https://doi.org/10.52326/ic-ecco.2022/CE.06

OPEN ACCESS CC BY NC ND

Design of Specialized Hardware Architectures for Industry 4.0

Silvia Munteanu [1], ORCID: 0000-0003-0749-8457
Viorel Cărbune [1], ORCID: 0000-0002-1556-4453
[1] Technical University of Moldova, Chișinău, Republic of Moldova, silvia.munteanu@calc.utm.md, viorel.carbune@calc.utm.md, http://www.utm.md

***Abstract*— In the process of transition to Industry 4.0, the importance of applying cutting-edge technologies such as machine learning and artificial intelligence to replace human operators in industrial processes is explained by the need to automate industrial production processes. Replacing qualified human experts with artificial neural networks opens up a lot of possibilities for the implementation of new methods of industrial process automation. The problem of industrial process automation is quite complex because the decision-making process of the human expert is accompanied by uncertainty.**

Artificial neural networks represent one of the basic branches of artificial intelligence. At the moment, they are used in various fields to solve problems for which classical methods are unable to provide practical solutions. Thus, the problem of developing and training artificial neural networks for solving industrial process automation problems acquires major importance in the design of artificial intelligence systems. The training process directly depends on the data set on the basis of which the neural network is designed.

***Keywords*—Industry 4.0; machine learning; artificial intelligence; industrial processes; artificial neural networks; FPGA.**

I. General Aspects of USING Neural Networks in Industrial Processes

The article presents a method of hardware implementation of artificial intelligence systems, the key elements for the application of these systems in the automation of industrial processes and the technologies used to implement intelligent solutions in the automation of industrial production systems.

Starting from the problem of implementing artificial neural networks [5, 6, 8, 9, 13] for the automation of industrial processes, it was proposed to implement artificial neural structures based on programmable circuits [4, 7, 10-12]. The problem of automating industrial processes is quite complex, and various alternative methods are used to solve it, including artificial intelligence. In order to solve this problem, it is proposed to develop the architectural support for the implementation of artificial neural networks. Emerging from the computation behavior of artificial neural networks, there is the need to ensure their hardware support by designing specialized architectures.

For assisted training of artificial neural networks, it is necessary to have a consistent set of data collected from the human expert and to choose the appropriate learning algorithm. The use of artificial intelligence in industrial processes offers enormous application potential for command and control of industrial processes.

II. Design of Artificial Neuron Architecture

The need to develop neural hardware architectures is driven by the actual requirements in industry. Artificial neural architectures have been developed and implemented with orientation to the requirements stipulated in the standards for the automation of industrial processes in Industry 4.0 [1-3].

Based on the specifics of the neural network implementation problem, programmable circuits were selected as the most suitable architectural support. Thus, the use of these architectures offers the possibility of parallelization and as a result the optimization and efficiency of neural calculations through the direct implementation of vector operations in hardware.

For the implementation of neural models, specialized hardware architectures are needed that would be able to perform the computational process. The hardware implementation of neural models requires the implementation of the architecture of the neuron based on programmable circuits. Since the basis of any neural network is the classical model of the neuron, the task is reduced to implementing its model in hardware [5, 13]. In order to simplify the implementation process of the artificial neuron, the modular design of its internal architectures was realized [4, 7]:

- *inputs;*
- *synaptic weights;*
- *adder;*
- *activation function;*
- *outputs.*

Thus, the operating principle of the artificial neuron can be described by summing the weighted inputs and generating at the output the activation function applied to the obtained sum. The architectural model of the neuron must have a sufficient number of inputs that can be determined based on the conditions of the problem for the solution of which its model will be involved. in general, each neuron has *n* external inputs. The artificial neuron model was divided into five functionally different components: *Inpts*, *Wghts*, *Snpss*, *Smtr*, *Trnsfr* and *Otpt*, Figure 1, where:

Inpts – inputs;
Wghts – weights;
Snpss – synapses;
Smtr – sumator;
Trnsfr – activation function;
Otpt – outputs.

The modular structure of the artificial neuron model will provide increased flexibility in the process of developing neural architectures, giving developers the opportunity to implement their own specialized architectures to solve certain problems. Thus, in the development process, engineers can select, for example, the activation function of each neuron.

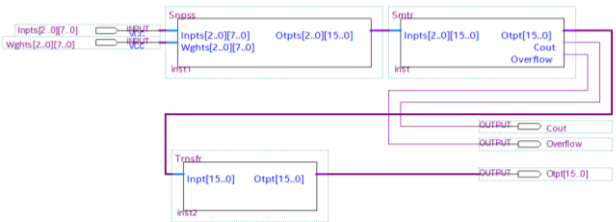

Figure 1. Structure of the digital neuron.

The synaptic block – *Snpss* (Figure 2), performs the multiplication of each input with its corresponding weight, Figure 1. The synaptic component in the structure of the neuron is elaborated using the multipliers that calculate the products between the respective pairs in parallel. The input vector of the RNA and the vector of weights are applied to the inputs of the synaptic component, and the output signal represents the vector of products obtained by multiplying each input with the respective weight.

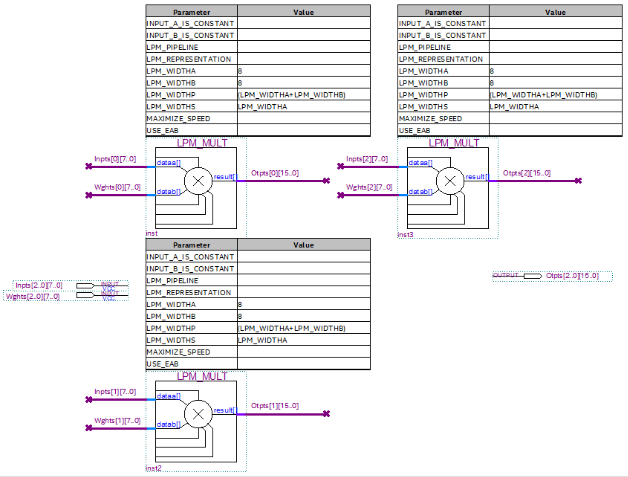

Figure 2. Structure of the synapses module.

The summation block – *Smtr* (Figure 3), performs the summation of all inputs. The summing component in the neuron structure is realized on binary adders. The vector of synaptic products is applied to the inputs of this component and the output is a scalar.

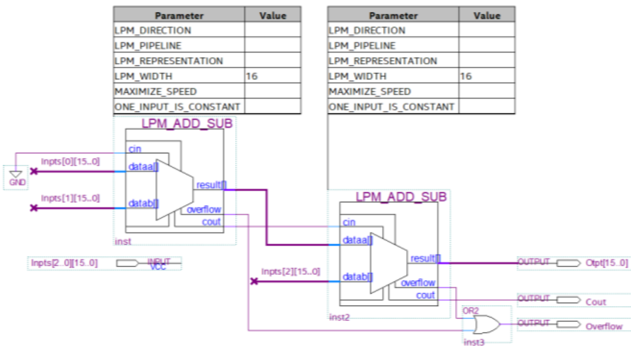

Figure 3. Structure of the summation module.

The block for implementing the activation function – *Trnsfr*, calculates the value of the activation function depending on the value of the total sum of the synaptic products applied to the input of this block. To implement different activation functions, several blocks have been developed for a series of classic functions such as: *step* (Figure 4), *linear*, etc.

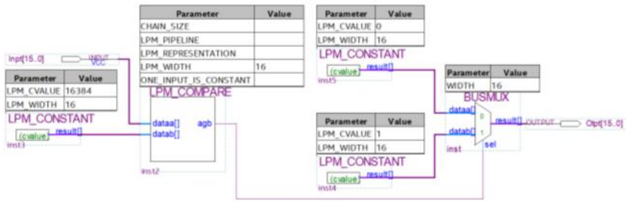

Figure 4. Structure of the step activatation function.

As a result of the synthesis of the structure of the neuron, the architecture presented in Figure 5 was obtained, in which the conceptual similarity with the model of the artificial neuron presented in Figure 1 can be noted.

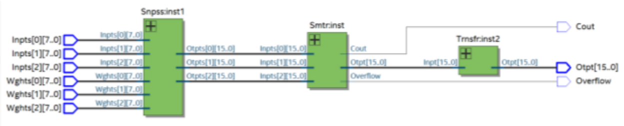

Figure 5. Architecture of the step activitation function.

Using the neuron model at the design stage of artificial neural networks significantly simplifies the process of their implementation in hardware. For the implementation of neural models in the automation of industrial processes, methods and tools for the synthesis of artificial neural networks based on programmable circuits were developed and implemented. The use of programmable circuits for the implementation of artificial neural networks presents remarkable engineering advantages because they essentially simplify the design stage and offer possibilities for the optimization of neural computations. An essential advantage of this architectural implementation method is the possibility of designing an entire system on a single circuit. The main advantage of implementing neural networks based on reconfigurable circuits can be fully exploited only in the case of using programmable circuits, a fact that offers the possibility of parallelizing the calculation processes carried out in the structure of each artificial neuron [5, 13].

III. Conclusion

Following the analysis, it was possible to identify the individual characteristics of the described method of architectural design of artificial neural networks. The essential advantages of hardware implementation of neural architectures have been determined. In the case of the implementation of intelligent systems for the automation of industrial processes based on neural architectures, the possibility of specialization of these architectures for the optimization through spatial parallelization of neural calculations was highlighted.

In the result, the specific computational features of the described architectures were identified. These achievements offer the possibility of implementing neural architectures based on programmable circuits based on the specifics and needs of the targeted problem.

Acknowledgment

The research carried out in this work is part of the activity of the Department of Informatics and Systems Engineering. This work was supported by the project *20.80009.5007.26 „Models, algorithms and technologies for the control, optimization and security of the Cyber-Physical systems*”.

References

[1] B. Paiva Santos, F. Charrua-Santos, T.M. Lima, ”Indusrty 4.0: An Overview,” *In: Proceedings of the World Congress on Engineering 2018, Vol. II,* July 4-6 2018, U.K., ISSN: 2078-0958.

[2] M. Capestro, S. Kinkel, ”Knowledge Management and Industry 4.0,” *Knowledge Management and Organizational Learning, Vol. 9, Springer Nature Switzerland AG 2020*, pp. 19-52, DOI: https://doi.org/10.1007/978-3-030-43589-9_2.

[3] M. Piccarozzi, B. Aquilani, C. Gatti, ”Industry 4.0 in Management Studies: A Systematic Literature Review,” *Sustainability 2018, 10, 3821,* 24p., DOI: 10.3390/su10103821.

[4] H. N. Teodorescu, A. Kandel, “Hardware Implementation of Intelligent Systems,” *Physica-Verlag Heidelberg,* 2001. 282 pp. ISBN: 978-3-7908-2491-9.

[5] C. Aggarwal, “Neural Networks and Deep Learning,” *Springer International Publishing,* 2018, 497 p. ISBN: 978-3-319-94462-3.

[6] D. Nauck, F. Klawonn, R. Kruse, ”Foundations of Neuro-Fuzzy Systems,” *New Zork: John Wiley&Sons,* 1997. 316 p. ISBN: 978-0-471-97151-1.

[7] N. Chokshi, D. Mcfarlane, ”A distributed architecture for reconfigurable control of continuous process operations,” *In: Journal of Intelligent Manufacturing. 2008, vol. 19,* pp. 215-232. eISSN: 1572-8145.

[8] N. J. Nilsson,”Artificial intelligence: A new synthesis,” *San Francisco: Morgan Kaufmann,* 1998. 513 p. ISBN 978-1-55860-467-4.

[9] S. Russell, P. Norvig, ”Artificial Intelligence: A modern approach,” *2nd ed. New Jersey: Prentice Hall,* 2003. 1132 p. ISBN: 0137903952.

[10] J. Hennessy, D. Patterson, ”Computer Architecture: A Quantitative Approach,” *5th ed. San Francisco: Morgan Kaufmann,* 2012. 857 p. ISBN: 978-0-12-383872-8.

[11] K. Hwang, N. Jotwani, ”Advanced computer architecture: Parallelism, Scalability, Programmability,” *2nd ed. New Delhi: McGraw-Hill Education,* 2010. 810 p. ISBN: 978-0070702103.

[12] J. Staunstrup, W. Wolf, eds. ”Hardware/Software Co-Design: Principles and Practice,” *Boston: Springer,* 1997. 396 p. ISBN: 978-1-4419-5018-5.

[13] B. Kosko, ”Neural Networks and Fuzzy Systems: A Dynamical Sistems Approach to Machine Intelligence,” *Englewood Cliffs: Prentice Hall,* 1992. 449 p. ISBN: 0-13-612334.