Vladislav SKORPIL, Roman PRECECHTEL

Brno University of Technology

Training a Neural Network for a New Node Element Design

Abstract *This paper describes training a neural network for a new node element design. Neural networks are used to control network nodes instead of the control by sequential circuits. This article primarily discusses the management of routers. Routers are generally able to use all available topologies of a network. Simulation in MATLAB-SIMULINK has tested how the training networks route the given packets. It has been shown that Artificial Neural Networks (ANN) are suitable for the described aim.*

Streszczenie. *W artykule opisano proces trenowania sieci neuronowej, wykorzystanej na potrzeby budowy nowego węzła sieci. Głównym zastosowaniem, omówionym w publikacji jest zarządzanie routerem. Badania symulacyjne przeprowadzone w programie Matlab-Simulink, potwierdziły skuteczność działania sztucznych sieci neuronowych w tej aplikacji. (**Proces trenowania sieci neuronowej do tworzenia nowego węzła sieci**).*

Keywords: Artificial Neural Network, Network Node, MATLAB –SIMULINK, Routing.
Słowa kluczowe: Sztuczne Sieci Neuronowe, węzeł sieci, Matlab-Simulink, routowanie.

Introduction

The work is based on [1] and [2]. An important part of communication and information networks is active network elements that allow you to route data from one place to another. Some services have a higher priority than others, in particular services in real time. The best-known example is the transfer of voice over the Internet (VoIP). These and other types of traffic make ever more exacting demands on active network elements, which must provide correct routing, sufficient bandwidth, and also ensure the required parameters for the connection of the quality of services (QoS). This is one of the main reasons for the development of telecommunications and permanent search for new ways of increasing their throughput and efficiency of [3].

Another important point of development is, for example, the increasing routing tables according to which the routers determine the path for each link [4]. The essence of active network elements is to provide a link for each device in the network and allow their mutual communication. Network elements transmit data units from the sender to the recipient on the basis of processing the results in the control unit of the network element [5]. A simplification of the process of processing the packets would be the omission of the requirements of the priorities for the management of individual requirements. It is requirement for QoS that makes great demands on the logical management of active control elements of the network. There are several options how to solve the problem of optimizing the management of the active element. This is a very complex and challenging task, with several conflicting requirements such as speed of processing the request, response to changes in network topology, memory size, price, etc [3].

Given the complexity of the problem, an appropriate solution method is processing via multiple parallel processes. This method is, inter alia, possible and also characteristic of neural networks, both organic and artificial. The latter, the case of artificial neural networks, is the subject of this work, [6]. It differentiates it from the classical approach, which is based on the sequential processing of incoming data [6].

State of the art

All communication networks are based on active elements, which process the transmitted data units, and on the basis of the processing results they pass the data units in the direction from the sender to the receiver. Today the most demanding task of active elements is determining at which instant of time and which data unit the system should process such the processing corresponds to the priority assigned to individual data units.

Classical sequential data processing is constrained by the speed of central processing unit. Increasing demands on the speed of processors place ever more increasing qualitative demands on the production technology. Another solution is offered by a parallel data processing. In the parallel sequencing of more complex functional elements, the demands on controlling their cooperation are greatly increased, so the increase in the effectiveness of the whole system is not so pronounced. A more advantageous possibility is frequent parallel interconnection of simpler function blocks with distributed memory. Coordinating their mutual cooperation is simpler and thus the overall effectiveness is greater than in the case of more complex elements.

An example of the second method, i.e. parallel interconnection of a great number of comparatively simple function blocks, can be seen in the artificial neural network. Artificial neural network is an array of mutually interconnected elementary function blocks with a special architecture; in many cases it offers a very promising solution to data processing. Based on neural cells of living organisms, a number of different types of neural network has been created. The paper concentrates on various backpropagation neural networks, which can be used to solve optimization tasks.

On the basic of the architecture of active elements it can be said that neural networks could be used very effectively for their control. The aim of the paper is therefore to seek an alternative way of the increase in the performance of systems whose activities can be realized by using the parallel architecture. In addition, the system in question must within its activity find a solution to the optimization problem. This intention characterizes relatively well the requirements on modern active network elements.

Implementation of the management

For the implementation of the management of the active element, programmable logical fields are reckoned with, which are formed by a field of programmable cells [7]. When they are connected appropriately, they provide the required logical function while the subsequent hardware implementation allows creating a specific active element capable of controlling the operation of the network [8]. The logical schema of such a connection, according to which a model would be created for processing in the form of a chip, is the subject of this work with a suitable simulations leading to an optimal solution.

Management of routing using artificial intelligence

The aim of this paper is to replace the controlling logical unit inside an active network element by a neural network [9]. Fig. 1 shows the basic schema of individual internal blocks. Input ports transmit incoming data to a central memory, which can have different forms of data processing or various types of queues [10]. The type of queue is one of the essential parameters of the proposal of logical routing management. Here, the basic model is FIFO (First In First Out), namely for each input port a separate queue [11].

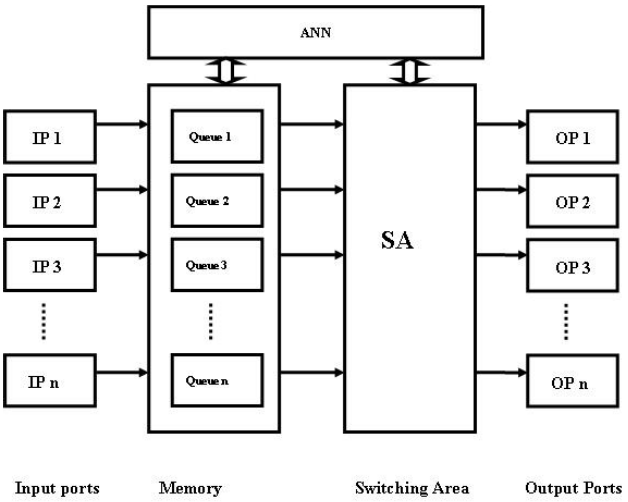

Fig. 1. Router block diagram
IP-Input Ports, OP-Output Ports, SA-Sampling Area,
ANN — Artificial Neural Network, Pr-priorities

Illustrative example of the principle of routing with priorities

The router receives packets at entry ports 1, 2 and 3. The packets are marked 112, 112, 112 in the first queue, 214, 211 in the second queue, and 311, 322, 322 in the third queue. The first number is the input port, the second is the number of the output port, and the third number gives the priority. The highest priority is 1. Packets from the first port thus have an address that corresponds to output port 1 and priority 2, the second highest. At the same time there is in the queue of the second input port also a packet with the destination address port 1, but with priority 1, i.e. the highest priority. On the third input port there is, at the same time as the previous two packets, a packet of priority 2 addressed to output port 2. Other ports at the same time detect no traffic.

Fig. 2 gives the central memory with packets. There is a collision on outgoing port number 1, when at the same time two different packets with different priorities are bound for this port. It is also possible that two or more packets with the same priority are on one output port at the same time. In this case there is again a number of options how the control unit can respond [10]. One of the variants is the random selection and subsequent switch, possibly also a decision based on another parameter such as the similarity of packets from the previous switch, etc. A packet collision on a single default port is shown in Fig.3.

It is desirable for the control unit in the form of ANN to be able to evaluate this situation as follows: the packet from input port 2 with priority 1 and the packet from input port 3 with priority 2 will be switched and their queue will be shifted by one packet. This is indicated by arrows in Fig. 4. The packet from queue number 1, which remained non-switched, will have, thanks to lower significance priority, the priority number reduced by one degree, as shown in Fig. 4 in the first queue, and the evaluation of the whole process is repeated. In the next step the packets from queue 1 and 3 will be switched.

The packet from queue 2 has this time priority 4 and must therefore give priority to a packet with higher routing priority (in this case, 1 is the highest priority for routing and takes priority over the others). At the same time, priority 4 is again reduced to 3 in the packet from queue 2, which had to give priority to the packet from queue 1. In the next instant of time, unless another packet comes, the two remaining packets will be routed simultaneously, since there is no collision on the output port and priority will play no role in this case.

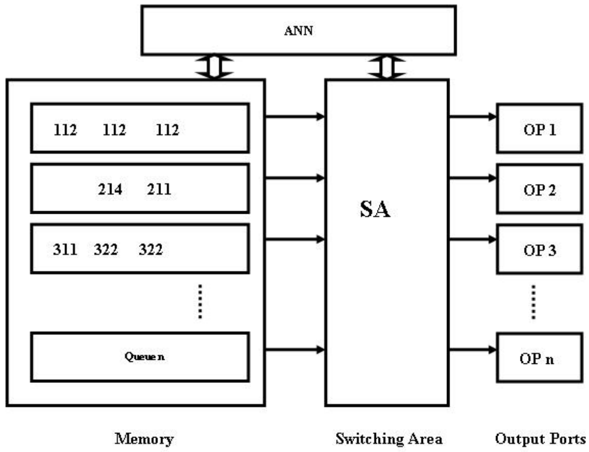

Fig.2. Central memory with packets

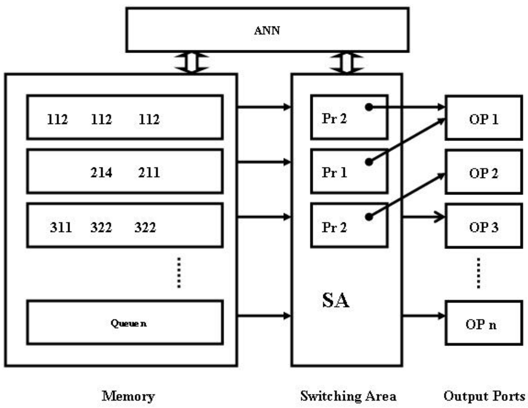

Fig. 3. Collision on outgoing port with different priority

Source & target training data

The most important issue for an overall routing management design of the active nodal control element is the proposal of logical control unit representing ANN and its so-called training [11]. Training is a process in which we place on the network input so-called training input and simultaneously output data, or the required data, i.e. the desired response to the entry of ANN. By so-called training process, ANN is trying to reach a consensus between the specified output data and the result of the calculation after the passage of training data through the network [12], in other words, it wants to learn the transfer function of the system. For a successful training it is therefore necessary to choose not only a correct ANN and its setting, but also appropriate training data [5] for ANN to be able to correctly set the network parameters or the priorities. However, even so we have no guarantee that the result will be acceptable; it will move in the range of permissible deviations from the required value. Table 2 shows the examples of training data.

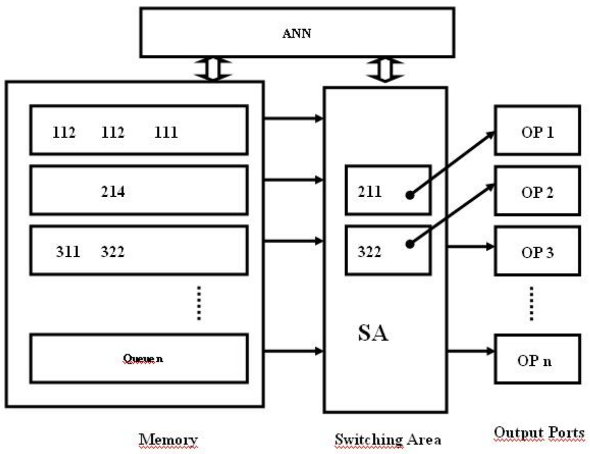

Fig. 4. Switch of packets with a higher priority and reduction of priority of non-switched packet

The Index column is designed to differentiate individual time cycles. It combines the data that belong together and distinguishes them from other data. If the index is not present, ANN is not able to distinguish which data cohere with each other and cannot learn the required routing logic. The basic idea of the nature of the submission of the training data is such that ANN should be able to learn the required routing logic. It is further stated that this index manifests itself in the decisions of the ANN in routing and is responsible for the routing error.

The Input Port column represents the data entry into the active network element.

The Output Port column indicates the port into which the packet should be routed from an input port. In a real case, it is the IP address, network mask, from which the switch determines the outgoing port by using a routing table.

The Priority column specifies the priority of the packet. One is of the highest priority and has precedence over all other priorities. For convenience, only four priorities are listed.

The Routing column determines whether or not to switch the packet to the output port. One means yes, minus one means no, the packet must wait. A different packet with a higher priority is routed first.

In the MATLAB there are training data from a CSV file, which is located in the same directory as the script that reads the data. Reading data proceeds from the first row and first column in the following order: column A is the Index, column B is the input port, column C is the output port, column D is the priority, and column E is the routing.

For the purpose of retrieving data a training script is created to retrieve data from the CSV file (Table 1.). The script loads the CSV file and saves it in a corresponding file with a mat suffix, and also creates two defined variables: source as the source data and target as the target data. If the file already exists, it is overwritten by new data. These data are further used in training ANN. The script reads the CSV file from the first character and transforms data for correctly entering ANN [13].

The training of ANN (Fig.6) is performed by the script, which first loads the column and target variables, which the script created for loading, and uses them as source and target data for the defined ANN. It is possible in the script to change all the network settings. The parameters that are not listed are set to the default values preset by the MATLAB.

Table 1. CSV file with training data

	A	B	C	D	E
1	0	1	1	1	1
2	0	1	1	2	-1
3	0	1	1	3	-1
4	1	1	2	1	1
5	1	1	2	2	-1
6	1	1	2	3	-1
7	2	1	3	1	1
8	2	1	3	2	-1
9	2	1	3	3	-1
10	3	1	1	1	1
11	3	1	2	2	1
12	3	1	3	3	1
13	4	1	1	3	1
14	4	1	2	2	1
15	4	1	3	1	1
16	5	1	1	2	1
17	5	1	2	1	1
18	5	1	3	3	1

For the simulation of forward neural networks (Fig. 5) we start from its typical characteristics that the linking of neurons is carried out only between the neurons of adjacent layers and that is in one direction. This means that in individual layers the neurons are not connected with one another. Numerous hidden layers and a large number of neurons in the hidden layers have often resulted in the so-called. "over-training" of the network, which does not lead to better results and improvement of the model of the system [14].

Training by using the Back Propagation Algorithm consists, in that file, of training set patterns repeatedly attached to the network inputs and the optimum setting of trigger function parameters which is looked for and in which the error between the input training and output source data will be minimum.

A training set can be obtained by calculation, logical functions, or experimentally directly by measuring the values of the parameter of a modelled process. In our case it is the routing of packets in the network node active element. Each passage of the training set over the neural network is called an epoch. The number of epochs can be set as the final parameter to end the simulation if the priority conditions of error size are not met. The number of passages need to be derived from the graph and from the experience of the given work (Fig. 7).

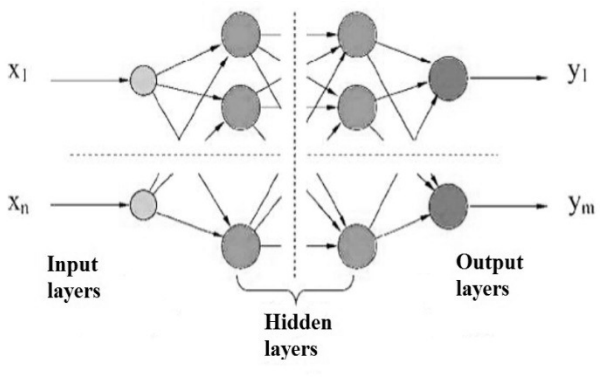

Fig. 5. Topology of a forward neural network

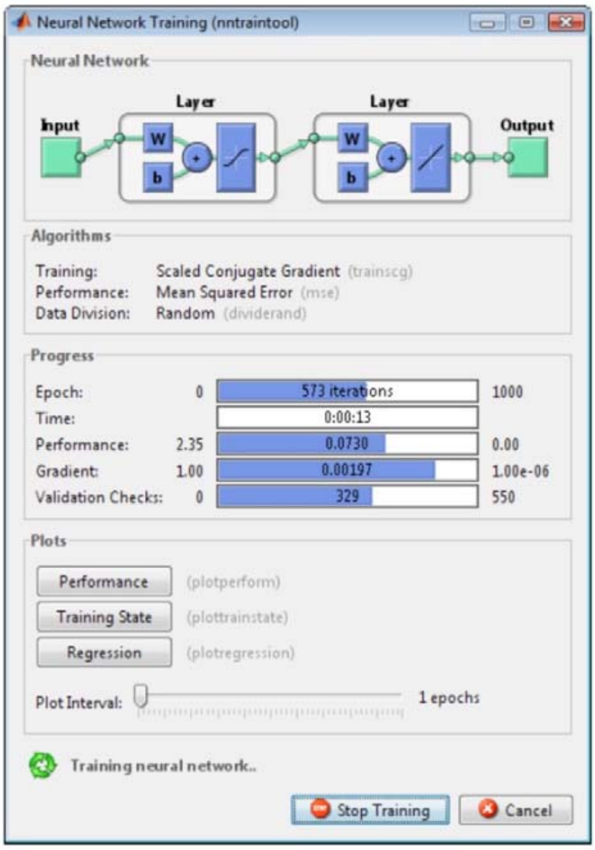

Fig. 6. ANN Training

A training set can be obtained by calculation, logical functions, or experimentally directly by measuring the values of the parameter of a modelled process. In our case it is the routing of packets in the network node active element. Each passage of the training set over the neural network is called an epoch. The number of epochs can be set as the final parameter to the end of the simulation if the priority conditions of error size are not met. The number of passages need to be derived from the graph and from the experience of the given work (Fig. 7).

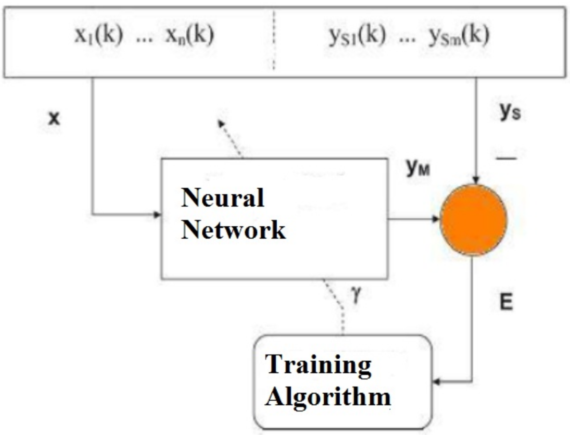

Fig. 7. Back-propagation Algorithm

The last script is designed to start quickly and check the teaching of ANN. It first runs the script for data reading, then it starts the script for the training, and finally the script for the simulation. In the script for the quick start and check of teaching the ANN the possibility of testing a single time iteration of incoming packets is also included, where the influence of the indexes on the routing is highlighted.

Table 2. An example of training data
IP – Input Port, OP – Output Port, Pr – Priority, Rou – Routing

Index	IP	OP	Pr	Rout
0	1	1	1	1
0	1	1	2	-1
0	1	1	3	-1
1	1	2	1	1
1	1	2	2	-1
1	1	2	3	-1
2	1	3	1	1
2	1	3	2	-1
2	1	3	3	-1
3	1	1	1	1
3	1	2	2	1
3	1	3	3	1
4	2	1	3	1
4	2	2	2	1
4	2	3	1	1
5	2	1	2	1
5	2	2	1	1
5	2	3	3	1

Examples of training data

Fig. 8 - Fig. 14 show the selected graphical results of MATLAB simulation. The number of layers ANN are 2 or 3, the number of neurons transmission function are 20, 10 or 5, transmission functions tansig, puklin, training function is trainscg, or trainbfg. We simulated Feed-forward back-propagation network, Cascade-forward backpropagation network, Elman backpropagation network for all above descripted ANN. The results demonstrate big mean squared errors, approximately 10^{-1}.

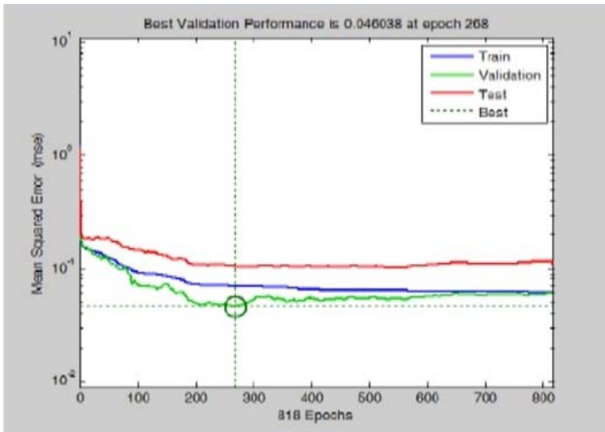

Fig. 8. ANN, 2 layers, 20 and 10 neurons, transmission functions: tansig and puklin, training function trainscg, Elman back-propagation network

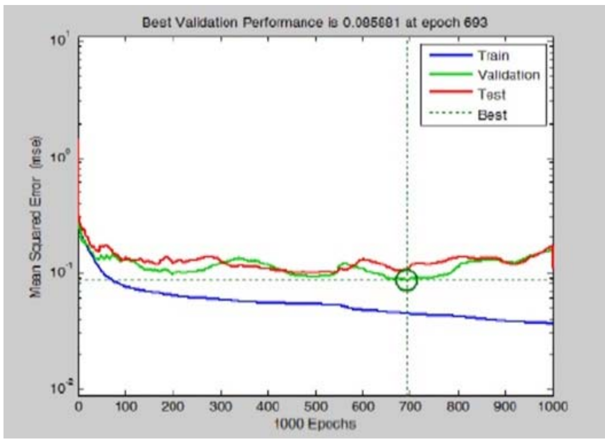

Fig. 9. ANN, 3 layers, 20 10 and 5 neurons, transmission functions: tansig and puklin, training function trainscg, Feed-forward back-propagation network

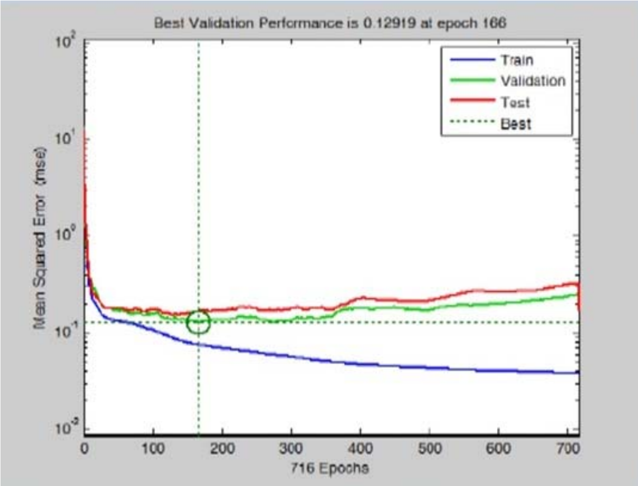

Fig. 10. ANN, 3 layers, 20 10 and 5 neurons, transmission functions: tansig and puklin, training function trainscg, Cascade-forward back-propagation network

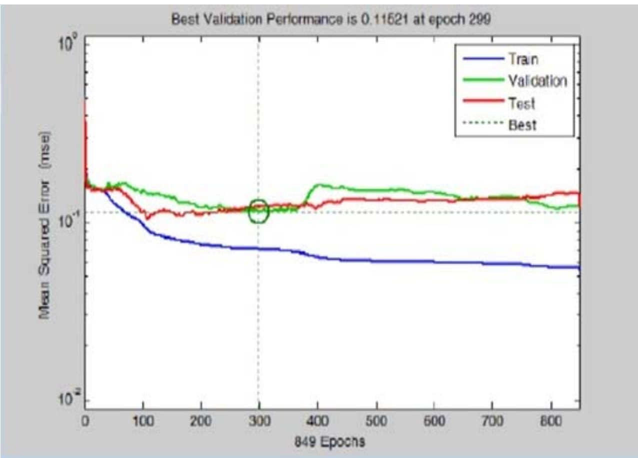

Fig. 11. ANN, 3 layers, 20 10 and 5 neurons, transmission functions: tansig and puklin, training function trainscg, Elman back-propagation network

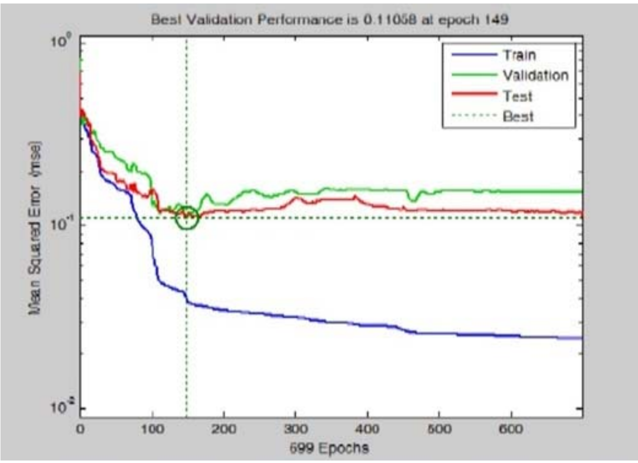

Fig. 12. ANN, 2 layers, 10 and 5 neurons, transmission functions: tansig and tansig, training function trainscg, Cascade-forward back-propagation network

Conclusion

The aim of this paper is to create an active network element with the logic of routing the switching operation by using artificial neural network, and to analyze and test the possibilities of using such networks for the problem solution.

The tool used for testing and analysis is MATLAB-SIMULINK with specialized extension Neural Network Toolbox for neural networks. The subject of testing is forward neural networks. The requirement is to create a logical routing with artificial intelligence, at least as efficient as that implemented currently via digital circuits. The most difficult part of the whole research is network training for the required accuracy and the bulk of the work devoted to this issue**.**

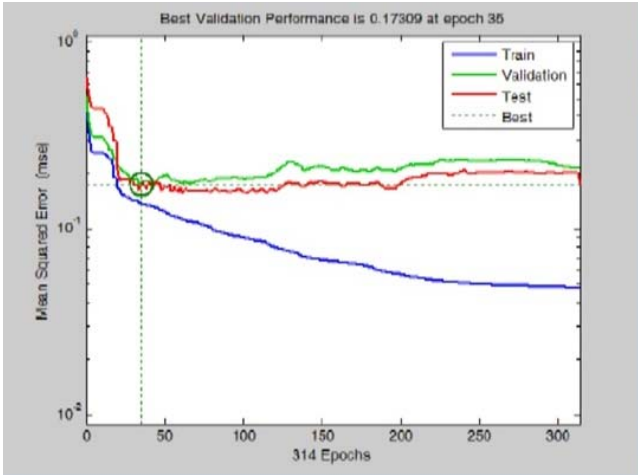

Fig. 13. ANN, 2 layers, 10 and 5 neurons, transmission functions: tansig and tansig, training function trainscg, Elman back-propagation network

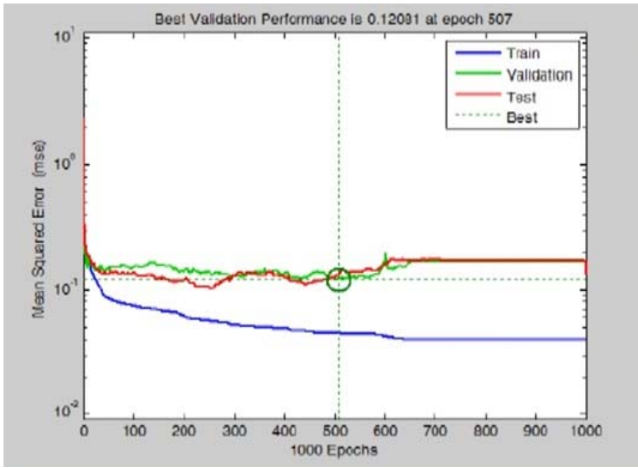

Fig. 14. ANN, 2 layers, 10 and 5 neurons, transmission functions: tansig and puklin, training function trainbfg, Feed-forward back-propagation network

The aim of the work was to create a new architecture of the management of active nodal network element. The issue of active network elements is briefly discussed with the emphasis on routing. The above theoretical analysis is complemented with the modelling of network elements in the MATLAB-SIMULINK program. The essence of the paper is to create a model of a logical management of the packet-switching and supporting priority processing. In the practical part, attention was given to finding the idea in the optimization of the setting of ANN, or its training, was to create a training set with the indexes of individual training models correct network settings. The entire work is an example of processing solutions in the field of artificial neural networks. The main, with forward neural networks having been selected for the training. Both of these ideas were aimed at exploring not very standard way of solving the problem and finding a completely new method of solution, which would ultimately contribute to the final solution. A new architecture has thus been proposed for the control of network elements but the proposed procedure does not yet meet the error-rate conditions. Reducing the error-rate will be the subject of further research.

This research was supported by the grants:
No CZ.1.07/2.2.00/28.0062, Joint activities of BUT and TUO while creating the content of accredited technical courses in ICT,
No. ED 2.1.00/03.0072, Centre of sensor information and communication systems (SIX)
No. FEKT-S-11-15 Research of electronic communication systems.

REFERENCES

[1] Skorpil,V.,Precechtel,R. Management of Routing Using Artificial Intelligence. In Proceedings of the 11th International Conference Knowledge in Telecommunication Technologies and Optics – KTTO 2011. pp. 29-32, June 22-24, 2011, Szczyrk, Poland, ISBN 978-80-248-2399-7
[2] Precechtel, R. Optimalization Drive of the Active Network Element. BUT, Brno 2009
[3] Demuth,H., Beale,M., Neural Network Toolbox for Use with MATLAB. Natick (USA): The MathWorks, Inc., 1994.
[4] Takizava and a. Fukasawa, Novel neural Network Scheme Composed of Prediction and Studies. *Proceedings of the 13 th WSEAS International Conference on systems*. Rodos, Greece, Iceland, WSEAS, pp. 611-615, Rodos, 2009, ISBN: 978-960-474-097-0, ISSN: 1790-2769 |
[5] Tuckova,J. Introduction to the theory and application of Artificial neural networks. CTU, Prague 2003. 103s. ISBN 80-01-02800-3.
[6] L.P.S. Fernandez and A.R. Ruiz and J.de J.M. Juarez, "Urban Noise Permanent Monitoring and Pattern Recognition". *Proceedings of the European Conference of Communications – ECCOM´10.* NAUN, Tenerife, Spain, Puerto De La Cruz 2010, pp. 143 - 148 ISBN: 978-960-474-250-9
[7] Bishop Artificial Intelligence I, II, III, IV. Academia, Prague, 1993, 1997, 2001, 2003.
[8] Danilo, P. Mandic,P., Recurrent neural networks for prediction, learning algorithms, architectures and stability. John Wiley, Chichester, 2001.
[9] Norgaard, M., Neural Networks for Modelling and Control of Dynamic Systems. Springer, London 2000.
[10] Drabek, O., Taufer,I. Seidl, P., Artificial neural networks--Theory and Application. CHEMagazin 1 (XVI), 2006, p. 12-14. ISSN 1210-7409.
[11] Drabek,P., Taufer,I., Seidl,P., Artificial neural networks--Theory and Application. CHEMagazin 5 (XVI), 2006, p. 6-8. ISSN 1210-7409.
[12] Snorek,M. Jirina,M., Neural Networks and Neural-copmputers. CTU, Prague 1996.124 s. ISBN 80-01--01455-X.
[13] Susnea and a. Filipescu and v. Minzu and g., Vasil, Virtual Pheromones and Neural Networks Based Wheeled Mobile Robot Control. *Proceedings of the 13 th WSEAS International Conference on systems*. Rodos, Greece, Iceland, WSEAS, pp. 511-516, Rodos, 2009, ISBN: 978-960-474-097-0, ISSN: 1790-2769 |
[14] Bogdamov and r. Mirsu and v. Tiponut, Matlab Model for Spiking Neural Networks, *Proceedings of the 13 th WSEAS International Conference on systems.* Rodos, Greece, Iceland, WSEAS, pp. 533-537, Rodos, 2009, ISBN: 978-960-474-097-0, ISSN: 1790-2769 |

***Authors**: Assoc. Prof. Ing. Vladislav Skorpil, CSc., MSc. Roman Precechtel, Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunication, Purkynova 118, 612 00 Brno, Czech Republic, E-mail: skorpil@feec.vutbr.cz.*