training leads to reduction in random fluctuations. However, for a signal of even errors there are two modes, with one of them being replaced by another, and for odd errors portraits give oscillations within a predefined region of the phase space.

Note that the initial design of the MANN did not have post-training, i.e. in fact had no memory. Control errors and post-training were introduced as one of the types of implicit memory. The program began to memorize the current error, then started a new epoch, received an error, and compared if it became better than the previous one. In the case of getting worse results, the retraining procedure was started until optimal results were obtained, but no more than a predetermined number of steps in post-training.

Unlike the first series of tests that were carried out on simulation models of railway lines sections, post-training was applied to the operation of the ANN on real lines of Russian Railways in the Eastern region for a network of 1920 neurons with 27 layers and a load of 170 trains per day.

The received error signal after changing the training scheme is given in Fig. 7 (1250 training epochs).

Its autocorrelation function is shown in Fig. 8.

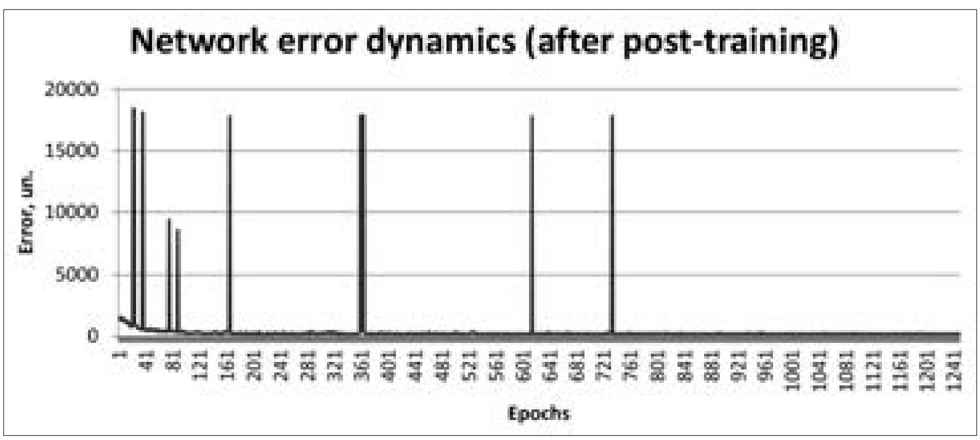


Fig. 7. Network error signal after the introduction of post-training

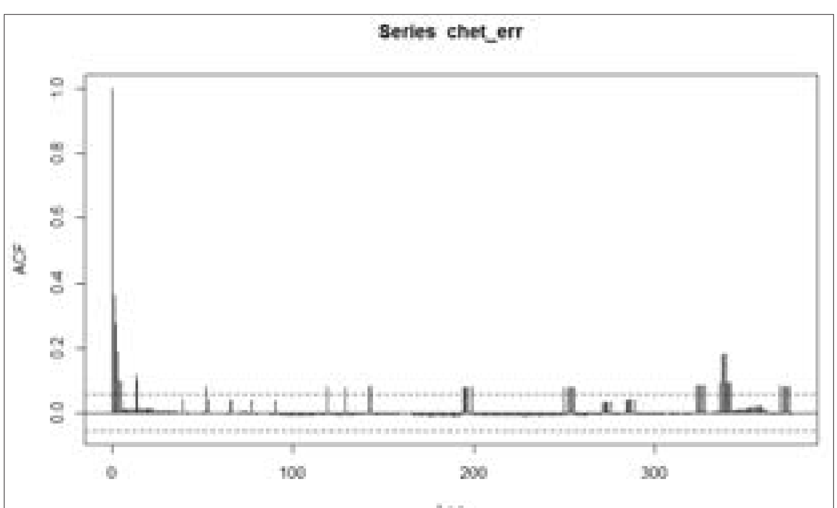


Fig. 8. Autocorrelation function of the network (with post-training)

As it can be seen from a comparison of Fig. 2 and Fig. 8, the following changes occurred in the behavior of the error signal:

- The severity of the correlation of the error signal with itself sharply decreased for time intervals of 0-60 epochs.
- Not strong, although distinct peaks are observed at intervals 13-14, 39, 52-53, 62-63. In their absolute value, they speak of an extremely weak degree of correlation.
- In general, which is confirmed by Fig. 3, the nature of the error signal became smoother, and in epochs with a significant number almost without sharp deviations and outliers.

The received new error signal was decomposed according to the LOESS method (Fig. 9) in the RStudio software environment. This method extracts the trend, periodic and residual random component from the error signal.

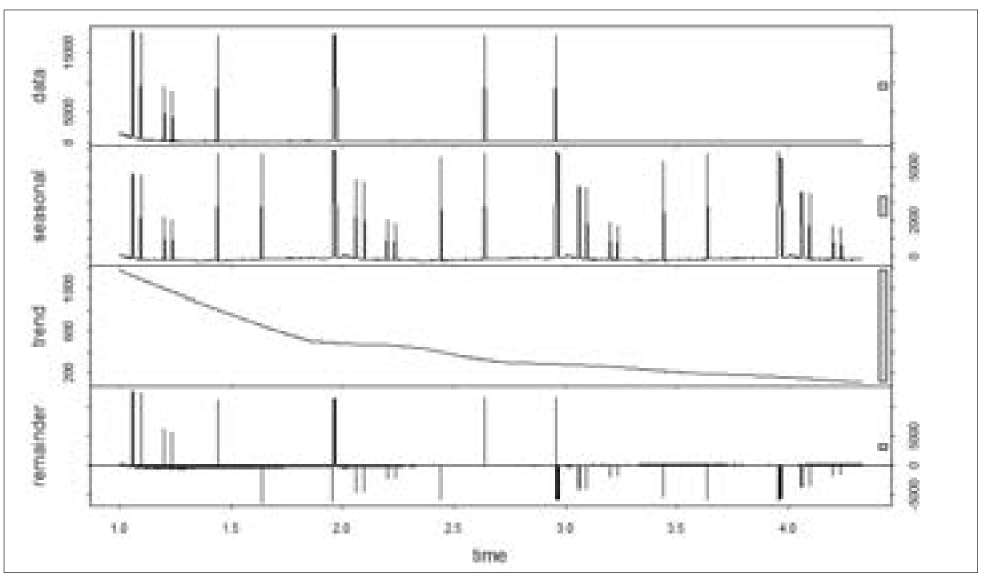


Fig. 9. *STL-decomposition of the error signal after ANN's post-training*

After receiving the results of errors of 1250 epochs, the authors conducted their analysis, built a graph, and displayed a trend. If we compare the graphs of the dynamics of the error of a multilayer ANN before and after post-training, then we can immediately note a decrease in the dynamics of the spread of an error, a transition to a more stable position of the network, stable points. Then the graphs were decomposed using the "STL" function of the "RStudio" software environment for a more detailed study, determination of the frequency of error occurrence, and identification of the causes of a sharp jump in error growth even from relatively stable positions.

It is seen that there is a significant correlation between the overshoots of the error signal in a number of decomposition residues and in the periodic component. At the same time, the contribution of the trend is low (1/15 at the beginning of training).

In this regard, it is necessary to create a mechanism for controlling the state of the ANN, which, without fixing a complete set of bond weights in each epoch, would eliminate the influence of the periodic and residual components. Such a problem can be solved, for example, using approaches to the synthesis of optimal regulators.

PID control

The application of a trend component to control such an ANN consists in the fact that its values can be used at the stage of training the network in question as the reference values of a control error, which will reduce the training time and improve the quality of constructed timetables.

In search for various ways to control the error signal, the idea was proposed to apply a PID controller that would correct the output signal trying to reduce the error of the output data and reducing the possibility of scatter and jumps.

PID control of the ANN error signal is implemented according to the following formula:

$$G(s) = K_1 + \frac{K_2}{s} + K_3 s$$

where s is the argument of the transfer function, K_1 is the coefficient of proportional regulation, K_2 is the coefficient of integral regulation, K_3 is the coefficient of differential regulation.

The PID controller algorithm is implemented in the programming language R in the RStudio environment.

As a setup, the approximation of the network error signal is set by an exponential function of the form Ae^{-kt}, where t is the number of the network operation epochs, A is the initial value of the network error with which training starts, and k is the coefficient of the degree of error attenuation. The choice of such a setting function is due to the fact that it corresponds to the most common form of error reduction in the training of traditional neural networks.

For the MANN, consisting of 27 layers and 1920 neurons in each layer and with 175 schedules as a load, the error signal without control changed according to Fig. 10.

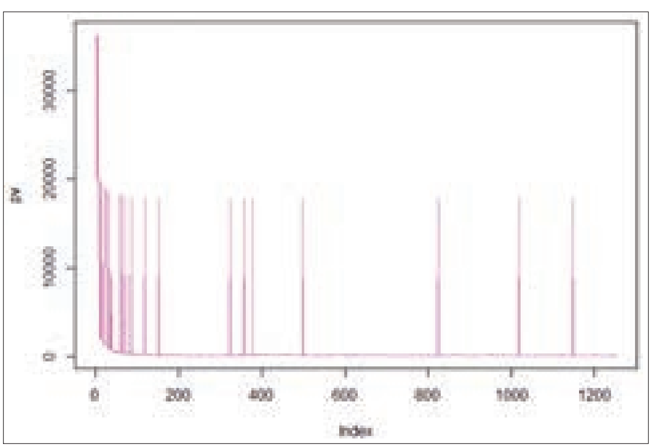


Fig. 10. *Network error signal*

The desired error change signal is shown in Fig. 11.

Application of PID controller to the error signal with parameters $K_p = 10$, $K_i = 1$, $K_d = 0.01$ showed that the control curve for the network error signal should look like this (Fig .12):

The algorithmic implementation of such control is carried out using the built-in techniques of the ANN and special techniques, including post-training etc.

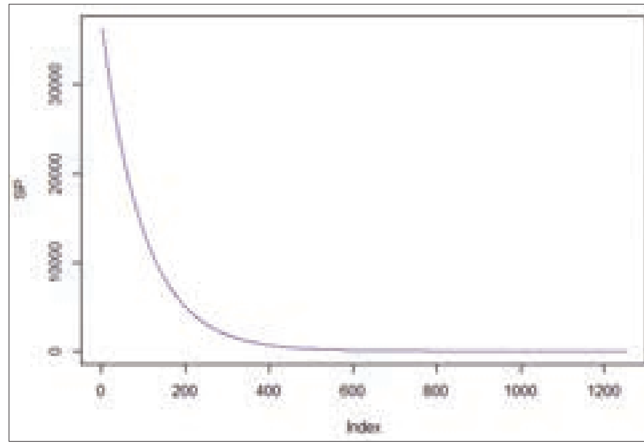


Fig. 11. *Setup change*

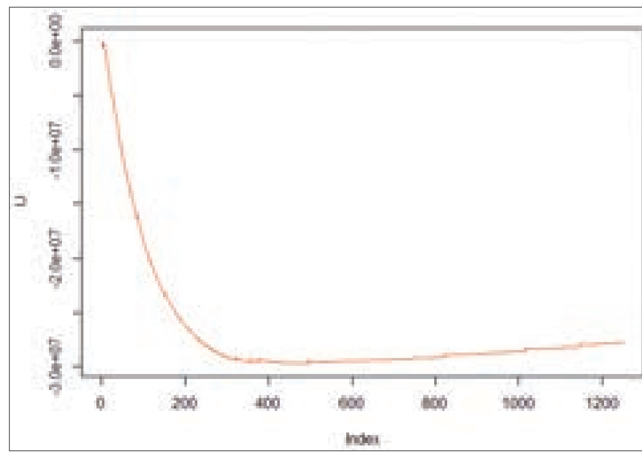


Fig. 12. *Control curve for a given network*

An analysis of Fig. 12 shows that the control found is quite adequate to the existing behavior of the ANN error.

A promising development of this control is the introduction of decomposition and prediction of an error signal in real time in combination with the applied PID controller.

For the developed control system, the price of sustainability will be performed as follows. For each epoch, an additional perturbation is introduced into the error signal in the work of the MANN. The magnitude of this perturbation is within [5 ... 100] and changes with step 5. The parameters of the PID controller during testing (Table 2, Columns 2,3) were initially set as:

PID controller of even error:

$$K_p = 0.1,\ K_I = 10,\ K_D = 0.5$$

PID controller of odd error:

$$K_P = 0.3,\ K_I = 30,\ K_D = 0.5$$

Table 1.

The magnitude of the disturbance, units/ epochs.	Test Series No. 1				Test series No. 2			
	Even		Odd		Even		Odd	
	% dist	% resp	% dist	% resp	% dist	% resp	% dist	% resp
1	2	3	4	5	6	7	8	9
5	0.31	101.83	0.4	104.15	0.28	100.42	0.33	98.8
10	0.56	103.01	0.74	105.53	0.55	101.91	0.65	102.00
15	0.89	104.19	1.23	106.91	0.88	102.76	0.95	105.01
				…				
95	5.18	120.95	7.16	140.09	5.15	119.11	5.80	122.85
100	5.24	122.88	6.89	144.24	5.53	120.81	6.09	128.46

The second series of tests (table 2, article 4.5):
***New** controller coefficients for an even error:*

$$K_P = 0.5, K_I = 30, K_D = 0.5$$

***New** controller coefficients for odd error:*

$$K_P = 1.5, K_I = 30, K_D = 0.5$$

The results are presented in Table 1.

Table 1 will be read as follows. The value of 0.31 in the column "%" returns indicates that when a disturbance value of 0.31% of the average error is added to the error signal, its output value in the column "% resp" increased by 1.83% from the average.

This was done for error signals in the even and odd directions.

Compared to the first series of tests, in the second series, the average response to disturbance decreased by 2%, which is insufficient.

To assess the stability value, let us introduce the stability coefficient (quality), which is calculated by the formula below:

$$K_{st} = \frac{K_{dist}}{K_{resp}},$$

Where:
K_{st} – *coefficient of stability (quality),*
K_{dist} – *coefficient of disturbance,*
K_{resp} – *coefficient ofresponse.*

The results are presented in Table 2.

The quality factor should be no more than 1, which means that the disturbance provided led to a response not exceeding in strength. Such modes with controller parameters $K_P = 1.5$, $K_I = 30$, $K_D = 0.5$ are observed in 2 cases out of 20.

For the purpose of identifying the most effective training modes of the network with the PID controller, a second series of tests was carried out, during which the disturbance applied to the network input for 4 error signals varied from 0 to 55 with the step of 5, and the PID controller coefficients changed according to the following scheme:

- K_P: from 0.1 to 1 in increments of 0.2;
- K_I: from 10 to 40 in increments of 10;
- K_D: from 0.1 to 4.1 in increments of 1;

As in the first series, the values of the stability coefficients and the disturbance coefficient were calculated.

Table 2.

The magnitude of the disturbance, units /epochs	Even (small)	Odd (small)	Even New	Odd New
1	2	3	4	5
5	5.96	10.45	1.49	-3.69
10	5.39	7.48	3.48	3.07
15	4.70	5.64	3.14	5.29
		…		
85	4.66	5.96	3.6	7.20
90	4.09	5.83	3.9	5.18
95	4.04	5.60	3.71	3.94
100	4.37	6.42	3.77	4.67

The best solutions for even and odd errors are shown in the figures below (Fig. 13 and Fig. 14).

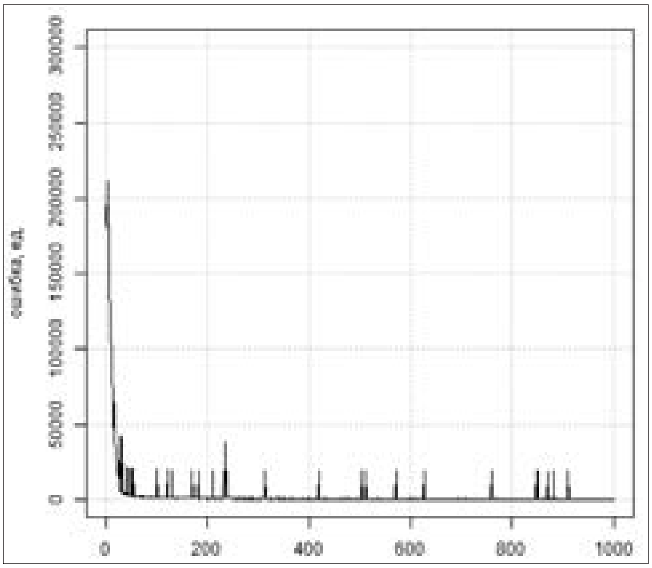


Fig. 13. *Dynamics of a clear network error with the parameters of the PID controller*
$K_P = 0.1$, $K_I = 40$, $K_D = 2.1$ with disturbance of 5

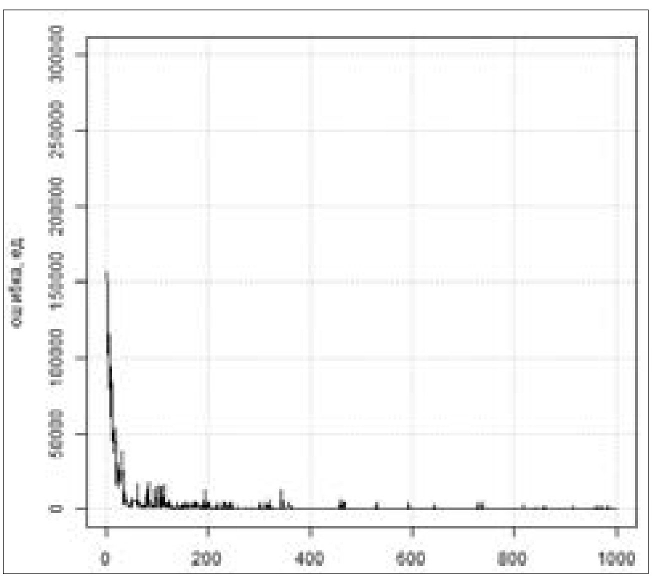


Fig. 14. *Dynamics of an even network error with the parameters of the PID controller*
$K_P = 0.1$, $K_I = 40$, $K_D = 2.1$ with disturbance of 5

Thus, we can draw the following conclusions:

1) A total of 1100 launches were carried out, of which 136 were sustainable;

2) The behavior of this network with a PID controller can be considered stable only for weak disturbances, the magnitude of which does not exceed 10-15% of the final steady-state network error;

3) The most stable mode under disturbance of 5 is the regime with the following coefficients:

- for an even controller, $K_P = 0.1$, $K_I = 40$, $K_D = 2.1$;
- for an odd controller, $K_P = 0.3$, $K_I = 10$, $K_D = 2.1$.

In addition to the PID controller and post-training with control errors or with a floating range of the links bundle for training, an alternative approach to training of such networks is proposed.

MANN control using a three-layer perceptron

[14] describes the idea of training an ANN through parallel training of two networks at once, in which the second ANN runs through a lot of epochs, calculating errors, comparing them with the given parameters. If the results are satisfactory, then these values are transferred to the first ANN. In this case, the second ANN acts as an invisible duplicating neuron in the main network, in which all calculations and determination of the best result are performed.

For the presented ANN design, such an algorithm is depicted in Fig.15.

We will implement this scheme in practice with a slight change using direct or inverse training of one neural network (MANN) using a multilayer perceptron.

The control ANN learns from a set of triples ("Error at the time" - "Status at the last moment" - "Control signal from the past at the given time") or ("Error at the past" - "Error now" - "Control").

The error output and the error itself (with a delay) are fed to the trained control network. The response of the control network is fed to the residual error and to the executive algorithm. Subsequently, the residual error integrator provides a signal to the control ANN as well.

As a practical implementation of the control found, we can use any of the algorithms (PID, post-training) to transmit a control signal. The error at the moment of time plus the magnitude of the control signal is the target error transmitted to the actuator.

Figures 17-18 present the graphs of an error under direct neurocontrol provided that the MANN is used separately for different flows of a software system – for freight and passenger flows.

Compared to the previously developed post-training techniques, the number of overshoots and their frequency, as well as the absolute value of the

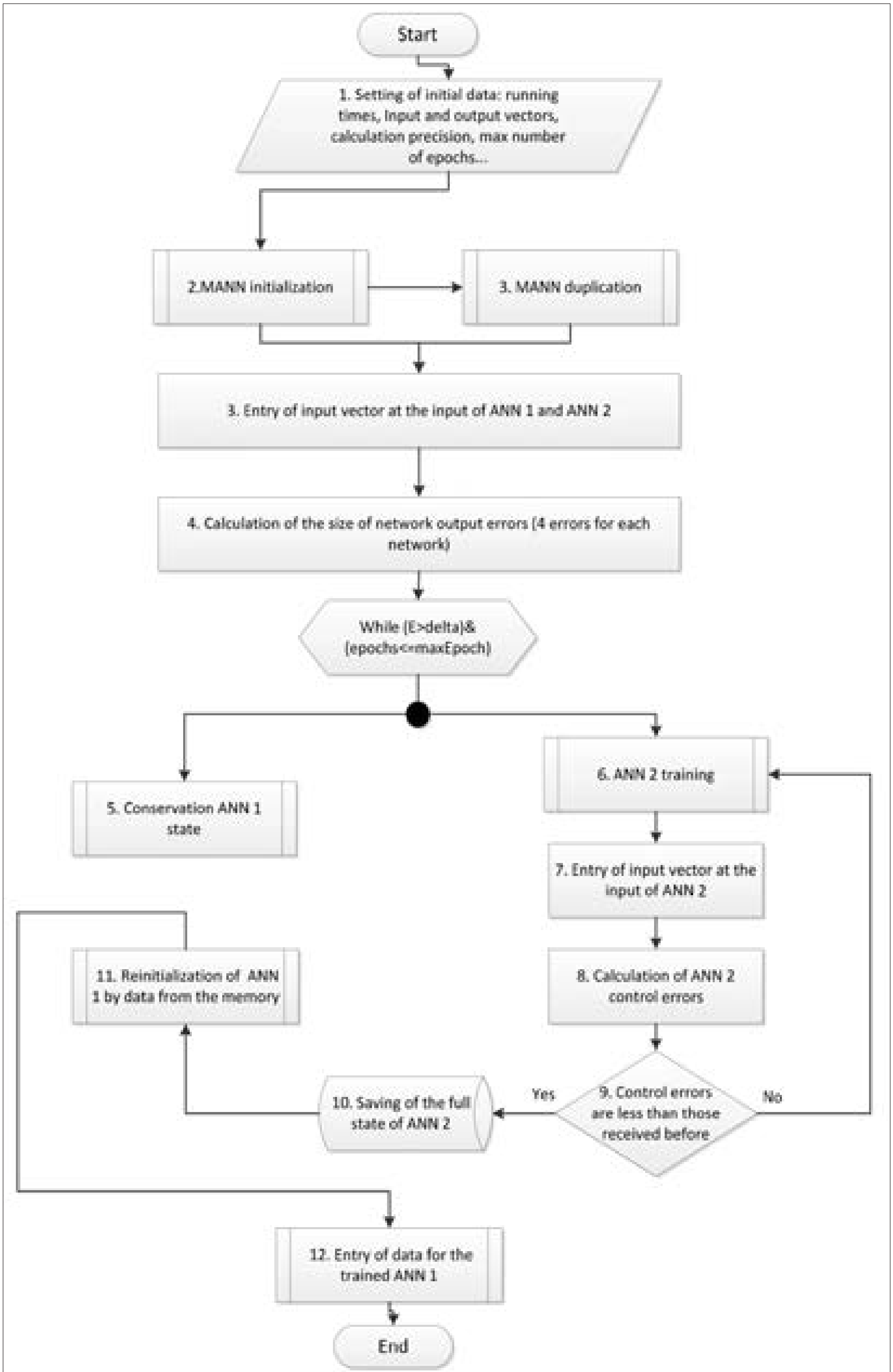


Fig.15. An *enlarged algorithm for simultaneous training of two networks*

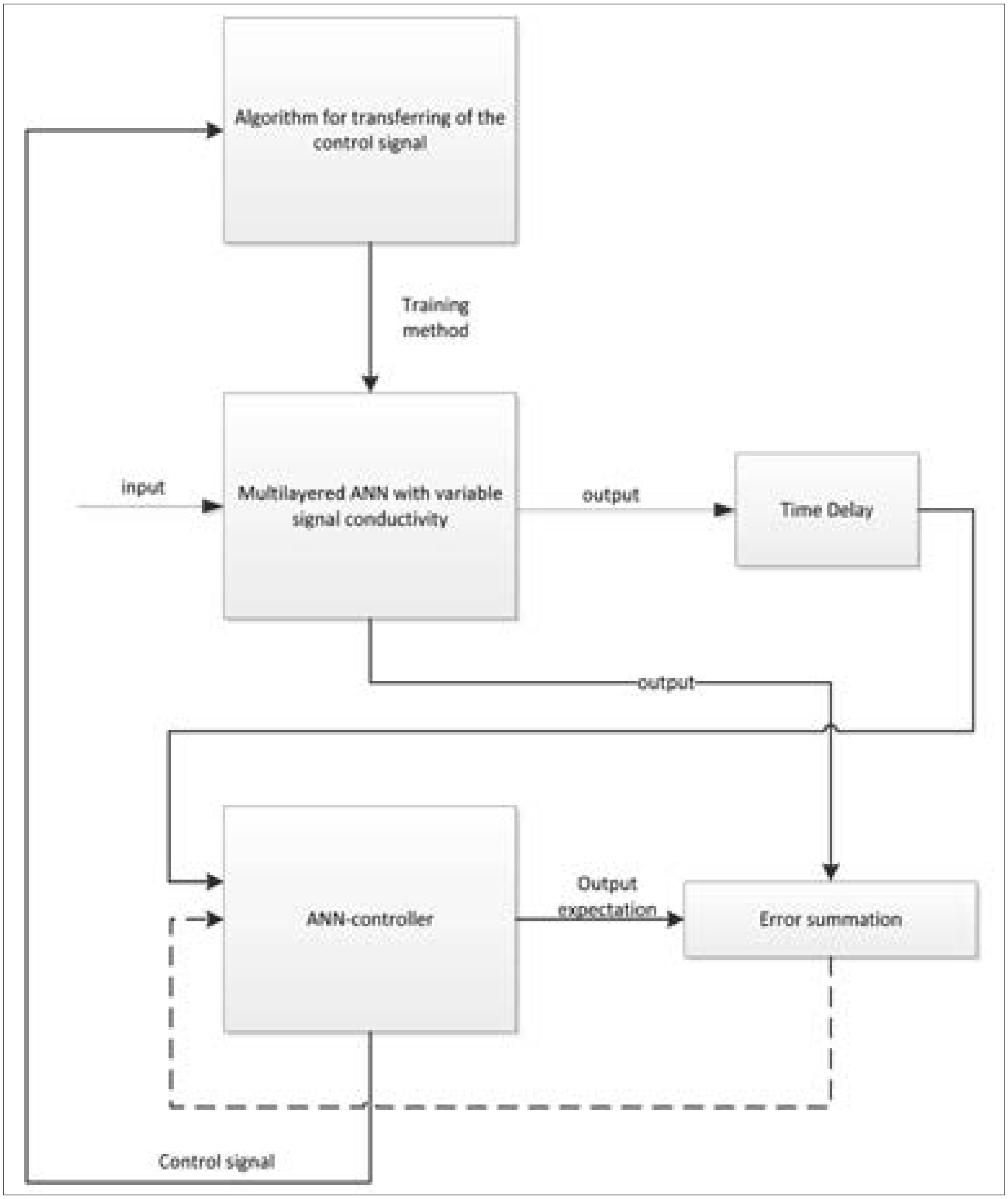


Fig. 16. *Neurocontrol scheme*

error values during overshoots were significantly reduced.

In some cases, a gain of the control signal is applied as:

$$K(t,E) = K_0 \pm \alpha * E(t),\ u(t) = K(t,E) * U(t)$$

where $E(t)$ is the magnitude of the current error, α is the proportionality coefficient, K_0 is the initial gain, $U(t)$ is the change in the error signal predicted by the multilayer perceptron, $K(t,E)$ is the resulting gain, u(t) is the final control signal.

As before, for testing of such approaches we chose the timetable for the line, where there are 185 trains, 27 stations and 24 hours.

Comparison shows that the introduction of amplification in neurocontrol in some cases reduces the amount of residual errors in the steady state mode of ANN operation. However, this decrease is not so great to speak about the significant difference between the control techniques in question.

In general, the comparative effectiveness of MINS training methods is given in Table 5:

Table 3. *Comparison of neurocontrol techniques (perceptron) for a multi-layered network for passenger trains*

Error	No gain		Linear gain	
	Even	Odd	Even	Odd
Median	152	54	155	53
Average	967	1070	1200	1225
Maximum	37050	56280	37040	56230
Minimum	152	0	152	0

Table 4. *Comparison of neurocontrol techniques (perceptron) for a multi-layered network for freight trains*

Error	No gain		Linear gain	
	Even	Odd	Even	Odd
Median	210	76	218	80
Average	756	717	746	802
Maximum	37040	56300	36940	56370
Minimum	184	38	185	32

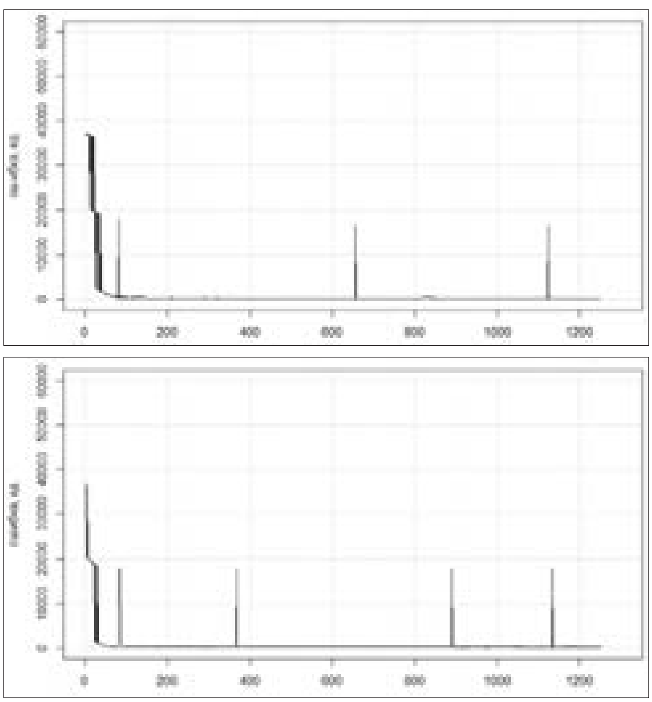


Fig. 17. *Network error dynamics (even error signal)*

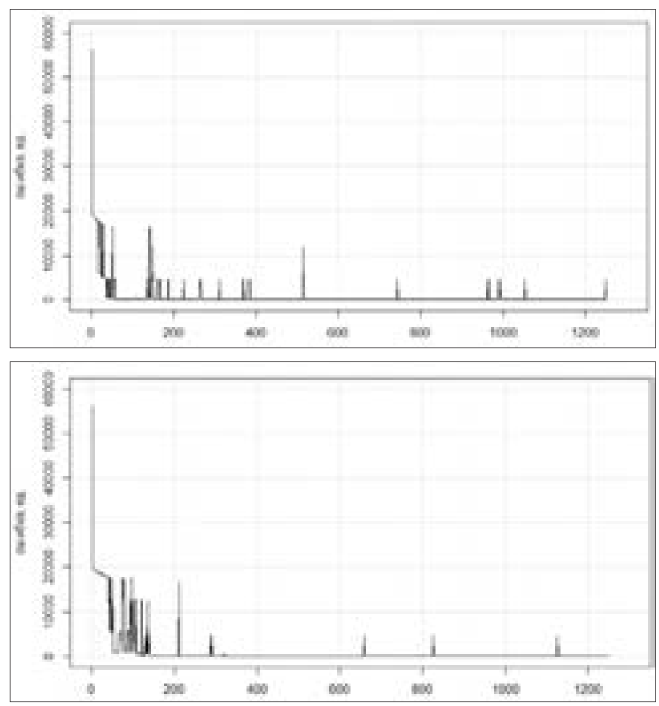


Fig. 18. *Network error dynamics (odd error signal)*

Table 5. *Comparative effectiveness of teaching methods MINS*

Indicator	Common Learning Algorithms	Post-training	PID controller	ANN (direct neurocontrol)
Minimum	75	0	362	193
Maximum	134795	54392	211585	57895
Median	5469	265	471	210
Average value	16548	1240	1830	384
SD	6687	4621	4485	1180
Burst Frequency for 100 epochs	50	3	15	0.4

Conclusion

1. The article shows the application of an artificial neural network with variable signal conductivity for solving the schedule problem.
2. The authors developed additional algorithms for managing the learning of a multilayer ANN with variable signal conductivity. The most effective from the standpoint of convergence to the solution are direct neurocontrol and post-training.
3. Direct neurocontrol also gives the smallest spread in the dynamics of error signals and the smallest value of the median error, i.e. 50 per cent of all estimates are in the range from 0 to 210.
4. A comparison of the methods showed the advantages of direct neurocontrol; however, in order to develop the most rational network operation modes, computational experiments for different tasks and network sections are required.

References

[1] Martinelli D.R. Optimization of railway operations using neural networks / D.R. Martinelli, H. Teng // Transpn. Res.-C. - 1996. - Vol. 4, No. 1. P. 33–49.

[2] Hopfield J. J. "Neural" Computation of Decisions in Optimization Problems / J.J. Hopfield, D.W. Tank // Biological Cybernetics. - 1985. - Vol. 52, Issue 3. - P. 141–152.

[3] Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities / JJ Hopfield // Proc. Natl. Acad. Sci. USA - 1982. - Vol. 79, No. 8. - P. 2554–2558.

[4] Kostenko, V. A. Local-optimal scheduling algorithms based on the use of Hopfield networks / V. A. Kostenko, A. V. Vinokurov // Programming. - 2003. - No. 4. - P. 27–40 (in Russian).

[5] Yakovlev, V.V. Solution of the task of timetabling classes at university using neural networks /V.V. Yakovlev, Yu. G. Vasiliev, B.M. Kalmykov // Bulletin of the Chuvash University. - 2008. - No. 2. [Electronic resource]. - URL: http://cyberleninka.ru/article/n/reshenie-zadachi-sostavleniya-raspisaniya-zanyatiy-v-vuze-s-pomoschyu-neyropodobnyh-sete y

[6] Cavaliery S. Optimal path determination in a graph by Hopfield neural Network / S. Cavaliery, A. Di Stefano, O. Mirabella // Neural Networks. - 1994. - Vol. 7, No. 2. - P. 397–404.

[7] Agafonov, A. A. Mathematical models and adaptive methods for short-term forecasting of traffic parameters: Thesis: 05.13.18 / Agafonov Anton Aleksandrovich. - Samara, 2014.- 164 p. (in Russian).

[8] Olshansky A.M., Ignatenkov A.V. One approach to control of a neural network with variable signal conductivity // Information Technologies and Nanotechnologies (ITNT-2017). - 201.- P. 984 -987.

[9] Ignatenkov A.V. Neural network modeling and streamlining of traffic flows on railway lines. Abstract of Thesis, 05.13.18. - Mathematical modeling, numerical methods and program complexes // Penza, PenzGU, 2018.-20 p., P. 14 (in Russian).

[10] Rozenberg Yefim, Olshansky Alexey et al. Some approaches to improving the quality of artificial neural network training// https://ceur-ws.org/Vol-2667/paper7.pdf, pp.27-29.

[11] IgnatenkovA.V. Model iskusstvennoy neyronnoy seti dlya postroeniya graphika dvizheniya poezdov na dvukhputnom uchastke / A.V. Ignatenkov // International Scientific Conference Proceedings "Advanced Information Technologies and Scientific Computing". - Samara, Samara Scientific Center of RAS Publishing, 2016 - P.619-623. - ISBN 978-5-93424-758-5 (in Russian).

[12] Olshansky, A. M., Ignatenkov, A. V. Development of an artificial neural network for constructing a train schedule // Bulletin of the Ryazan State Radio Engineering University (RGRU-2016). - 2016. - No. 55. - P. 73-80 (in Russian).

[13] Cleveland RB et al. STL: A Seasonal-Trend Decomposition // Journal of Official Statistics. - 1990. - T. 6. - No. 1. - P. 3-73.

[14] Géron Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc., 2022.

Received: September 7, 2023
Accepted: November 21, 2023

About the authors

Alexey Olshansky is the Head of Centre for Mathematical and Computer Simulation of JSC NIIAS, PhD in Engineering. His areas of interest are system analysis, theory of automatic control of socio-economic systems, artificial neural networks and heuristic methods for designing control systems. He is the author of over 150 scientific papers and 20 patented inventions.

For citation

Alexey Olshansky, Study of Training Quality of Multilayer Artificial Neural Networks with Variable Signal Conductivity in Scheduling Problems, *JITA – Journal of Information Technology and Applications, Banja Luka*, Pan-Europien University APEIRON, Banja Luka, Republika Srpska, Bosna i Hercegovina, JITA 13(2023) 2:76-90, (UDC: 004.032.26:556.131), (DOI: 10.7251/JIT2302076O, Volume 13, Number 2, Banja Luka, December (57-120), ISSN 2232-9625 (print), ISSN 2233-0194 (online), UDC 004