Memristor-based Hardware Neural Networks Modelling Review and Framework Concept

[1] *D.D. Kozhevnikov < ddkozhevnikov@edu.hse.ru >*
[2] *N.V. Krasilich < nadezhda.krasilich@mail.ru>*
[1] *National Research University Higher School of Economics,*
20, Myasnitskaya st., Moscow, 101000, Russia.
[2] *National Research University Higher School of Economics,*
38, Studencheskaya st., Perm, 614070, Russia.

Abstract. This paper is a report of a study in progress that considers development of a framework and environment for modelling hardware memristor-based neural networks. An extensive review of the domain has been performed and partly reported in this work. Fundamental papers on memristors and memristor related technologies have been given attention. Various physical implementations of memristors have mentioned together with several mathematical models of the metal-dioxide memristor group. One of the latter has been given a closer look in the paper by briefly describing model's mechanisms and some of the important observations. The paper also considers a recently proposed architecture of memristor-based neural networks and suggests enhancing it by replacing the utilized memristor model with a more accurate one. Based on this review, a number of development requirements was derived and formally specified. Ontological and functional models of the domain at hand have been proposed to foster understanding of the corresponding field from different points of view. Ontological model is supposed to shed light onto the object-oriented structure of memristor-based neural network, whereas the functional model exposes the underlying behavior of network's components which is described in terms of mathematical equations. Finally, the paper shortly speculates about the development platform for the framework and its prospects.

Keywords: memristor; memristor model; hardware neural network model; memristor-based neural networks.

DOI: 10.15514/ISPRAS-2016-28(2)-16

For citation: Kozhevnikov D.D., Krasilich N.V. Memristor-based Hardware Neural Networks Modelling Review and Framework Concept. *Trudy ISP RAN/Proc. ISP RAS*, vol. 28, issue 2, 2016, pp. 243-258. DOI: 10.15514/ISPRAS-2016-28(2)-16

1. Introduction

Until 1970-s the world has been aware of only three passive elements of electrical circuitry: resistors, capacitors and inductors. The three stated elements coupled with natural relationships provide five connections for four basic notions of electrical circuit theory (voltage, charge, current and flux). Mathematics, however, claims that four things can be mutually interconnected in six different ways. Indeed, the relation between charge and flux was not present. It wasn't until 1971 that the discordance has been formulated and solved. A new element – memristor - has been proposed by Leon Chua in his paper in IEEE Transactions on Circuit Theory completing the mathematical symmetry of circuit theory. It took nearly 40 years for memristor to transform from a purely theoretic concept into feasible implementation. In 2008 a group of scientists from Hewlett-Packard Labs lead by Stan Williams has finally built working memristors [1].

One of the most promising domains of memristor application, seem to be artificial neural networks [2]. These often come in either software or hardware implementations, sometimes in a combination of both. While digital neural networks simulate the data processing mechanism of biological neural networks, hardware ones strive to emulate it. It is worth mentioning that since most of computer architectures conform to the von Neumann architecture, neural network simulation becomes a challenging task because of the paradigm mismatch. Instead of simulating the ways of nature, hardware neural networks try to directly replicate them, creating non-von-Neumann architectures. In comparison with digitally simulated networks, hardware ones can achieve better speed, less power consumption and chip space.

On the other hand, hardware networks often prove to be far less accurate that their software counterparts, due to the nonuniformity of analog components [3]. Another disadvantage of modern hardware neural networks, which they actually share with the software ones, is the volatile storage of synaptic weights. There are ways to achieve the nonvolatile weight storage within hardware networks, but usually such weights are either static (cannot be changed once manufactured), quickly digress (require frequent updating) or are rather hard to program [4]. The emergence of memristor, however, seems to have opened new possibilities in addressing the stated problems. Memristors seem to be a perfect match for synapses, making hardware implementations of neural networks more reliable and greatly increasing productivity of neural computations [5].

Nevertheless, memristors are still scarcely availably and lack industrial-grade production. Being such a new technology, they are often hard and expensive to acquire for experimentation, but a large variety of memristor models has already been produced, making it possible to model memristor-based devices.

Thus, considering the domain of artificial intelligence, a need in profound and correct model of artificial memristor-based feedforward neural network arises. Such model would be of great help in assessing the qualities of modeled system: computation performance, time and energy expenses, material costs, etc. Consequently, the goal of the research is to develop a framework for modelling artificial memristor-based neural networks.

2. Theoretical Memristor

The concept of memristor has been recognized since 1971, when Leon Chua has proposed for the first time in a well-organized and mathematically described way [6].

The 1971 Chua's paper in IEEE Transactions on Circuit Theory is considered to be the pioneer work in the corresponding field of research. Although, the concept of memristor-like devices has been suggested earlier in 1960 by Bernard Widrow, Leon Chua was the first one not only to provide a feasible foundation for memristor's existence, but also to estimate and mathematically describe its' supposed behavior and properties.

Memristor fulfills the mathematical symmetry of relationships between major circuit notions. The relationship created by a memristor, according to Chua, is expressed as follows:

$$v(t) = M(q(t))i(t),$$

where M(q(t)) is the memristance defined as

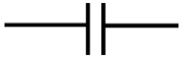
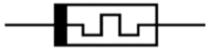
$$M(q) \equiv \frac{d\varphi(q)}{dq}.$$

The definition of memristance may be represented in a more convenient form by substituting flux and charge with their integral definitions:

$$M(q(t)) = \frac{d\varphi/dt}{dq/dt} = \frac{d\left[\int_{-\infty}^{t} v(\tau)d\tau\right]/dt}{d\left[\int_{-\infty}^{t} i(\tau)d\tau\right]/dt} = \frac{v(t)}{i(t)}.$$

The similarity of memristor to the remainder of classical circuit elements can be better reflected by expressing their definitions via differential equations as it is done in fable 1.

Table 1. Differential equations of basic circuit elements

Device	Electronic Symbol	Unit	Differential equation
Resistor		R, ohm	$R = \frac{dv}{di}$
Capacitor		C, farad	$C = \frac{dq}{dv}$
Inductor		L, $\frac{Wb}{A}$ or henry	$L = \frac{d\varphi}{di}$
Memristor		M, $\frac{Wb}{C}$ or ohm	$M = \frac{d\varphi}{dq}$

The first important property of memristors, which commonly is referred to as memristance and stands for the ability to change its resistance gradually via a controlled mechanism (e.g. memory of device's history of charge).

The second significant attribute of memristors, figured out by Chua, is the non-volatility property, which stands for the absence of internal power supply. In other words, Chua proposed that memristor is able to store the value of own resistance without the need to be connected to a power source.

In 1976, Leon Chua and his fellow colleague Sung Kang proceeded exploring the mathematical and physical properties of the memristor [7].

They had come to an understanding, that since memristor is a dynamic device, one equation is not enough to describe it, henceforth memristor's behavior is represented by following equations for current-controlled memristor

$$x = f(x, i, t)$$
$$v = R(x, i, t)i$$

and for voltage-controlled one

$$x = f(x, v, t)$$
$$v = R(x, v, t)i$$

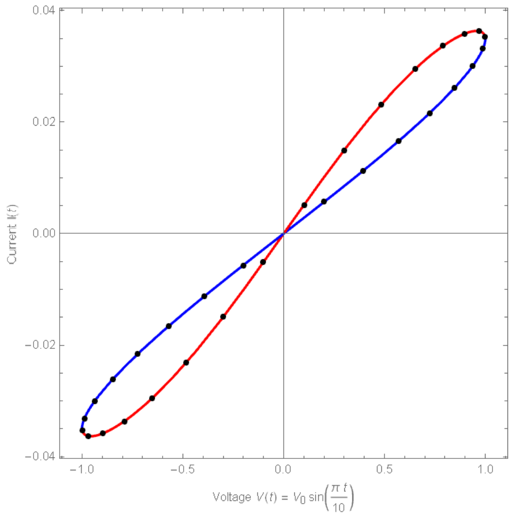


Fig. 1. Pinched hysteresis loop in the i-v curve

where v and i denote the input voltage and current respectively and x stands for the internal state of the device. In their paper, Chua and Kang also provided a more generalized concept of memristive systems with no specific reference to particular physical variables.

One noteworthy peculiarity derived from these equations is that regardless of the state x (which implements the memory effect), the output voltage is equal to zero whenever input voltage or current are equal to zero as well. This zero-crossing property, Chua and Kang write, manifests itself vividly in the form of a Lissajous figure, which always passes through the origin. Thus, they extended the definition of memristor that is now to encompass any system able to demonstrate a Lissajous figure (later called pinched hysteresis loop by Chua) in the i-v curve, which is presented on fig. 1.

3. Memristor Models

However, the true interest has been sparked by the notable work of Richard Stanley Williams' group of researchers at Hewlett-Packard laboratories. Despite this fact, the idea of memristors not being a purely theoretical concept has captivated minds of many researchers around the world, resulting in more than 120 publications about memristors and memristive systems by 2011. [8].

After the concept of memristor was brought back to the public's sight, several implementations of memristors and memristive systems have been proposed. Different implementations of memristor rely on various physical and chemical reactions that give rise to both memristance and nonvolatility, properties essentially constituting the definition of memristor. There have been reported polymeric [9,10], spintronic [11], ferroelectric [12] and layered [13] implementations of memristor, but titanium dioxide memristors remain the most well studied group. During this research four models were closely considered, namely linear ion drift model[1], nonlinear ion drift model[14], Simmons tunnel barrier mode[15], and threshold adaptive memristor model (TEAM)[16]. Unfortunately, due to the paper size considerations only the last one of them will be reported. This model, however, was decided to be further utilized throughout the work.

TEAM model, proposed by Kvatinsky et al., incorporates advantages of ion drift models' explicitness and Simmons tunnel barrier accuracy, yet manages to preserve relatively high computational performance and generalizability. TEAM model is based on the same physical behavior as Simmons tunnel barrier model. But it manages to convey it with simpler mathematical functions. The model introduces several assumptions for the sake of analytical simplicity: state variable does not change below a certain threshold and exponential dependence is replaced with a polynomial one. Detailed mathematical foundation of the model may be found in the corresponding paper.

A major advantage of such a relation is the explicitness of current and voltage relationship as opposed to the Simmons tunnel barrier model. Nevertheless, Kvatinsky et al. were able to perform a fitting procedure forcing TEAM model to match the latter with reasonable and sufficient accuracy. In their paper, authors of TEAM model also report the results of comparison between the fitted TEAM and Simmons tunnel barrier model. The feasible preciseness of TEAM model was proved by the average discrepancy between models' state variable difference of only 0.2%. The maximum difference of this value constituted 12.77%, however the run time of the model was nearly halfed (47.5%) Kvatinsky et al. had been also able to fit the model with different types of physical memristor models, namely STT-MRAM and Spintronic memristors.

4. Memristor Bridge Neural Network

This paper considers the neural network architecture proposed by Adhikari et al. in 2012 [4]. The architecture is based on the memristor-bridge synapse [17] and aims to solve the issue of nonvolatile synaptic weight storage and implement a newly proposed hardware learning method.

4.1 Memristor Bridge Synapse

Memristor bridge synapse architecture was first proposed in [17], it is a Wheatstone-bridge-like circuit that consists of four identical memristors of opposite polarities. When positive or negative strong pulse v_{in} (t) is applied at the input, the memristance of each memristor is increased or decreased depending upon its polarity.

Kim et al. write, that if input pulse voltage is equal to v_{in}, voltages at memristors can be calculated according to "voltage-divider formula". Then given memristances M_1, M_2, M_3, and M_4 stand for the corresponding memristors at time t, the output voltage is reported to be equal to the voltage difference between terminals A and B:

$$v_{out} = v_A - v_B = \left(\frac{M_2}{M_1 + M_2} - \frac{M_4}{M_3 + M_4}\right) v_{in}.$$

4.2 Memristor Bridge Neuron

In artificial neural networks neurons are required to sum a set of input postsynaptic signal and, according to the activation function, propagate (or not propagate) the signal further on to the next layer of the network. The neuron is then required to sum the input postsynaptic signals. Kim et al. point out, that the signal summing operation is easier to be performed in current mode: postsynaptic signals should be connected to a single node, so that the following neuron would receive the sum of currents via Kirchhoff's current law. In order to achieve current summation, the memristor bridge synapse has to be modified because it provides voltage output. Kim et al. suggest combining the memristor bridge with differential amplifier. The latter converts post-bridge negative and positive voltage into corresponding currents. Hence, for a set of synapses there exist two nodes: one for positive postsynaptic current and one for negative postsynaptic current. These nodes sum the output currents of each individual synapse in the set. Neuron itself is then comprised of the summation nodes, but also of the active load circuit that implements the activation function as in fig. 2. The sum of all postsynaptic currents is converted back to voltage (presynaptic signal for next layer of neural network) by the active load circuit according to the activation function. In their paper, Adhikari et al. also provide rigorous mathematical explanation of the suggested architecture behavior.

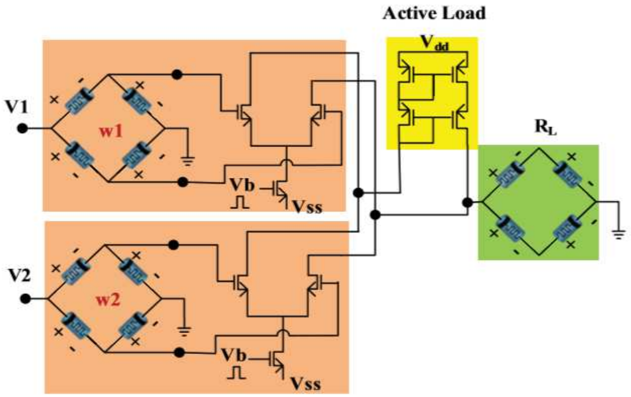


Fig. 2. Memristor Neural Network Circuit Fragment [4]

4.3 Neural Network Training

A composition of an arbitrary number of neurons connected via memristor-bridge synapses therefore constitutes the artificial network. Adhikari et al. intend to use Chip-in-the-Loop technique for training the network of proposed architecture. They, however, suggest modifying this technique slightly in order to take into account peculiar properties of memristor-based circuits. This technique is a viable choice since it provides a way to deal with memristor bridge non-idealities without explicitly modelling these nonidealities. According to this technique, the circuit performs the forward computation of the network, whereas back-propagation and weight update is done on the software side.

The hardware circuit network is reproduced by a software clone, which is used to process the training data. After the computer network has processed all the training data, synaptic weights of each individual synapse circuit are programmed by direct application of strong voltage pulses in order to match with the weights from computer network's weight matrix. Hence, the whole of the hardware network is treated as it consists of a set of simple single-layer networks. Each one of those single layer networks is trained separately, according to the weight matrix. Because of the nature of memristor bridge synapses, the need in additional circuitry is eliminated.

5. Framework Concept

As one can see, plenty of research has been carried out in the field of memristors and memristor-based neural networks. Multiple approaches to both creating and modelling memristors have been mentioned in previous sections.

It is needed to create a reliable framework for simulating memristor-based neural networks. So far, rather abundant overview of the domain has been presented. Despite the vast variety of works mentioned, the domain at hand lacks general integrity and is not formalized enough to start composing the framework at least in its basic form. Hence, the domain must be formalized to a certain extent. In order to derive this degree of formalization, the requirements for the stated framework are to be determined. This will enable framework to be designed properly and will ensure it complies with the needs and wants of its users. Requirements are decided to include four major points: accuracy, performance, flexibility, and explicitness. Accuracy stands for reliability of framework and if its output data can be trusted. Performance reflects how quick does the simulation proceed. Flexibility corresponds to how easy it is to swap components and models within framework. Finally, explicitness is determined by the overall convenience of the framework and how well does it represent results of the simulation. Insights into these requirements can be better revealed according to the SMART criteria (a project management technique for elaborating objectives), which is done in Table 2.

The requirements described above help determine what is to be expected from the framework, what kind of formalization for the domain is required, and set guidelines for further process of design and development. The domain may be formalized by representing it as a graphical scheme, henceforward called ontological model. The reason for such naming is that this model encompasses relevant entities of the domain under discourse, as well as reflects their major properties and interrelations, which in turn roughly corresponds to the definition of ontology. This model will limit the complexity of the field of memristor-based neural networks and expose the intrinsic connections between the notions at hand.

First, let us derive a set of entities to be found within this model. At the very core of every network there are neurons and synapses. These three notions (neural network, synapse and neuron) constitute the heart of designed model as well.

Multilayer network usually distinguishes between input layer neurons, output layer neurons and hidden layer neurons, which may slightly differ. Input neurons should be able to receive input signals, which may not necessarily coincide with how the signals are conveyed within the network. Similarly, output neurons must provide output signals. Consequently, input and output program modules should be introduced, in order to convert electrical output signals into human-comprehensible format and vice versa for the input signals.

Table 2. Framework requirements according to SMART

Criterion	Accuracy	Performance	Flexibility	Explicitness
Specificity	Results of simulation within framework must coincide with corresponding experimental data.	Simulation processing must be performed in a reasonable time.	Frameworks components must be easy to change and replace, due to the domain's novelty.	Simulation results should be clear and easy to observe.
Measurability	Given the same input data the framework must produce the same output data as in either experimental data or in verified models. Thus, the discrepancy between these results may be used to measure accuracy of the framework.	Time taken to perform the simulation and calculate the results reflects how well does the framework perform in terms of performance.	Framework's flexibility can measured in regard with how many approaches to memristor modelling and network training and architecture does it implement.	Explicitness is the most subjective of all requirements and should be estimated by direct responses of framework's users.
Achievability	Accuracy is achieved through testing the framework and tuning it match with known data.	Performance is achieved through optimization of frameworks algorithms and architecture.	If designed correctly the architecture (structure) of the framework should provide sufficient flexibility.	Various parameters of framework's components must be accessible and visualization methods (graphs, visual models, etc.) should be provided.
Relevance	Accuracy is arguably the most important requirement, without sufficient accuracy, the purpose of the framework is defeated.	Performance is quite relevant since long runtime may hinder the research progress when using framework.	Because the domain is so new, it is extremely important to make the framework able to adapt to possible changes.	Visual representation of simulation results is very important for the end user.
Timeliness	Accuracy may be achieved after tuning the initial version of framework.	Performance should be taken into account during the development, but can be also improved by later optimization.	Flexibility must be ensured from the very beginning of the development.	Visualization may be introduced after the basis of the framework is complete.

Both neurons and synapses of hardware neural networks are implemented through

Both neurons and synapses of hardware neural networks are implemented through

circuits. Circuit design may vary from one implementation to another, therefore, the general concept of neurons and synapses should be decoupled from its' particular hardware implementation to ensure flexibility. This will enable the framework to safely switch between specific circuit implementations of neurons and synapses, but will also ensure framework's operability. The framework must as well be able to switch between different realizations of memristor, namely, memristor models. Hence, the latter should be considered a separate entity, which is contently used as a component in synapse circuitry. For the time being only the metal dioxide class of memristors is considered to limit already reasonable complexity of the framework.

Finally, the network must should be able to employ different learning techniques. Despite the fact that this work considers only chip-in-the-loop method, the framework should be designed being able to implement various ways of network training. Here it is necessary to take into account not only the learning algorithm, but also how this algorithm is applied to hardware circuit components of the network.

The ontological model is depicted on fig. 3. Solid border circles correspond to the entities of the domain; dashed border circles stand for the properties (attributes) of certain entities; filled arrows represent association relation between entities; empty arrows reflect inheritance (or, possibly, interface implementation); finally, dashed lines reflect attribution connections.

It must be noticed, that the ontological model is likely to be changed in the following works and presented version is not final. Some of the anticipated issues include particular implementations of learning techniques, for instance, chip-in-the-loop does not require auxiliary circuitry, whereas spike timing-dependent plasticity usually does. Another bottleneck to be expected relates to the circuit implementations of neurons and synapses. The latter may consist of multiple circuits that should be represented as separate entities in order to preserve flexibility of the system, yet should conform to the same interface for the sake of integrity.

In this way we shed light onto the structural peculiarities of the future framework. This model is to help composing the classes to be implemented as well as their interrelations. Let us now consider the other side of the developed system, namely, its functional requirements. In this paper, the latter refer to a certain number of capabilities expected by users from the framework.

Framework under development strives to model memristor-based neural network suggested by Adhikari et al., which is described in the previous section. It is also expected to make possible modeling with better level of preciseness by enabling swappable memristor models. For instance, employing TEAM memristor model may significantly raise the relevance of proposed hardware neural network model through fostering the accuracy of memristor's physical model.

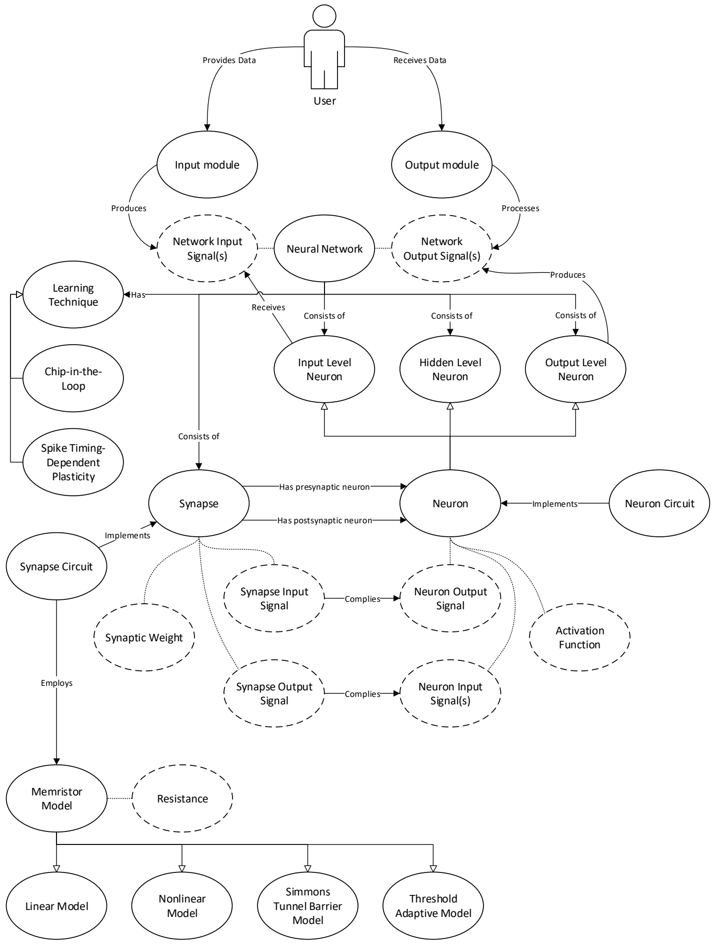


Fig. 3. Domain's Ontological Model

The functional scope of the framework may be represented as a set of intertwined mathematical equations that describe various parts of the network model. Each entity of the framework can be characterized with equations that have adjustable parameters, which are usually derived by the authors of corresponding models from experimental data analysis. These equations are extracted from relevant models and are bound in such way, that one equation's output usually corresponds to input of the other equation. This set of equations is depicted on fig. 4.

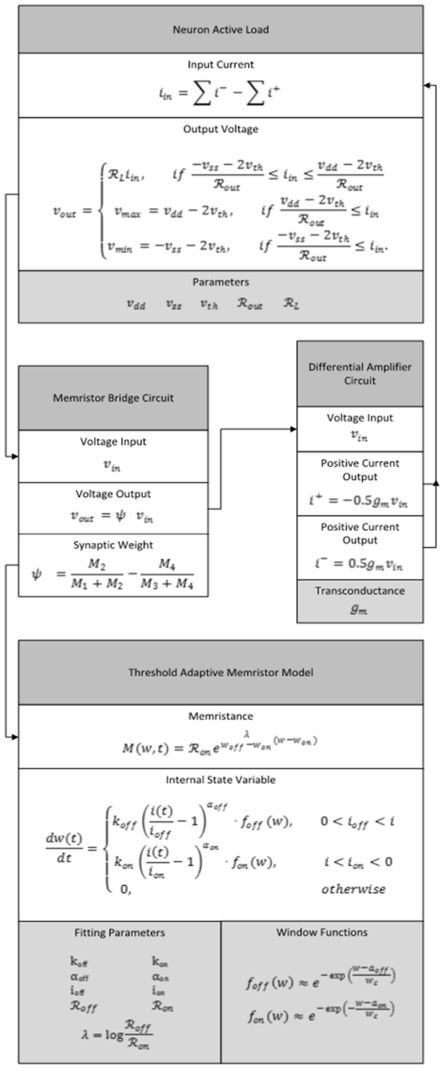

Neuron Active Load
Input Current
$i_{in} = \sum i^- - \sum i^+$
Output Voltage
$v_{out} = \begin{cases} \mathcal{R}_L i_{in}, & if\ \frac{-v_{ss}-2v_{th}}{\mathcal{R}_{out}} \le i_{in} \le \frac{v_{dd}-2v_{th}}{\mathcal{R}_{out}} \\ v_{max} = v_{dd} - 2v_{th}, & if\ \frac{v_{dd}-2v_{th}}{\mathcal{R}_{out}} \le i_{in} \\ v_{min} = -v_{ss} - 2v_{th}, & if\ \frac{-v_{ss}-2v_{th}}{\mathcal{R}_{out}} \le i_{in}. \end{cases}$
Parameters
v_{dd} v_{ss} v_{th} $\mathcal{R}_{out}$ $\mathcal{R}_L$
Memristor Bridge Circuit
Voltage Input
v_{in}
Voltage Output
$v_{out} = \psi\ \ v_{in}$
Synaptic Weight
$\psi = \frac{M_2}{M_1 + M_2} - \frac{M_4}{M_3 + M_4}$
Differential Amplifier Circuit
Voltage Input
v_{in}
Positive Current Output
$i^+ = -0.5 g_m v_{in}$
Positive Current Output
$i^- = 0.5 g_m v_{in}$
Transconductance
g_m
Threshold Adaptive Memristor Model
Memristance
$M(w,t) = \mathcal{R}_{on} e^{\frac{\lambda}{w_{off} - w_{on}}(w - w_{on})}$
Internal State Variable
$\frac{dw(t)}{dt} = \begin{cases} k_{off}\left(\frac{i(t)}{i_{off}} - 1\right)^{\alpha_{off}} \cdot f_{off}(w), & 0 < i_{off} < i \\ k_{on}\left(\frac{i(t)}{i_{on}} - 1\right)^{\alpha_{on}} \cdot f_{on}(w), & i < i_{on} < 0 \\ 0, & otherwise \end{cases}$
Fitting Parameters
k_{off} k_{on}
α_{off} α_{on}
i_{off} i_{on}
$\mathcal{R}_{off}$ $\mathcal{R}_{on}$
$\lambda = \log \frac{\mathcal{R}_{off}}{\mathcal{R}_{on}}$
Window Functions
$f_{off}(w) \approx e^{-\exp\left(\frac{w - a_{off}}{w_c}\right)}$
$f_{on}(w) \approx e^{-\exp\left(-\frac{w - a_{on}}{w_c}\right)}$

Fig. 4. Functional Structure

Each separate square on the scheme reflects an entity of the framework, while arrows denote the input-output connections between equations. One may notice that relations of equations form a cycle, where one iteration of this cycle corresponds to one layer of hardware memristor network. This figure depicts what set of functions is expected to be provided by the future framework.

6. Conclusion and Prospects

In this paper, a range of memristor models has been reviewed together with some of the fundamental papers on memristor-related technologies. Based on this review, a concept of framework for modeling memristor-based hardware neural networks has been proposed. This framework represents an implementation of neural network architecture considered in the paper, but implies ability to swap memristor models in order to increase the overall flexibility and, possibly, relevance of models generated with the help of proposed framework. The ability to switch between model is also expected to help comparing suggested implementations. In the process of framework structure discovery a set of criteria has been formulated to assess the future software product, domain of memristor-based neural networks has been formalized to a certain extent, and, finally, the framework has been given a functional structure strictly defining its' capabilities.

Specific platform for framework implementation is yet to be chosen. As of current state of affairs, Unity engine is expected to be the most favorable candidate. Its architecture perfectly fits the nature of soft simulation (which the framework ultimately represents), providing some software patterns that greatly alleviate the development. Considered engine is also able to realize extensive visualization of models as well as equip them with user-friendly interface to further enhance model explicitness and facilitate employment of the future framework for academic purposes. Finally, implementing a circuit simulation framework in Unity also pursues an exploration goal, since such attempts have not been previously well studied.

References

[1]. D. Strukov, G. Snider, D. Stewart and R. Williams, "The missing memristor found", Nature, vol. 453, no. 7191, pp. 80-83, 2008.

[2]. J. Mullins, "Memristor minds: The future of artificial intelligence", NewScientist Magazine, no. 2715, 2016.

[3]. S. Draghici, "Neural Networks in Analog Hardware - Design and Implementation Issues", International Journal of Neural Systems, vol. 10, no. 1, pp. 19-42, 2000.

[4]. S. Adhikari, Changju Yang, Hyongsuk Kim and L. Chua, "Memristor Bridge Synapse-Based Neural Network and Its Learning", IEEE Trans. Neural Netw. Learning Syst., vol. 23, no. 9, pp. 1426-1435, 2012.

[5]. T. Simonite, "A Better Way to Build Brain-Inspired Chips", Cacm.acm.org, 2015. [Online]. Available: http://cacm.acm.org/news/186782-a-better-way-to-build-brain-inspired-chips/fulltext. [Accessed: 30- Mar- 2016].

[6]. L. Chua, "Memristor-The missing circuit element", IEEE Trans. Circuit Theory, vol. 18, no. 5, pp. 507-519, 1971.

[7]. L. Chua and S. Kang, "Memristive devices and systems", Proceedings of the IEEE, vol. 64, no. 2, pp. 209-223, 1976.

[8]. A. Thomas, "Memristor-based neural networks", Journal of Physics D: Applied Physics, vol. 46, no. 9, p. 093001, 2013.

[9]. V. Erokhin and M. Fontana, "Electrochemically controlled polymeric device: a memristor (and more) found two years ago", Arxiv.org, 2008. [Online]. Available: http://arxiv.org/abs/0807.0333. [Accessed: 30- Mar- 2016].

[10]. F. Alibart, S. Pleutin, D. Guerin, C. Novembre, S. Lenfant, K. Lmimouni, C. Gamrat and D. Vuillaume, "An Organic Nanoparticle Transistor Behaving as a Biological Spiking Synapse", Adv. Funct. Mater., vol. 20, no. 2, pp. 330-337, 2010.

[11]. X. Wang, Y. Chen, H. Xi, H. Li and D. Dimitrov, "Spintronic Memristor Through Spin-Torque-Induced Magnetization Motion", IEEE Electron Device Lett., vol. 30, no. 3, pp. 294-297, 2009.

[12]. A. Chanthbouala, V. Garcia, R. Cherifi, K. Bouzehouane, S. Fusil, X. Moya, S. Xavier, H. Yamada, C. Deranlot, N. Mathur, M. Bibes, A. Barthelemy and J. Grollier, "A ferroelectric memristor", Nature Materials, vol. 11, no. 10, pp. 860-864, 2012.

[13]. A. Bessonov, M. Kirikova, D. Petukhov, M. Allen, T. Ryhänen and M. Bailey, "Layered memristive and memcapacitive switches for printable electronics", Nature Materials, vol. 14, no. 2, pp. 199-204, 2014.

[14]. E. Lehtonen and M. Laiho, "CNN using memristors for neighborhood connections", 2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA 2010), 2010.

[15]. M. Pickett, D. Strukov, J. Borghetti, J. Yang, G. Snider, D. Stewart and R. Williams, "Switching dynamics in titanium dioxide memristive devices", J. Appl. Phys., vol. 106, no. 7, p. 074508, 2009.

[16]. S. Kvatinsky, E. Friedman, A. Kolodny and U. Weiser, "TEAM: ThrEshold Adaptive Memristor Model", IEEE Trans. Circuits Syst. I, vol. 60, no. 1, pp. 211-221, 2013.

[17]. H. Kim, M. Sah, C. Yang, T. Roska and L. Chua, "Memristor Bridge Synapses", Proceedings of the IEEE, vol. 100, no. 6, pp. 2061-2070, 2012.

Обзор предметной области и концепция фреймворка для разработки моделей мемристоров и мемристорных нейронных сетей

[1] Д.Д. Кожевников <ddkozhevnikov@edu.hse.ru>
[2] Н.В. Красилич <nadezhda.krasilich@mail.ru>
[1] НИУ Высшая Школа Экономики,
101000, Россия, г. Москва, ул. Мясницкая, д. 20.
[2] НИУ Высшая Школа Экономики,
614070, Россия, г. Пермь, ул. Студенческая, д. 38.

Аннотация. В данной работе представлены предварительные результаты текущего исследования по разработке среды моделирования аппаратных мемристорных нейронных сетей. Проведен анализ релевантных трудов, описаны фундаментальные работы по мемристорам и мемристорным технологиям, рассмотрены различные физические реализации мемристоров, а также несколько математических моделей мемристоров из металло-диоксидной группы. Одна из таких моделей более подробно представлена в работе, описаны ее основные механизмы и наиболее интересные свойства. В работе также рассматривается недавно предложенная архитектура мемристорной нейронной сети, описывается методика обучения подобной аппаратной нейронной сети, реализация еу компонент: нейронов и синапсов на основе мемристорных мостов. В данной работе также выдвинуто предложение по улучшению этой архитектуры путем использования более точной модели мемристора в рамках сети. Основываясь на проведенном анализе предметной области, составлены и формально описаны требования к разработке среды моделирования мемристорных нейронных сетей. Кроме того, для лучшего понимания рассматриваемой предметной области составлены онтологическая и функциональная модели. Первая модель необходима для формализации объектной структуры предметной области, в то время как вторая модель используется для явного представления математических формул, описывающих физическое поведение соответствующих объектов. В совокупности обе модели позволяют составить полное, формализованное и многостороннее описание предметной области мемристорных нейронных сетей и перейти к процессу проектирования и разработки программного продукта. В конце работы кратко представлены дальнейшие перспективы разработки среды моделирования мемристорных нейронных сетей.

Ключевые слова: мемристор; модель мемристора; аппаратная нейронная сеть; мемристорная нейронная сеть.

DOI: 10.15514/ISPRAS-2016-28(2)-17

For citation: Кожевников Д.Д., Красилич Н.В. Обзор предметной области и концепция фреймворка для разработки моделей мемристоров и мемристорных нейронных сетей. Труды ИСП РАН, том 28, вып. 2, 2016, стр. 243-258 (на английском). DOI: 10.15514/ISPRAS-2016-28(2)-16

References

[1]. D. Strukov, G. Snider, D. Stewart and R. Williams, "The missing memristor found", Nature, vol. 453, no. 7191, pp. 80-83, 2008.

[2]. J. Mullins, "Memristor minds: The future of artificial intelligence", NewScientist Magazine, no. 2715, 2016.

[3]. S. Draghici, "Neural Networks in Analog Hardware - Design and Implementation Issues", International Journal of Neural Systems, vol. 10, no. 1, pp. 19-42, 2000.

[4]. S. Adhikari, Changju Yang, Hyongsuk Kim and L. Chua, "Memristor Bridge Synapse-Based Neural Network and Its Learning", IEEE Trans. Neural Netw. Learning Syst., vol. 23, no. 9, pp. 1426-1435, 2012.

[5]. T. Simonite, "A Better Way to Build Brain-Inspired Chips", Cacm.acm.org, 2015. [Online]. Available: http://cacm.acm.org/news/186782-a-better-way-to-build-brain-inspired-chips/fulltext. [Accessed: 30- Mar- 2016].

[6]. L. Chua, "Memristor-The missing circuit element", IEEE Trans. Circuit Theory, vol. 18, no. 5, pp. 507-519, 1971.

[7]. L. Chua and S. Kang, "Memristive devices and systems", Proceedings of the IEEE, vol. 64, no. 2, pp. 209-223, 1976.

[8]. A. Thomas, "Memristor-based neural networks", Journal of Physics D: Applied Physics, vol. 46, no. 9, p. 093001, 2013.

[9]. V. Erokhin and M. Fontana, "Electrochemically controlled polymeric device: a memristor (and more) found two years ago", Arxiv.org, 2008. [Online]. Available: http://arxiv.org/abs/0807.0333. [Accessed: 30- Mar- 2016].

[10]. F. Alibart, S. Pleutin, D. Guerin, C. Novembre, S. Lenfant, K. Lmimouni, C. Gamrat and D. Vuillaume, "An Organic Nanoparticle Transistor Behaving as a Biological Spiking Synapse", Adv. Funct. Mater., vol. 20, no. 2, pp. 330-337, 2010.

[11]. X. Wang, Y. Chen, H. Xi, H. Li and D. Dimitrov, "Spintronic Memristor Through Spin-Torque-Induced Magnetization Motion", IEEE Electron Device Lett., vol. 30, no. 3, pp. 294-297, 2009.

[12]. A. Chanthbouala, V. Garcia, R. Cherifi, K. Bouzehouane, S. Fusil, X. Moya, S. Xavier, H. Yamada, C. Deranlot, N. Mathur, M. Bibes, A. Barthelemy and J. Grollier, "A ferroelectric memristor", Nature Materials, vol. 11, no. 10, pp. 860-864, 2012.

[13]. A. Bessonov, M. Kirikova, D. Petukhov, M. Allen, T. Ryhänen and M. Bailey, "Layered memristive and memcapacitive switches for printable electronics", Nature Materials, vol. 14, no. 2, pp. 199-204, 2014.

[14]. E. Lehtonen and M. Laiho, "CNN using memristors for neighborhood connections", 2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA 2010), 2010.

[15]. M. Pickett, D. Strukov, J. Borghetti, J. Yang, G. Snider, D. Stewart and R. Williams, "Switching dynamics in titanium dioxide memristive devices", J. Appl. Phys., vol. 106, no. 7, p. 074508, 2009.

[16]. S. Kvatinsky, E. Friedman, A. Kolodny and U. Weiser, "TEAM: ThrEshold Adaptive Memristor Model", IEEE Trans. Circuits Syst. I, vol. 60, no. 1, pp. 211-221, 2013.

[17]. H. Kim, M. Sah, C. Yang, T. Roska and L. Chua, "Memristor Bridge Synapses", Proceedings of the IEEE, vol. 100, no. 6, pp. 2061-2070, 2012.