The analysis of reachability of inhibitory Petri nets, in its turn, is reduced to solving a system of equations (2) with the incidence matrix A^I:

$$M_{max} = A^I \cdot S_{max} . \quad (4)$$

If each vector $\Delta\mu \in \Delta M$ is fully defined on the set of input and output positions of the network, the analysis of reachability of stable states of automaton is performed (verification of the structural diagram of automaton). The system of equations (4) can have only one solution for each vector $\Delta\mu$.

The generation of reachable steady states of CDA is performed in the case, if each vector $\Delta\mu$ is undefined totally on a set of input and output positions of the network. In case of an uncertainty or incomplete definition of the vector $\Delta\mu$, the system of equations (4) has a set of solutions for each vector $\Delta\mu \in \Delta M$. The entire set of solutions can be obtained even in the case of complete uncertainty of the set ΔM. The number of solutions corresponds to the number of possible switchings of the automaton or the number of sets of the truth table (switching table).

The problem is that the well-known methods for generating solutions of linear systems of equations in nonnegative integer numbers have asymptotically exponential computational complexity, which makes it difficult to use them to analyze real systems. The time of generation of the minimum generating set of solutions (MGSS) on the set of unexpressed variables is critical from the point of view of efficiency. The solution to the problem can be obtained as a result of taking into account the specifics of CDA logic circuits. The MGSS generation of the CDA state equation is performed proceeding from the principle of component activity (composite transitions of the network model) for each state of the automaton. In the network model of each component, only one simple transition can be activated at a time (a set from the component truth table). Accordingly, the number of units in the combination is equal to the number of active transitions of the circuit components. This constraint is determined by the specificity of the CDA firing and is necessary to minimize the iteration of transition combinations, as well as to eliminate possible invalid solutions. Practically, the set of solutions obtained in the process of training for CDA for the minimum number of training sets can be used as the MGSS.

V. Model testing

The mathematical model of CDA was tested on an n-bit parallel adder. In the course of the training prcedure, the mathematical model of CDA was formed which made it possible to reproduce the entire set of the truth table of the n-bit parallel adder. With an increase in the bit depth of the adder (which leads to an exponential growth of the total training set), almost a linear dependence of the number of elements in the minimum training sets was achieved. This results in an acceptable time of learning. The maximum bit depth of the adder is only limited by the memory capacity and the speed of the computer.

VI. Conclusion

The development of the ideas underlying the concept of a cognitive digital automaton has been proposed. The construction of the mathematical model has been described together with the logic synthesis and solution generation procedures.

The proposed approach has a number of benefits compared to conventional feed-forward neural networks. First, the CDA demonstrates the capability of being trained on incomplete training sets, which has been confirmed by the example of arithmetic and logical operations. It has been shown that the minimum number of rows of the truth table required for training grows linearly with the bit depth while the total size of the training set increases exponentially.

Second, the consequence of the proposed training algorithm is an “on the fly” learning mode of the network that could be useful in various applications, e.g. robotics.

Third, the results obtained could help outline an alternative solution to the problem of catastrophic forgetting known for feed-forward neural networks [8]. In the case of CDA, knowledge fusion is a simple combination of separate matrices which form a non-uniform inhibitory Petri net.

Acknowledgment

The work was supported by the Russian Fund for Basic Research and the Government of the Ulyanovsk region under Grants No. 18-47-732015 and No. 19-47-730016.

References

[1] V. V. Kozhevnikov, “Fundamentals of Mathematical Modeling of Cognitive Digital Automata”, J. Num. Anal., Industr. and Appl. Math., vol. 13, No. 1-2, pp.15-27, 2019.

[2] V. V. Kozhevnikov, “The method of mathematical modeling of the logic circuits of digital automata”, Automation of Control Processes, No. 1(30), pp. 97–101, 2012 (in Russian).

[3] V. V. Kozhevnikov, “Reachability analysis method for the stable states of the logic circuits of digital automata”, Automation of Control Processes, No. 1(35), pp. 99–108, 2014 (in Russian).

[4] T. Murata, “Petri nets. Properties, analysis and applications”, Proc. IEEE, Vol. 77, Issue 4, pp. 541–580, 1989.

[5] J. L. Peterson, Petri Net Theory and The Modeling of Systems. Prentice-Hall, 1981.

[6] V. V. Kozhevnikov, “Reachability analysis method for inhibitory Petri nets”, Automation of Control Processes, No. 3(33), pp. 29–34, 2013 (in Russian).

[7] V. V. Kozhevnikov, “The concept of Petri Nets with programmable logic”, Scientific Notes of Ulyanovsk State University: Math. and Inf. Tech. Series, Vol. 1, No 6, pp. 150-155, 2014 (in Russian).

[8] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, et al., “Overcoming catastrophic forgetting in neural nets”, Proc. Nat. Acad. Sci., Vol. 114 (13), pp. 3521-3526, Mar 2017.