INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTERS IN SIMULATION Volume 14, 2020
DOI: 10.46300/9102.2020.14.3

Prediction of solutions of arithmetic and logical operations on the basis of the mathematical model of cognitive digital automata

Valeriy Kozhevnikov
S.P. Kapitsa Technological Research Institute
Ulyanovsk State University
Ulyanovsk, Russia

Victor Prikhodko
S.P. Kapitsa Technological Research Institute
Ulyanovsk State University
Ulyanovsk, Russia

***Abstract*—An approach to the problem of solution prediction of arithmetic and logical operations on the basis of the mathematical model of cognitive digital automata (CDA) is proposed. A particular advantage of the proposed approach is that the training procedure can be performed on limited (minimum) training sets. Prediction or generation of solutions is performed on the basis of the mathematical model of CDA which is formed in the course of training. As a testbed for the approach, the modeling of an n-bit parallel adder was implemented. The mathematical model of the adder was formed, which made it possible to reproduce the entire truth table for the n-bit parallel adder. The results obtained could be useful as an alternative solution to a number of problems known for conventional feed-forward neural networks, e.g. on-the-fly learning and catastrophic forgetting.**

Keywords*—cognitive digital automata, prediction, Petri nets, state equation, arithmetic operations, logical operations, mathematical modeling, solution generation, logic synthesis

I. Introduction

Recently, the concept of cognitive digital automata has been developed by one of the authors [1]. The mathematical model of CDA, in fact, represents a further development of the theory of digital automata and, accordingly, is based on the methods of mathematical modeling of conventional digital automata [2-4]. A particular advantage of the proposed mathematical model of CDA is that the training procedure can be performed on limited (minimum) training sets. The cognitivity of the digital automaton is determined by the possibility of generating solutions that have not been provided during the training procedure.

The mathematical model is constructed on the basis of a representation of CDA in the form of the state equation of Petri nets (PNs) from the class of Murata equations (matrix equations) [4] or a system of linear algebraic equations (SLAE). The mathematical apparatus of PNs [5] has been proposed as a tool for constructing the mathematical model of CDA: marked graphs, inhibitory PNs [6], and PNs with programmable logic (PNPL) [7]. Unlike in inhibitory PNs, the firing logic of the transitions of the PNPL is not pre-set, and any input arc of the transition can be inhibitory, and the transition can be programmed to perform any logical function. In this case, the logic of the components of the initial structure of an automaton in the incidence matrix is given implicitly.

The formation of the configuration of the initial structure of an automaton is implemented as a result of the cluster analysis of training sets. Based on the results of the cluster analysis or classification of training sets, one can determine the number of inputs and outputs, the number of layers of initial structure, the number of components in each layer, the structure of connections between components . The regression analysis of the data of training sets is performed in the process of training or synthesis of the logic of the initial structure of automaton.

The initial structure of an automaton is represented as a universal matrix, where the connections between the components are built on the “all with all” principle. Various multi-level configurations of the initial structure are also possible where the “all with all” connections between the components are only created between different levels of the initial structure similar to that in feed-forward neural networks.

The possibility of generating a formula (or network algorithm) of CDA depends on the critical mass of elements of the training set and the training algorithms. Hence, the task of generating the training sets with a minimum number of elements for a given CDA function or experimentally determined function is of particular importance. Prediction or generation of solutions, in its turn, is performed on the basis of the mathematical model of CDA obtained in the course of training.

II. Construction of the Mathematical Model

The initial structure of CDA is represented in the form of a marked graph (structure diagram) by interpreting the inputs and outputs of the structure and structural components by the positions of the marked graph, and the components themselves and the connection lines as composite and simple transitions, respectively. The set of inputs and outputs of the structure diagram is interpreted as a set of input and output positions of the network. The availability of information is interpreted as a token in the network position. At the logical presentation level, a token in a network position is interpreted as a logical unit, and its absence is interpreted as the logical zero. The movement of information is interpreted as the movement of tokens.

As the minimum set of training sets (highlighted in gray), those rows are selected from the truth table which give ‘1’ in only one output of the adder taking account of the transfer to the senior (third) class. Thus, for each output of the adder, the corresponding class of training sets is formed. Further, from the selected sets for each output class, only sets with one ‘1’

DOI: 10.46300/9102.2020.14.3

at the inputs of the adder, with two '1', etc., were selected. According to this principle, the corresponding subclasses of training sets have been formed.

When forming the initial structure of the adder, sets of subclasses constitute the first layer of the structure diagram, where one component is given to each subclass. The set of classes constitutes the second layer of the structure diagram, where one component also corresponds to each class. As a result, the initial structure of the adder consists of two layers. The number of components of the first layer is equal to the number of subclasses. The number of components of the second layer is equal to the number of classes. In the case when a class consists of one subclass, only one component is formed in the first or second layer. The structure of connections between the inputs of the structure diagram and the inputs of the components of the first layer, the outputs of the components of the first layer and the inputs of the components of the second layer is formed according to the principle of "all with all".

The representation of the CDA structure diagram in the form of a marked graph allows one to go from the description of the structure diagram to its mathematical representation in the form of the incidence matrix: $A = A^+ - A^-$. Constructing a complex mathematical model of CDA is done on the basis of the fundamental state equation of Petri nets from the class of Murata equations [2]:

$$\Delta\mu = A \cdot \tau , \quad (1)$$

where $\Delta\mu = \mu - \mu_0$, μ_0 is the initial network marking vector, μ is the finite network marking vector, τ is the network transition coverage vector, which only determines the composition and does not determine the sequence of transition firings. The vector $\Delta\mu$ is defined on a set of network positions P. The vector τ is defined on a set of network transitions T. A set of vectors $\Delta\mu$ forms a set of ΔM, where $\Delta\mu \in \Delta M$. A set of transition coverage vectors τ forms network coverage S, where $\tau \in S$.

The set ΔM specified on the set of input and output positions of the network is interpreted as the initial truth table or the transition table of CDA states. As a set of training sets, either all elements of the truth table can be used, or only the minimum set.

On the set of training sets, the CDA network model can be represented as a system of matrix equations of Petri nets:

$$\Delta M_{min} = A \cdot S_{min} . \quad (2)$$

The marking of the internal positions of the set ΔM is not determined. The composition of the coverage transitions S is also not determined. Only the initial incidence matrix A is completely determined.

III. Logic Synthesis

To solve the CDA logic synthesis problem, the methods of calculating the invariants of the state equation of the marked graph of the automaton structure diagram are used. Marked graph invariants are a powerful tool for studying the structural properties of networks and are solutions of homogeneous systems of equations.

The projection of implicitly defined input logic of composite transitions of components and output logic of simple transitions of connection lines to the initial structure diagram of an automaton is reduced to solving a system of homogeneous equations (2) with an undetermined incidence matrix:

$$\Delta M_{min} = A^{\alpha} \cdot S_{min} , \quad (3)$$

where $A^{\alpha} = A^{+\alpha} - A^{-\alpha}$ is an undetermined incidence matrix for which $a_{ij} = \{ 0,1,-1, \alpha,- \alpha \}$.

Relevant unknowns are introduced in the incidence matrix A in (1) for the input arcs of composite transitions in accordance with the expression: $a_{ij} = -1 \rightarrow a_{ij} = -\alpha$ and for the output arcs of simple transitions in accordance with the expression: $a_{ij} = 1 \rightarrow a_{ij} = \alpha$, where $\alpha = \{0,1\}$.

The corresponding matrix A^{τ} is calculated for each vector τ . The values of the unknowns for inhibitor arcs are defined implicitly and are equal to zero, which provides a solution to the problem of the matrix representation of inhibitory Petri nets. At the same time, for each compound transition that is a part of the vector τ , the corresponding simple transition and its structural connections characteristic only for the given vector τ are determined. As a result of the union of the matrices A^{τ}, the inhibitor incidence matrix is formed: $A^I = \bigcup A^{\tau}$.

Practically, the projection of implicitly defined logic onto the initial structure diagram of the automaton is reduced to zeroing the rows of the original incidence matrix for each vector τ on a set of network positions that are not part of the corresponding coverage R with the subsequent combination of the matrices for each R. Null rows in the incidence matrix A^I are deleted.

The logic of the components (compound transitions) of the initial CDA structure diagram in the synthesis process (training) of each learning step may vary, i.e. the truth tables of network components do not have a fixed size and, accordingly, a fixed logic function. The process of forming the logic of components is limited only by the number of component inputs or the exhaustive search through possible combinations of signals at the inputs of each component. When there is a sufficiently large number of inputs of each component, the process of forming the logic of components and the network as a whole is almost endless.

To develop the formula (or network algorithm) of the original function defined on the basis of training sets, operations of relational algebra and more complex relational calculus over A^{τ} sets can be used. Eventually, the set of generated matrices A^{τ} make up the combined matrix of a non-uniform inhibitory Petri net or a network algorithm (logical circuit model) of CDA. Further, the network algorithm obtained in the form of the incidence matrix A^I is used as the initial information to solve the problems of the reachability analysis and generation of reachable stable states of CDA.

IV. Solution Generation

The mathematical model of CDA makes it possible to reproduce both the set of solutions specified in the training process and the set of solutions that were not set in the training process. The task of generating solutions of the mathematical model of CDA is reduced to solving the problem of reachability of inhibitory Petri nets.

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.