Journal of Theoretical and Applied Information Technology
15th October 2021. Vol.99. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A METHOD OF DESIGN OF NEURAL NETWORKS BUILT ON FPGA

[1]A.V. KOVALEV, [2]O.B. SPIRIDONOV, [3]I.E. LYSENKO, [4]O.A. EZHOVA
[1,2,3,4]Southern Federal University, Rostov-on-Don, Russia
E-mail: ingvarlys@gmail.com

ABSTRACT

Artificial intelligence technologies are based on the use of neural networks. At the same time, the actual issue is the implementation of neural networks on various software and hardware platforms. In this paper, we consider an effective method to design fully connected neural networks or convolutional neural networks and to implement these networks on FPGAs using the Xilinx System Generator for DSP and Matlab/Simulink package. The developed method is design of neural networks using the Xilinx System Generator for DSP library imported into the Matlab/Simulink environment and using the built-in MatlabC language to generate a given ANN neural networks structure automatically. The artificial neural networks designed according to this method are easily reconfigurable. Also such neural networks allow solving the following tasks: image recognition, optimal filtering. This method based on neural networks gives an opportunity to improve recognition procedures of technical conditions of monitor and diagnostics of complex objects. Also such methods could be a practical alternative of strict mathematical methods to solve problems of identification of objects of various types. Based on the results of the work, the following conclusions can be done. Using developed method it is impossible to implement even the input layer consisting of 6400 neurons (image 80x80). The structure of the neuron could be parallelized the multiplication and accumulation operations to speed up the work. At the same time the requirements to FPGA resources increase.

Keywords:*Neural Networks, Integrated circuits, FPGA, Method, Xilinx, Simulink.*

1. INTRODUCTION

A promising direction of information data analysis is the use of neural networks – a class of methods based on a biological analogy with the human brain and designed after passing the "training" stage on available data to solve various data analysis problems [1-3].

When using them, the most important and difficult choice is the specific network architecture (the number of "layers" and the number of "neurons" in each of them). The size and structure of the network must correspond to the essence of the phenomenon under study. The constructed network undergoes a process of so-called learning, during which the neurons of the network iteratively process the input data and adjust their weights so that the network best predicts the data on which the "training" is performed. After training on the available data, the network is ready to work and can be used to build forecasts. The neural network obtained as a result of "training" expresses the patterns present in the data, being the functional equivalent of some model of dependencies between variables, similar to those that are built in traditional modeling. The method is focused exclusively on the practical result, and not on the essence of the mechanisms underlying the phenomenon or the correspondence of the results obtained to any existing theory [2, 4-6].

The methods of neural networks can also be used in studies aimed at building an explanatory model of a phenomenon, since neural networks help to study data in order to find significant variables or groups of such variables, and the results obtained can facilitate the process of subsequent model construction. Moreover, there are now neural network programs that can use complex algorithms to find the most important input variables, which directly helps to build a model. There are also neural network programs in which artificial intelligence methods are used to solve the time-consuming task of finding the best network architecture. One of the main advantages of neural networks is that, at least theoretically, they can approximate any continuous function, and therefore the researcher does not need to make any hypotheses about the model in advance and even, in

some cases, about which variables are really important [1, 7-10].

Amount of research in the field of artificial intelligence increases every year. The areas of application of artificial intelligence are quite extensive: automation, analysis of large amount of data, smart home technologies, machine vision, etc.

Artificial intelligence technologies are based on the use of neural networks. At the same time, the actual issue is the implementation of neural networks on various software and hardware platforms: programmable logic integrated circuits of the FPGA type (Field Programmable Gate Array), special-purpose integrated circuits (Application-Specific Integrated Circuit, ASIC), GPU, CPU, etc. FPGAs work best in low-power mobile systems. ASICs have the highest performance, but they have a downside: the high development cost [11-14].

The usual methods of prototyping projects based on neural networks implemented on FPGAs are the use of HDL languages, HDL encoders, and graphical programming. Problems of such prototyping projects are that: either such project is complex and time-consuming to debug (HDL languages), or the resulting code is not optimal (HDL coders), or it takes a lot of time to develop a project and the neural network is difficult to reconfigure (graphical programming). To eliminate this problem, the method described in this paper is proposed [15-18].

In this paper, we consider an effective method to design fully connected neural networks or convolutional neural networks and to implement these networks on FPGAs using the Xilinx System Generator for DSP and Matlab/Simulink package. Neural networks designed according developed method are easily reconfigurable and allow solving the following tasks: image recognition, optimal filtering.

The method consists in designing convolutional neural networks using the Xilinx System Generator for DSP library, imported into the Matlab/Simulink environment, and using the built-in MatlabC language to automatically generate a given neural network structure. The artificial neural networks generated by this method are easily reconfigurable and allow solving the following tasks: image recognition, optimal filtering. It is obvious that the use of artificial intelligence methods based on neural networks will, among other things, bring to a qualitatively new level the procedures for recognizing technical states when monitoring and diagnosing complex objects, as well as become a practical alternative to strict mathematical methods for solving problems of identifying objects of various types. Artificial neural networks are a paradigm of learning and automatic processing based on the principles of the animal nervous system. Such neural networks are a combination of various systems, called neurons, which are combined with each other, creating excitation at the output.

2. ANALYSIS OF THE CURRENT SITUATION

Recently, interest in the study and application of neural networks has increased significantly. In this paper, we consider the design of neural networks implemented on FPGAs. FPGA is a platform that are the most suitable for energy-efficient systems. Basically, fully connected neural networks or convolutional neural networks are used to implement on FPGA [19-21].

Problems of such prototyping projects are that: either such project is complex and time-consuming to debug (HDL languages), or the resulting code is not optimal (HDL coders). Design of fully connected and convolutional neural networks using the Xilinx System Generator for DSP is described in [1]. But design of fully connected neural networks with a large number of neurons is quite time-consuming.

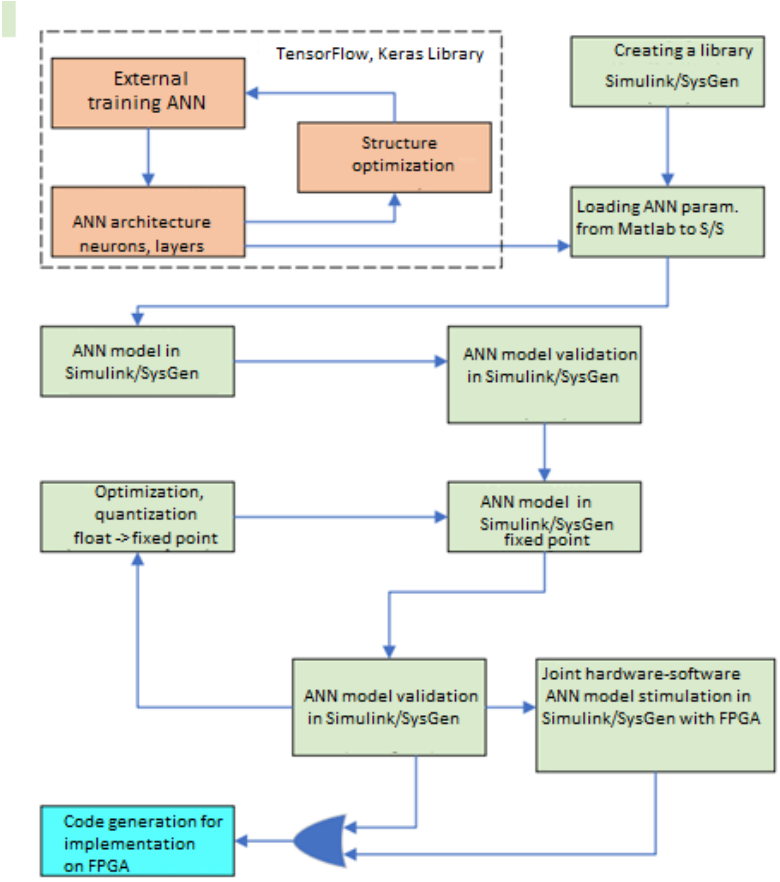
Feedforward neural networks implemented on FPGA using the Simulink / Xilinx System Generator for DSP is described in [2]. The increasing popularity of impulse neural networks implemented on FPGA are also presented in [3]. Thus, the approaches of implementation of neural networks of various types (direct propagation, convolutional, pulse) on FPGAs were considered in [3-6]. Literature review shows that the most developed technology is the implementation of feedforward neural networks, convolutional neural networks are the most difficult to implement on FPGA because such types of neural networks are the closest to biological neural networks. Neural networks training is usually done using Matlab (Deep Learning Toolbox) or other tools such as TensorFlow, Shogun, etc. The activation function is regular sigmoid or logical function. In this case, it is necessary to use different approximations of the selected function using the CORDIC, LUT structures built into FPGA. Modeling neural networks functional schemas in Xilinx System Generator makes it much easier and faster to create a project in ISE or Vivado. Often the technology of automatic generation of HDL code and Matlab/Simulink are used, direct writing of HDL

code in hardware development environments (Vivado, Quartus) is rarely used. Result of analysis of the implementation of neural networks on FPGAs shows that the most reasonable choise is design of fully connected neural networks or convolutional neural networks. These networks is implemented on the target platform using of Matlab/Simulink and the Xilinx System Generator library.

3. NEURAL NETWORKS DESIGN FLOW IMPLEMENTED ON FPGA

In this paper a method solving the problem mentioned above is proposed. The method is design of neural networks using the Xilinx System Generator for DSP library imported into the Matlab/Simulink environment and using the built-in Matlab C language to generate the specified neural networks structure automatically. The neural networks designed according to this method are easily reconfigurable and allow solving the following tasks: image recognition, optimal filtering. This method based on neural networks gives an opportunity to improve recognition procedures of technical conditions of monitor and diagnostics of complex objects. Also such methods could be a practical alternative of strict mathematical methods [7] to solve problems of identification of objects of various types.

The developed design flow of neural networks using Matlab/Simulink and the Xilinx System Generator library is shown in Figure 1. In this diagram the operations that are performed using the neural networks design, training and testing tools with the appropriate libraries (TensorFlow, Keras) are highlighted in orange, the letters ANN designate a neural network. The operations performed in the Simulink/Xilinx System are highlighted in green. These green operations are necessary to prepare the project to implement the neural networks on the target platform. The Xilinx element library is described in [8]. Creating your own library is necessary to design the neural networks architecture in Simulink/SysGen automatically. A description of work in Simulink and generating a model with commands is presented in [9].



ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In this paper the structure of the convolutional neural network is taken as a basis of automatic generation according to the developed method. The structure of the single-channel convolutional neural network is shown in Figure 2.

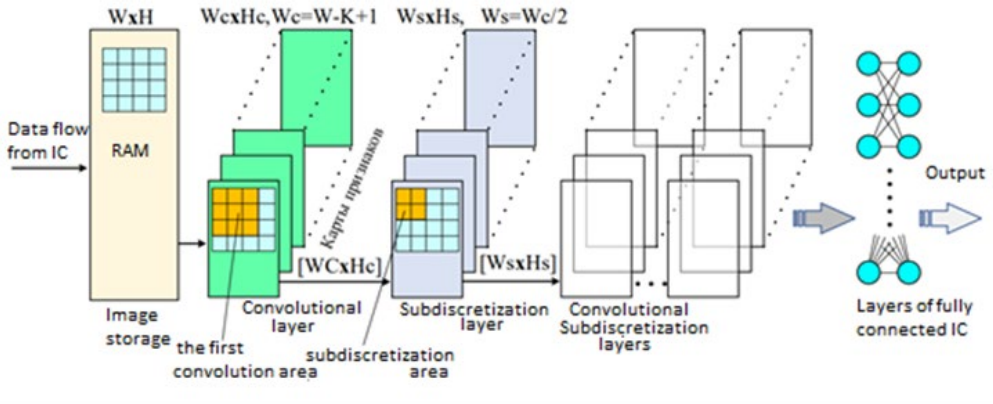


Figure 2: Convolutional Neural Network Structure

In the Figure 2 the "Image storage" block is a set of components (RAM, ROM). In cells of RAM images are stored. These images are fed to the input in a streamed form. The "Image storage" block generated with blocks Simulink/SysGen is presented in the Figure 3.

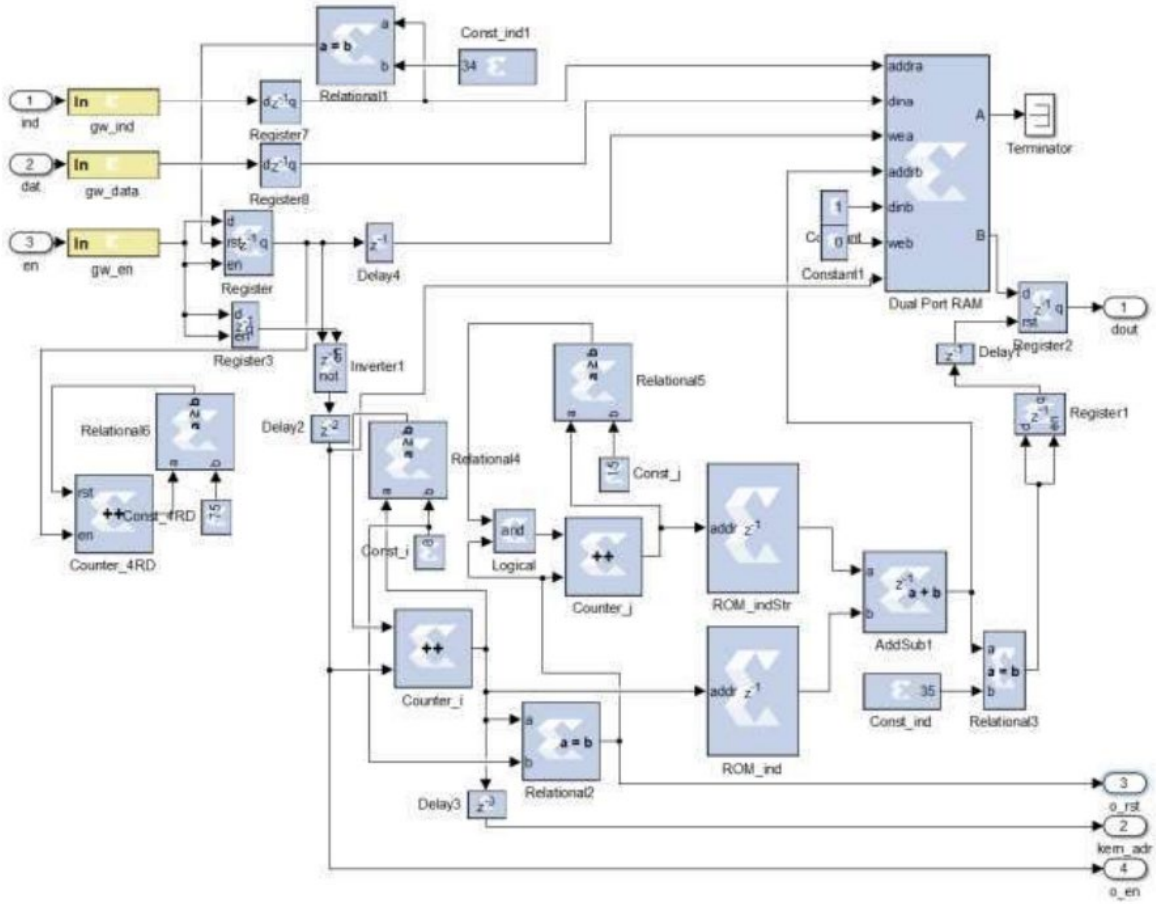


Figure 3: "Image Storage" Block

The tasks of the convolutional layer is to formate a two-dimensional convolution between the input image and the core of the convolution filter. The subsampling layer reduces the image by 2 times (horizontally and vertically). After the two-dimensional convolution and subsampling operations the signal from all streams, associated with the number of feature maps, is multiplexed into a single stream and fed to the fully connected part of the neural network.

Automatic generation of the neural network structure involves writing scripts in Matlab for each individual Simulink/SysGen block (layers of convolution, subsampling, PNS neurons), as well as connecting them to each other in a single structure.

For the convenience of working with the neural network in the Simulink/Xilinx System it is necessary to create a library of those elements that is used in the developed neural network. The basis of the neuron (Neuronxo) is shown in Figure 4.

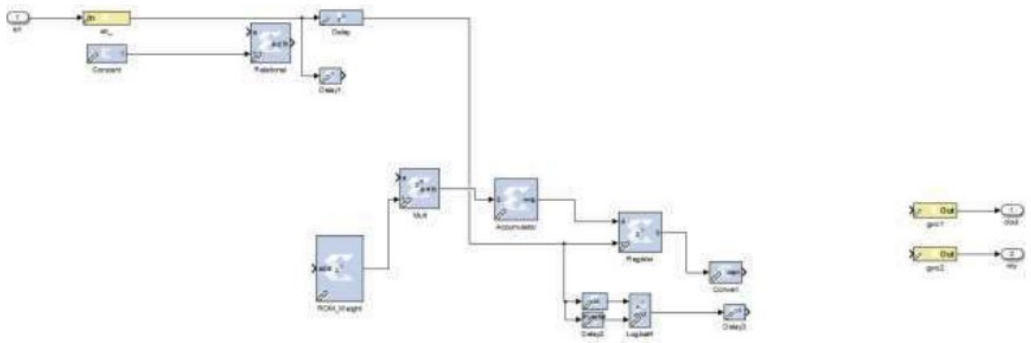


Figure 4: The Basis of a Fully Connected Neural Network Neuron

The basis of a fully connected neural network neuron for a floating pointneural network is shown in Figure 5. Using such a neuron it is possible to generate both a parallel structure with a multiplexer and a chain structure.

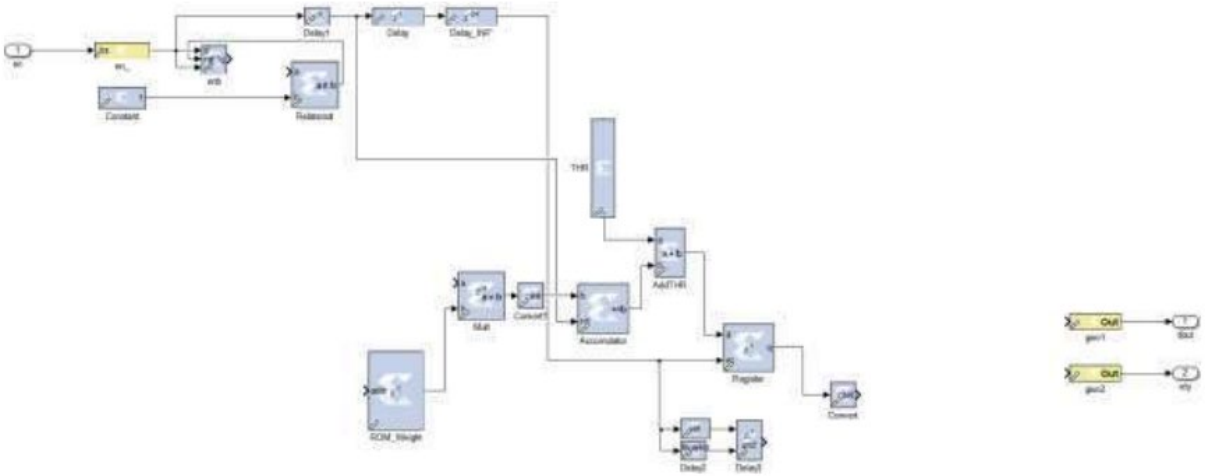


Figure 5: The Basis of a Neuron of a Fully Connected Neural Network

In the process of generating a neural network, a counter, a multiplexer depending on the number of input values, as well as an activation function block are added to the neuron. An activation function block is shown in Figure 6.

Journal of Theoretical and Applied Information Technology
15th October 2021. Vol.99. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

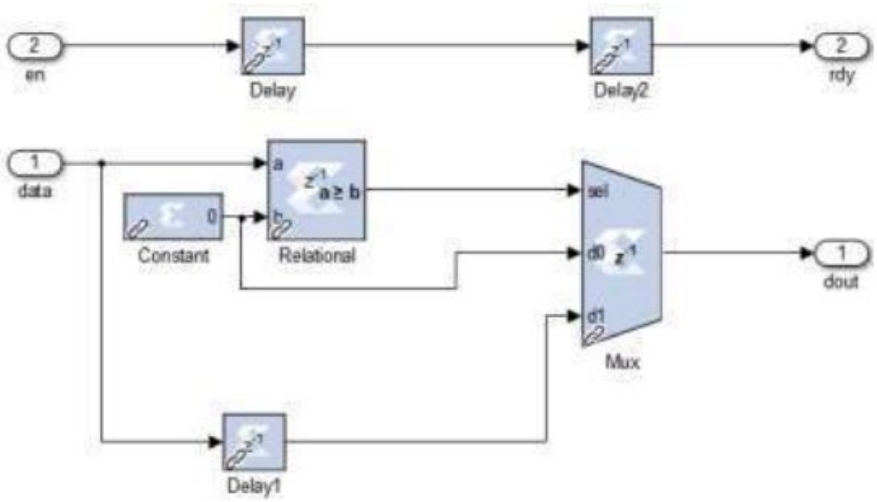


Figure 6: Activation Function

The time diagram of the activation function in Simulink/SysGen is shown in Figure 7.

An example of an automatically generated artificial neural network containing 3 layers and 15 neurons is shown in Figure 8.

Figure 9 shows a model of a convolutional layer of a single map (6x6 in 4x4) with a 3x3 core in Simulink/System Generator.

Figure 10 shows the result of modeling the operation of the model shown in Figure 9.

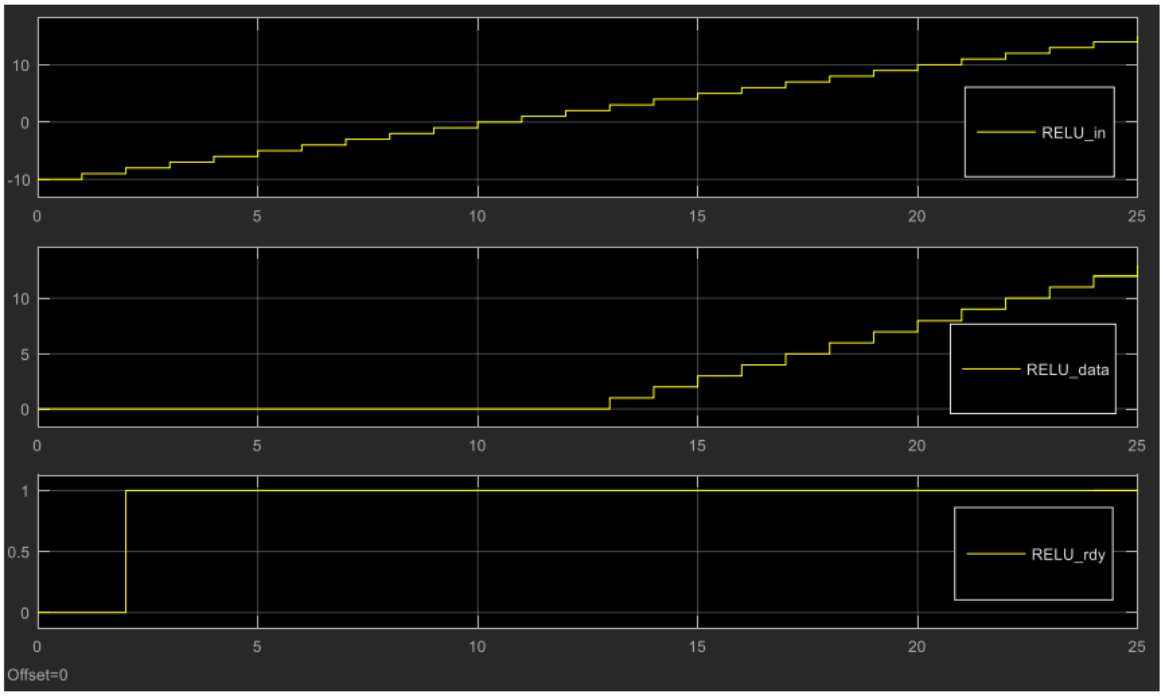


Figure 7: The Result of the Activation Function

Journal of Theoretical and Applied Information Technology
15th October 2021. Vol.99. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

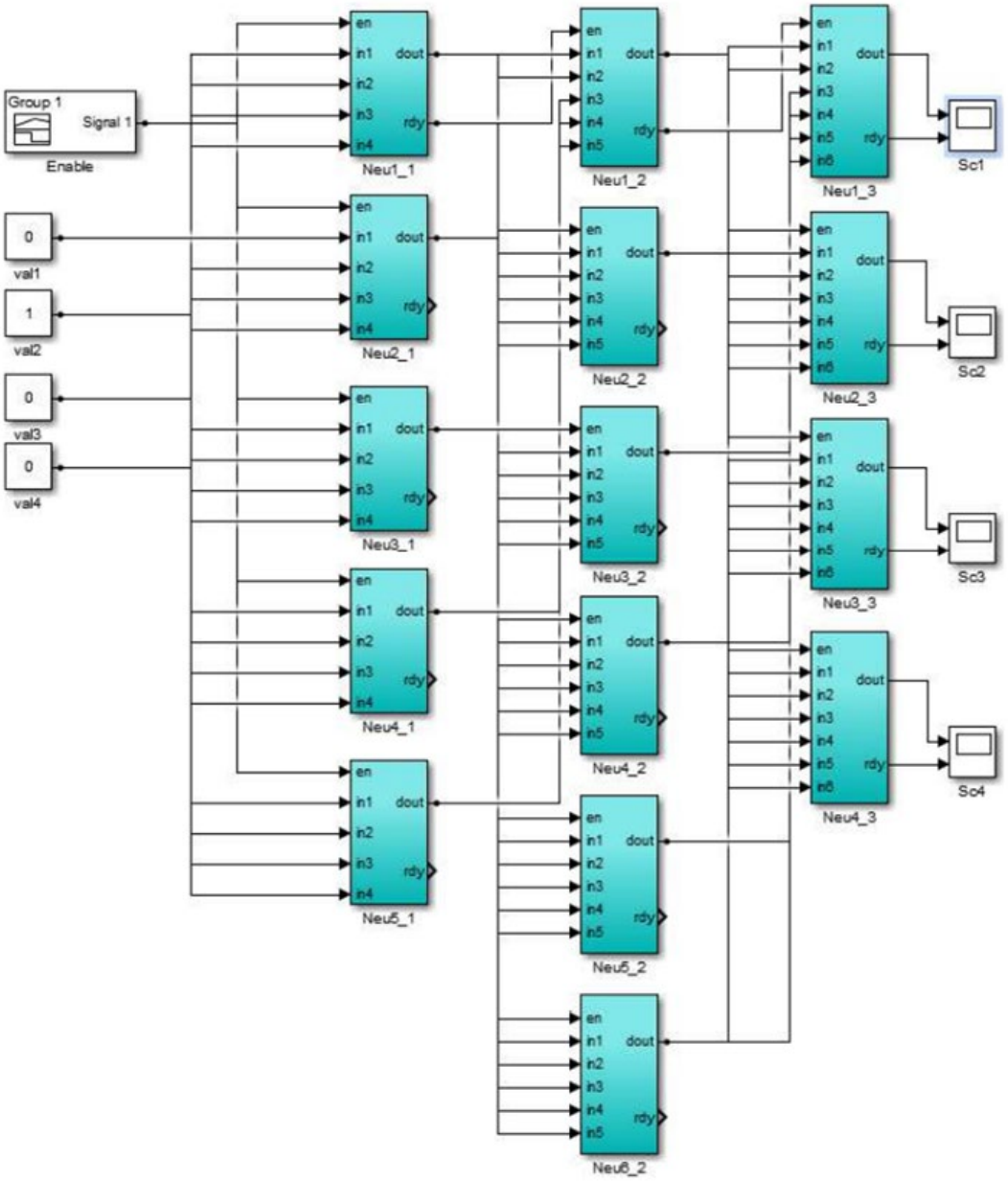


Figure 8: Automatically Generated Artificial Neural Network

Journal of Theoretical and Applied Information Technology
15th October 2021. Vol.99. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

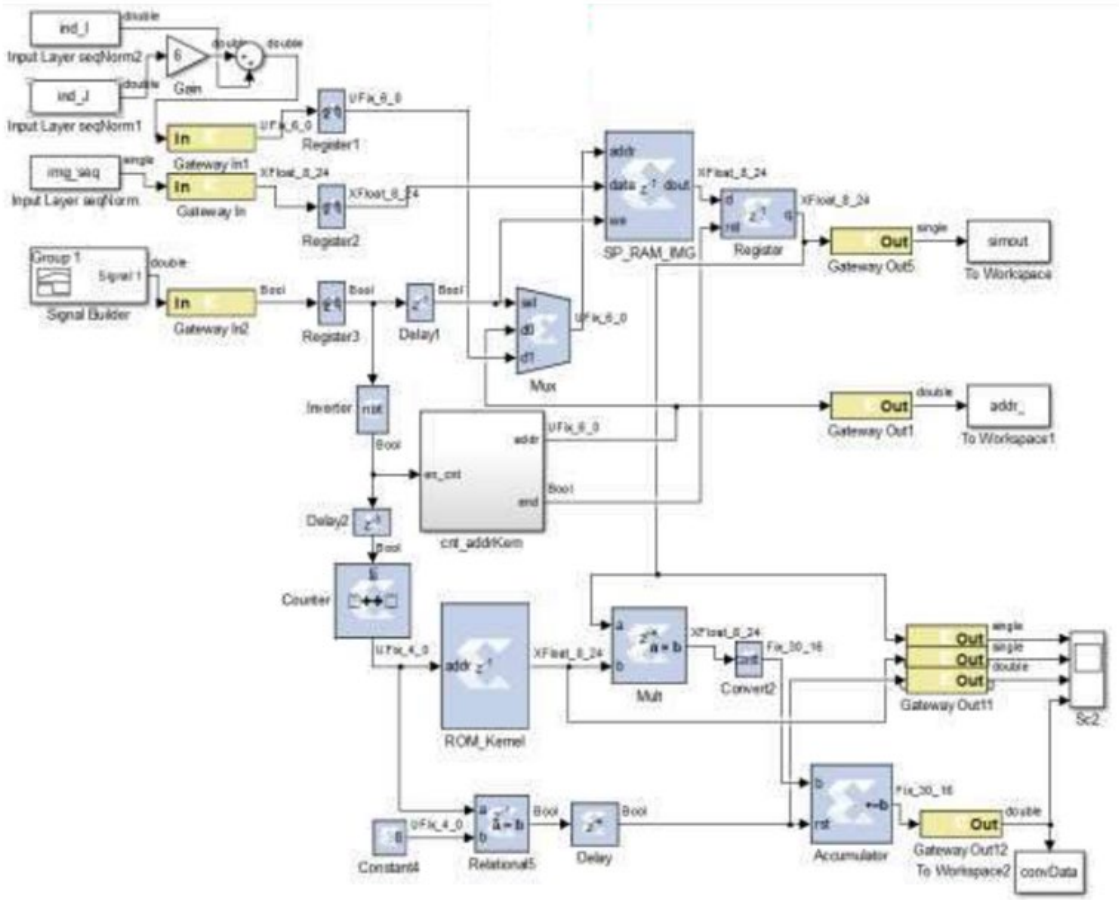


Figure 9: Convolutional Layer Model

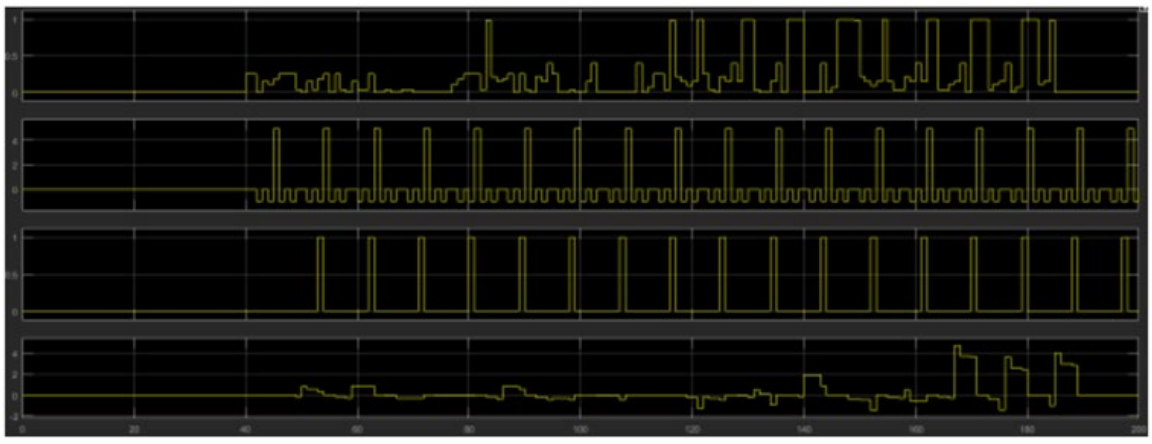


Figure 10: Modeling the Operation of a Convolutional Layer

4. RESULTS OF NEURAL NETWORK IMPLEMENTATION ON FPGA

To estimate the resource consumption of FPGA (KintexUltraScale xcku040-2 - effva1156) a neural network consisting of three layers (1 layer – 32 neurons, 2 layer – 32 neurons, 3 output layer – 2 neurons) is used. It should be noted that this

analysis is performed for components that work with floating point (float).

Table 1 shows total resource consumption of FPGA.

Table 1: Total Resource Consumption of FPGA with Neural Network

Resources	Used	Available	Usage (%)
Numberofregistercells	11924	484800	2.46
NumberLUTs	43986	242400	18.14
NumberBlock RAM	33	600	5.5
NumberDSPsBlock	72	1920	3.75

Resource consumption of neurons of different layers is shown in Table 2.

Table 2: Resource Consumption of Neurons of Different Layers

Resources	Layers	Used	Available	Usage (%)
Numberofregistercells	1	184	484800	0,038
	2	152		0,031
	3	586		0,12
LUTsNumber	1	831	242400	0,343
	2	408		0,168
	3	2169		0,895
Block RAM Number	1	0.5	600	0,833
	2	0.5		0,833
	3	0.5		0,833
NumberDSPsBlock	1	1	1920	0,052
	2	1		0,052
	3	4		0,21

The resource consumption of FPGA are estimated with the System Generator for DSP in Post Implementation mode with standard synthesis and implementation strategies (for the purity of the assessment).

Analyzing the data presented above, we can conclude that with the help of such structure, it is impossible to implement even the input layer consisting of 6400 neurons (image 80x80). The structure of the neuron could be parallelized the multiplication and accumulation operations to speed up the work. At the same time, however, the requirements to FPGA resources increase. Thus, it is necessary to find a compromise between resource consumption and speed.

5. DISCUSSION

Automatic generation of the neural network structure involves writing scripts in Matlab for each individual Simulink/SysGen block (layers of convolution, subsampling, PNS neurons), as well as connecting them to each other in a single structure. So it is possible to set almost any neural network structure and generate a model in Simulink in accordance with this structure [1-8]. Result of analysis of the implementation of neural networks on FPGAs shows that the most reasonable choise is design of fully connected neural networks or convolutional neural networks. These networks is implemented on the target platform using of Matlab/Simulink and the Xilinx System Generator library.

6. CONCLUSION

The current state of artificial intelligence technologies is characterized by the combination of methods and approaches developed within the framework of unrelated research. According to international experts, there are good opportunities for the development of artificial intelligence technologies in Russia today. The breakthrough in this area was made possible due to the development of modern information technologies and the development of new computing tools. Artificial intelligence technologies are based on the use of neural networks. For the development of neural networks, the possibility of neural networks working on various software and hardware platforms is an urgent issue. The development of neural network design methods is an urgent task, as it contributes to the further development of all technologies related to artificial intelligence. In this paper, there is an effective method to design fully connected neural networks or convolutional neural networks and to implement these networks on FPGAs using the Xilinx System Generator for DSP and Matlab/Simulink package. So, the developed design flow of neural networks using Matlab/Simulink and the Xilinx System Generator library is suitable for neural network design in

projects of radar signal processing in radio engineering systems, as well as in simulation modeling projects. Writing or reading an image in RAM should be represented in a quadratic form when implementing neural networks according to this method. The developed method is used to design neural networks recognizing linear signals (one-dimensional), then the indexes to write or read the signal in RAM have a simpler appearance. The other modules of the developed library are not change the appearance because of the versatility of the modules. The developed methods contribute to the creation of conditions for improving the efficiency and the formation of fundamentally new areas of activity in the field of artificial intelligence and neural networks.

7. ACKNOWLEDGEMENTS

The research was carried out under the project "Creation of high-tech production of hardware and software systems for processing agricultural raw materials based on microwave radiation" (Agreement with the Ministry of Education and Science of the Russian Federation No. 075-11-2019-083 of 20.12.2019, SFedU Agreement No. 18 of 20.09.2019, number of work in SFedUNo. XD/19-25-RT).

REFRENCES:

[1] J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, T. Tang, N. Xu, S. Song, Y. Wang, and H. Yang,"Going Deeper with Embedded FPGA Platform for Convolutional Neural Network", in*Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '16*, Monterey, California, USA, February 21-23, 2016, pp. 26-35.https://doi.org/10.1145/2847263.2847265

[2] S.P. Shipitsin, and M.I.Yamaev,"Development of hardware-oriented neural networks on FPGA and ASIC",*Bulletin of the Perm National Research Polytechnic University. Electrical engineering, information technology, control systems*, No. 31, 2019, pp. 177-192.

[3] M. Alçınİ.Pehlivan, and İ. Koyuncu,"Hardware design and implementation of a novel ANN-based chaotic generator in FPGA",*Optik*,Vol. 127,No. 13, 2016, pp. 5500-5505.

[4] A.N. Vorob'ev,D.Yu.Shidlovsky, and A.A.Bagrov,"Hardware implementation of convolutional neural network on FPGA using model-oriented design",in *Digital signal processing and its application–DSPA-2019*,Moscow, Russia, March 27-29, 2019, pp. 423-429.

[5] M. Bahoura,"FPGA implementation of an automatic wheezing detection system",*Biomedical Signal Processing and Control*,Vol. 46, 2018,pp. 76-85.

[6] S. Oniga,"A new method for FPGA implementation of artificial neural network used in smart devices",in *International computer science conference microCAD*,Miskolc, Hungary, March 2005, pp. 31-36

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner,"Gradient-based learning applied to document recognition",*Proceedings of the IEEE*,Vol. 86,No. 11, 1998, pp. 2278-2324.

[8] P. Sibi, S.A. Jones, and P.Siddarth,"Analysis of different activation functions using back propagation neural networks",*Journal of Theoretical and Applied Information Technology*,Vol. 47,No. 3, 2013,pp. 1264-1268.

[9] S. Jeon, and K.C. Lee,"Comparison of cephalometric measurements between conventional and automatic cephalometric analysis using convolutional neural network",*Progress in Orthodontics*, Vol. 22, No. 1, 2021, 14. https://doi.org/10.1186/s40510-021-00358-4

[10] Y. Li, R. Chen, B.Sensale-Rodriguez, W. Gao, and C. Yu, "Real-time multi-task diffractive deep neural networks via hardware-software co-design",*Scientific Reports*, Vol. 11,No 1, 2021, 11013. https://doi.org/10.1038/s41598-021-90221-7

[11] Y. Han, and Y. Han, "A deep lightweight convolutional neural network method for real-time small object detection in optical remote sensing images",*Sensing and Imaging*, Vol. 22,No 1, 2021, 24. https://doi.org/10.1007/s11220-021-00348-0

[12] M. Ala'raj, M.F.Abbod, and M. Majdalawieh, "Modelling customers credit card behaviour using bidirectional LSTM neural networks",*Journal of Big Data*, Vol. 8,No. 1, 2021, 69. https://doi.org/10.1186/s40537-021-00461-7

[13] T. Jiang, X.-L. Hu, X.-H. Yao, L.-P. Tu, J.-B. Huang, X.-X. Ma, J. Cui, Q.-F. Wu, and J.-T. Xu,"Tongue image quality assessment based on a deep convolutional neural network",*BMC Medical Informatics and*

Decision Making, Vol. 21, No. 1, 2021, 147. https://doi.org/10.1186/s12911-021-01508-8

[14] Z. Pourtousi, S.Khalijian, A.Ghanizadeh, M.Babanezhad, A.T.Nakhjiri, A. Marjani, and S.Shirazian, "Ability of neural network cells in learning teacher motivation scale and prediction of motivation with fuzzy logic system",*Scientific Reports*, Vol. 11,No. 1, 2021, 9721. https://doi.org/10.1038/s41598-021-89005-w

[15] Y.V. Shanin, A.S. Bondar, F.V.Chmilenko, and Q. Zhang, "Neural network for predicting the thermal conductivity of steel with the Bayesian method using matlabsoftware",in *Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*,Moscow and St. Petersburg, Russia, January 26-29, 2021, pp.1083-1087.

[16] A.D.M. Africa, D.A.P.Abaluna, A.J.A.Abello, and J.M.B. Lalusin, "Implementation of Neural Network Control in a Nonlinear Plant Using MATLAB",in*IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM 2020)*, Manila, Philippines, December 3-7, 2020,pp. 1-6.https://doi.org/10.1109/HNICEM51456.2020.9400007

[17] M. Kopka, W.Paskal, A.M.Paskal, P.Pietruski, R.Kopka, and P.K.Wlodarski, "Automated nerve fibres identification and morphometry analysis with neural network based tool in MATLAB",in *Studies in Health Technology and Informatics*, Vol. 270. IOS Press, 2020, pp.1209-1210. https://doi.org/10.3233/SHTI200366

[18] S.-C. Chen, and H.-K.Hoai, "Implementing a neural network controller for a permanent magnet synchronous motor based on MATLAB function",in*IEEE International Conference on Consumer Electronics (ICCE-TW 2019)*, Yilan, Taiwan, May 20-22, 2019,pp. 1-2. https://doi.org/10.1109/ICCE-TW46550.2019.8991724

[19] Matworks. Help Center,*Create a Custom Library*. [Online].Available:https://uk.mathworks.com/help/simulink/ug/creating-block-libraries.html(access date: April 20, 2020).

[20] D.E. Khodja, A.Kheldoun, and L.Refoufi,"Sigmoid function approximation for ANN implementation in FPGA devices", in *Proceedings of the 9thWSEASInternationalConference on Circuits, Systems, Electronics, Control and SignalProcessing*, Athens, Greece, December 2010, pp. 112–116.

[21] V.V. Bakhchevnikov,"Electrodynamic model of a radio signal scattered on a multilayer structure using physical optics and the ray tracing method",*News of higher educational institutions of Russia. Radio electronics*,Vol. 22,No. 6, 2019, pp. 25-36.