Chaos, Solitons and Fractals 143 (2021) 110548

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

Nonlinear Science, and Nonequilibrium and Complex Phenomena

journal homepage: www.elsevier.com/locate/chaos

Mathematical model of a neuromorphic network based on memristive elements

Alexander Yu. Morozov[a,b], Karine K. Abgaryan[a,b,*], Dmitry L. Reviznikov[a,b]

[a] Federal Research Centre "Information and Control" of the Russian Academy of Sciences, Russia, 119333, Moscow, st. Vavilova, 44, bld. 2
[b] Moscow Aviation Institute, Department of Information technology and applied mathematics, Russia, 125993, Moscow, Volokolamskoe Highway, 4

ARTICLE INFO

Article history:
Received 12 August 2020
Revised 19 November 2020
Accepted 4 December 2020

Keywords:
Memristor
Hafnium oxide
Neuromorphic network
Spiking neural network
STDP
Recognition

ABSTRACT

The article discusses the modeling of interconnected memory elements within a neuromorphic network. A mathematical model is proposed that describes a hardware analog implementation of a spiking neural network with memristive elements as synaptic weights and a learning mechanism based on the STDP (Spike-Timing Dependent Plasticity) rule. As a memristor model, a model describing a hafnium oxide (HfO2) element is used. A numerical simulation of the operation of one and two interconnected neurons with 64 and 128 synapses has been carried out. The process of training the network to recognize certain templates is demonstrated.

1. Introduction

In recent years, the development of areas related to neuromorphic systems has been active. A new generation of processors based on principles of human brain action is being created. For example, in 2014 IBM Research company presented TrueNorth system within the framework of DARPA SyNAPSE program; the system consists of 1 million digital neurons and 256 million synapses. This system is a hardware implementation of spiking neural networks, which are the third generation of artificial neural networks.

A distinctive feature of the existing systems is working in a digital format (all signals are coded with zeros and units). It is known that analog calculations are performed orders of magnitude faster than those performed with the use of a central processor and graphic accelerators; in this regard, the actual task is to create analog neuromorphic systems.

A memristor is a relatively new electrical element and it represents a resistor, which conductivity changes depending on the total electrical charge flowing through. In this case, the established resistance in the absence of current does not change over time. That is a memristor is an elementary long-term non-volatile memory cell [1-2]. The association of memristors into a matrix is called a memristor crossbar. Thanks to Ohm's law and the Kirchhoff's law, a fast analog product of the matrix per vector can be performed on the basis of a crossbar [3-4].

In addition to effective application of memristive elements in parallelization of matrix-vector operations, their use for analog implementation of self-training spiking neural networks is promising. Here a certain similarity of a memristive element to a biological synapse is used. Indeed, due to some schematic design analog solutions, which will be discussed in the article, there is an opportunity to change the conductivity (synaptic weight) directly in the process of the system, thus performing training network at the hardware level.

The paper deals with the mathematical model of a two-layer full-coupled network with one layer of memristor elements (synapses). Network training is organized using the Hebb's rule and synaptic plasticity (STDP, or Spike-Timing Dependent Plasticity method, according to which the change in weights of neuronal synapses depends on the time difference between input and output pulses) [5-10].

In the second section, we consider one of the mathematical models of memristor and compare its characteristics with experimental data for the memristor made on the basis of hafnium oxide (HfO_2). The third section formulates a mathematical model for a schematic design solution that implements a single-layer self-training spiking neural network with memristive elements as synaptic weights. In the fourth section, numerical simulation of one and two interconnected neurons with 64 and 128 synapses is performed, respectively. In conclusion, the main results of the work are formulated.

* Corresponding author:
E-mail addresses: morozov@infway.ru (A. Yu. Morozov), kristal83@mail.ru (K.K. Abgaryan), reviznikov@mai.ru (D.L. Reviznikov).

https://doi.org/10.1016/j.chaos.2020.110548
0960-0779/

2. Mathematical model of a memristor

The memristive effect usually occurs due to moving ions in an ultra-thin dielectric layer when an electric field is applied. In the case of various oxides it is often said about the movement of oxygen vacancies and the formation / destruction of conductive filaments. Most known memristor models are formulated as a dynamic system relative to the state of the memristor. Memristor state parameter is the value that corresponds to the position of the boundary separating the areas with low and high concentration of oxygen vacancies, thickness of a conducting layer, or thickness of a non-conducting barrier in which the tunnel current of electrons occurs. Depending on the law of change in the memristor state parameter, several mathematical models can be distinguished - in particular, linear [11] and nonlinear [12] drift models, and a model based on Simmons barrier [13]. Special window functions are introduced [14–17] to limit the state variable. Experimental data show that the state change does not occur at any voltage value, but starting from a certain threshold; therefore, threshold conditions are added to the model [18–20].

The model with nonlinear voltage dependence is considered. In general, the equation describing the state of the memristor can be presented in the following way:

$$\frac{dx}{dt} = a \cdot f(x) \cdot v^s,$$

Where $x \in [0,\ 1]$ is a state variable, a is a constant determined by material properties, v is a current voltage value, s is an odd integer, $f(x)$ is window function used for approximate representation of nonlinear ionic drift effects and boundary constraints. The Biolek window function is often used [15]:

$$f(x, v) = \begin{cases} 1 - (x-1)^{2p},\ v \le 0, \\ 1 - x^{2p},\ v > 0. \end{cases} \quad (1)$$

In the present work, the memristor model of this class is used, which is proposed in [21]:

$$\begin{gathered} \frac{dx}{dt} = a \cdot v^s \begin{cases} 1 - (1-x)^{2round\left(\frac{b}{|v|+c}\right)},\ v \le -v_{thr}, \\ 1 - x^{2round\left(\frac{b}{|v|+c}\right)},\ v > v_{thr}, \\ 0,\ -v_{thr} < v \le v_{thr}, \end{cases} \\ I = x^n \beta \sinh(\alpha_M v) + \chi[\exp(\gamma v) - 1], \end{gathered} \quad (2)$$

where I, v are current and voltage values; v_{thr} - activation voltage threshold value; n, β, α_M, χ, γ are adjustment parameters in the expression for current; *round* is the function for obtaining integer result; b, c are adjustment coefficients of the basic equation. It is not difficult to verify the similarity of the window function used in the model with the window function (1).

The memristor operation is simulated with the following values of parameters: $n = 5$, $\beta = 7.069 \cdot 10^{-5}$ A, $\alpha_M = 1.8$ V^{-1}, $\chi = 1.946 \cdot 10^{-4}$ A, $\gamma = 0.15$ V^{-1}, $a = 1$ V^{-5}, $s = 5$, $b = 15$ V, $c = 2$ V, $v_{thr} = 1$ V, $x(0) = 0.4$. The values of some parameters (α_M, γ, a, s, b, c, v_{thr}, $x(0)$) correspond to the values in the original work [21], and parameters β, χ are calculated to best fit (in root-mean-square metric) the experimental data on hafnium oxide (HfO_2) given in [21]. Fig. 1a shows a comparison of the obtained voltage-ampere characteristic with the experimental curve for HfO_2. Here the voltage time dependence (Fig. 1b) was taken from [21].

There is a satisfactory alignment of the simulation results with experimental data. Fig. 2a shows a model I – V characteristic for the voltage time dependence in the form of a sine (Fig. 2b).

The choice of this model is due to the fact that it implements the Lehtonen-Laiho memristor model which is based on physical experiments and on the mechanism of current flow through semiconductor amorphous transition metal oxides, has good accuracy and gives the possibility of fine tuning. The window function used in (2) makes it possible to apply the model for various voltages. The results of calculations show that the model is applicable for different voltage forms.

Without loss of generality, we can suppose that the memristor model can be represented in the form of two equations; the first one determines the dependence of the rate of change of the memristor state on current (I), applied voltage (v) and state (x), and the second one determines the value of the memristor resistance:

$$\frac{dx}{dt} = F_X(I, v, x),\ R = F_R(x, v)$$

An important feature of the memristor elements is their possible imperfection [22], which leads to uncontrolled changes in the conductivity level during system operation or initiation. In this case the memristor model can be described as a dynamic system with interval parameters and appropriate methods can be used to study the system [23-24].

One of the important issues of memristor properties is stochastic switching dynamics which is mainly resulted from random motion of ions (or oxygen vacancies) in dielectric film under the action of electric field. Detailed experimental and theoretical study of this effect is carried out in [25, 26] and [27]. It was noted that internal and external noise can play a positive role in the resistive switching process, inducing growth of contrast ratio between low and high resistive states as well as acceleration of relaxation process. These effects can be considered as nonlinear relaxation phenomena in metastable condensed matter systems under the noise action [28, 29].

The main approaches to taking stochastic effects into account in mathematical model are described in paper [25]. One of the simplest ways is to use stochastic differential equation for state variable instead of the deterministic one. In [30] authors used the equation of state with additive noise $d\mathbf{x} = -\nabla V(\mathbf{x})dt + G(t)dW$ (where $\mathbf{x}$ is the vector of variables of the stochastic process; $V(\mathbf{x})$ - potential inherent in the system; W - Wiener process; $G(t)$ is a function that specifies the signal entering the system) and observed effect of stochastic resonance in memristor. A more comprehensive model based on 1D drift - diffusion equation for oxygen vacancies concentration in memristor material is proposed in [25]. The model allowed authors to investigate thoroughly switching dynamics of memristors taking into account internal and external noise.

It has to be noted that complex mathematical model of spiking neuromorphic network presented in section 3 allows us to implement stochastic single memristor models of varying degrees of complexity. In the present paper we do not consider stochastic features of memristor switching dynamics, it is the subject of forthcoming work. However, we use random activation of arbitrary synapses in the network learning process, what is described in section 4. This ensures the memorization of information by the neural network.

3. Mathematical model of the neuromorphic network

The issues of constructing and simulation of circuit solutions for neuromorphic networks were considered in [31-35]. The ability of SDTP-training to pattern recognition at analog (hardware) level was shown for such networks. In the present paper, a rigorous mathematical model of a neuromorphic network is formulated in the form of a nonlinear dynamic system.

Let us consider the operation of the schematic design solution for a single-layer self-training analog spiking neural network with memristive elements in the capacity of synaptic weights (Fig. 3). Input pulses V_g open the corresponding transistors that lead to the current flow through the memristors with their subsequent summation in the neurons. The schematic design model of a neuron

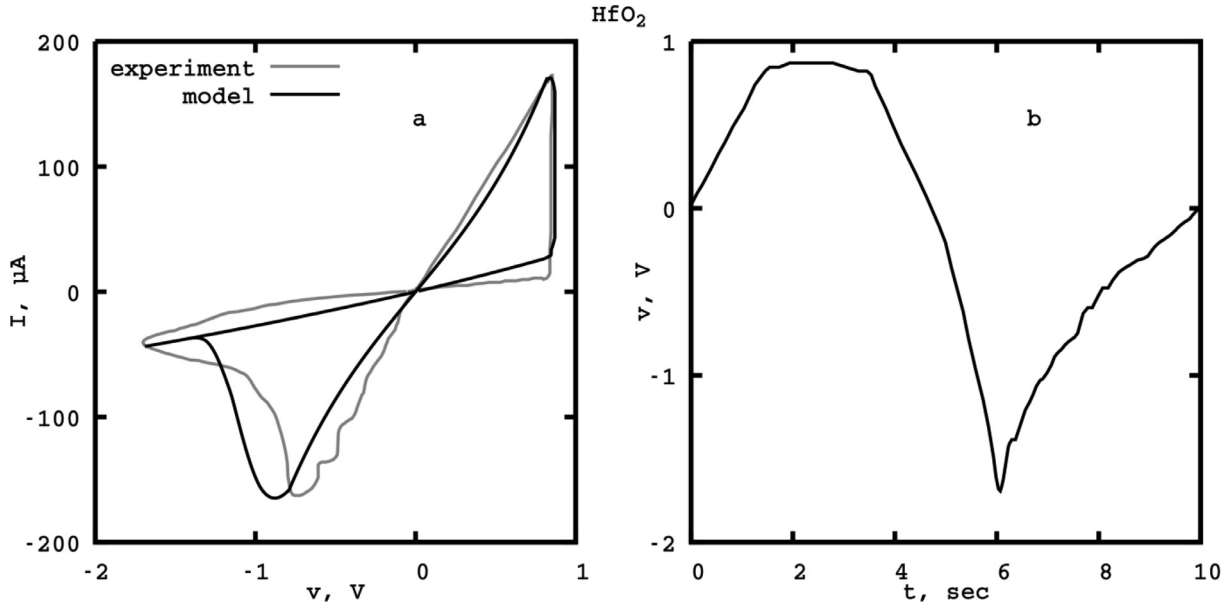

Fig. 1. Comparison of the volt-ampere characteristic of the model (2) with experimental data on hafnium oxide (a) at a certain form of input voltage (b).

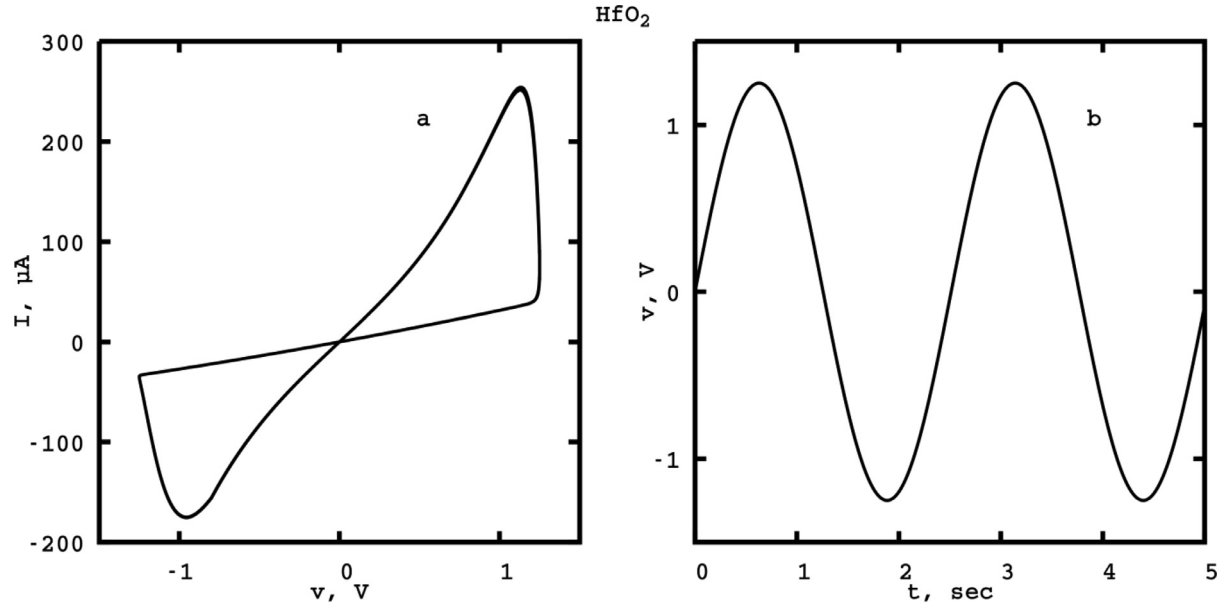

Fig. 2. Volt-ampere characteristic of model (2) (a) at the sine form of input voltage (b).

(the gray circle in Fig. 3) represents a parallel *RC* chain and an abstract pulse generator *G* (Fig. 4). As soon as the potential value on the capacitor exceeds some threshold, its potential is reset, and the pulse generator provides an output signal V_{out} and a feedback signal V_{te}. In addition, some small potential is continuously supported in the feedback line, which is necessary for the network to function in normal mode. Capacitance C_{int} and resistance R_{int} values are responsible for the rate of accumulation and decrease of potential in the neuron.

The network training process is carried out according to the STDP rule (those synaptic links that have led to the activation of the neuron are amplified, others are faded). This training mechanism is implemented by feedback from neurons (V_{te}). At the moment when the neuron is activated two pulses with opposite signs with delays are received through the feedback channel. If there is an activity on the synapse and a positive feedback pulse is received, the conductivity value of the corresponding memristor increases; if a negative feedback pulse is received, the conductivity of the memristor decreases. Fig. 5 illustrates the process of changing the synaptic weights by the example of one synapse.

Network training comes as follows: either random noise or a predefined template is applied to the network input with equal probability. After a while, the network adapts to the template recognition. In case of several templates, there will be several neurons with additional link in the output layer (in Fig. 3 it is marked with a letter α). When one neuron is activated, it suppresses other neurons (decreases their potential value). Templates are distributed among neurons in the training process.

Let us formulate a complex mathematical model of a single-layer self-training spiking neural network (Fig. 3). Let's define the main variables of the model: n — number of inputs, m — number of neurons; V_g^i — current voltage value at the i-th input of the neural network; V_{te}^j — current feedback voltage of the j-th neuron; V_{te}^+, V_{te}^-, V_{te}^0 — feedback pulse amplitude values and default volt-

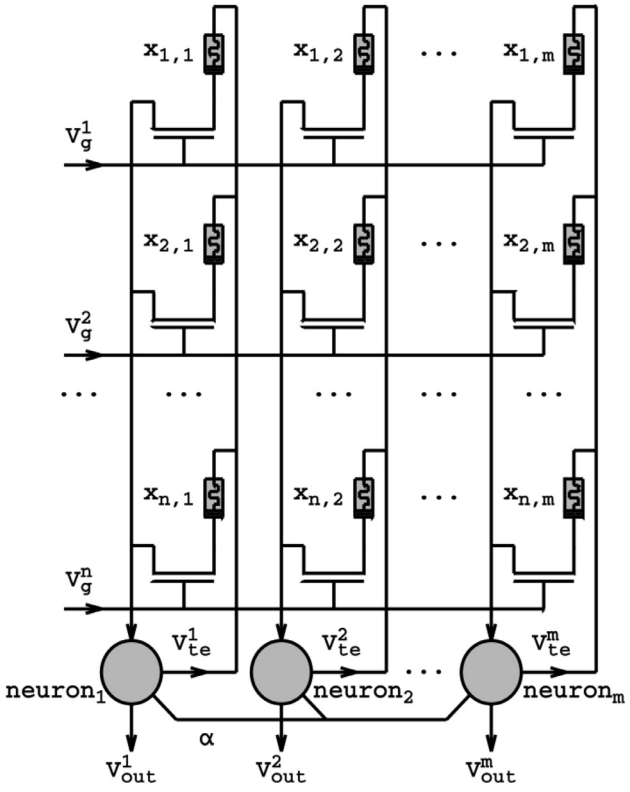

Fig. 3. Schematic design implementation of a spiking neural network.

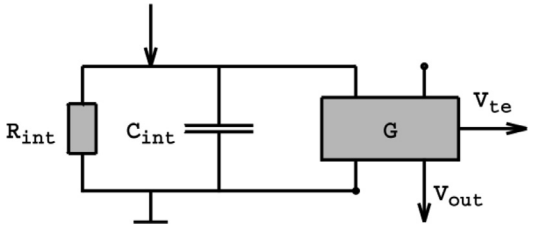

Fig. 4. Schematic design implementation of a neuron.

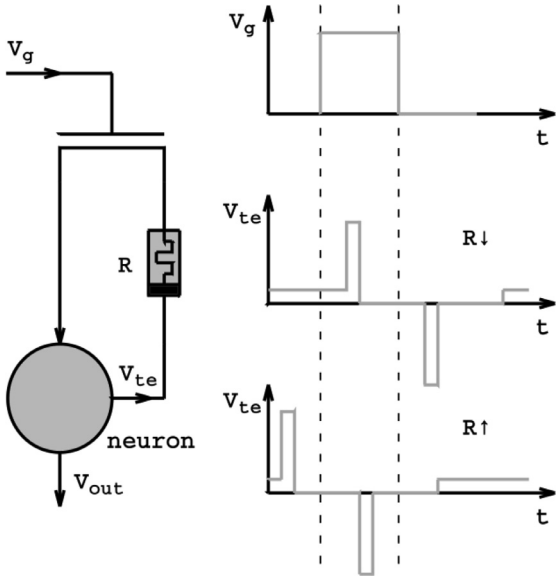

Fig. 5. Schematic design implementation of the STDP learning rule.

age values; V_{out}^{j} — current output voltage of the j-th neuron; V_{int}^{j} — voltage at the j-th neuron capacitor; V_{out}^{+} — output pulse amplitude; V_{th} — neuronal activation voltage level; R_{int}, C_{int} — neuronal resistance and capacitance values; $R_{i,j}$ — resistance value for the i-th synapse of the j-th neuron memristor; $x_{i,j}$ — condition of the i-th synapse of the j-th neuron memristor, $x_{i,j} \in [0,\ 1]$; τ_r — feedback signal duration after neuron activation; τ_s — single pulse duration in feedback signal, $2\tau_s < \tau_r$; τ_{out} — single output pulse duration; τ_j — the time elapsed since the last activation of the j-th neuron; α — suppression ratio.

The mathematical model is set by the following equations:

$$\begin{cases} \frac{dx_{i,j}}{dt} = \begin{cases} F_X\left(\frac{V_{\text{te}}^{j} - V_{\text{int}}^{j}}{R_{i,j}},\ V_{\text{te}}^{j} - V_{\text{int}}^{j}, x_{i,j}\right), & V_{\text{g}}^{i}(t) > 0, \\ 0, & V_{\text{g}}^{i}(t) = 0, \end{cases} \\ R_{i,j} = F_R(x_{i,j}, V_{\text{te}}^{j} - V_{\text{int}}^{j}), \end{cases} \tag{3}$$

$$\frac{dV_{\text{int}}^{j}}{dt} = \frac{1}{C_{\text{int}}}\left[\sum_{\substack{i=1 \\ V_{\text{g}}^{i}(t)>0}}^{n} \frac{\hat{V}_{\text{te}}^{j} - V_{\text{int}}^{j}}{R_{i,j}} - \frac{V_{\text{int}}^{j}}{R_{\text{int}}}\right] - \frac{\max}{i=\overline{1,m}}\left[\theta\left(V_{\text{int}}^{i} - V_{th}\right)\hat{\alpha}_{i,j}\right]\delta\left(\prod_{i=1}^{m}\left(V_{\text{int}}^{i} - V_{th}\right)\right)V_{\text{int}}^{j}, \tag{4}$$

$$\frac{d\tau_j}{dt} = 1 - \delta(V_{\text{int}}^{j} - V_{th})\tau_j, \tag{5}$$

$$V_{\text{te}}^{j} = \begin{cases} V_{\text{te}}^{+},\ \tau_j \le \tau_s, \\ 0,\ \tau_s < \tau_j \le \frac{\tau_r}{2}, \\ V_{\text{te}}^{-},\ \frac{\tau_r}{2} < \tau_j \le \frac{\tau_r}{2} + \tau_s, \\ 0,\ \frac{\tau_r}{2} + \tau_s < \tau_j \le \tau_r, \\ V_{\text{te}}^{0},\ \tau_r < \tau_j, \end{cases} \tag{6}$$

$$V_{\text{out}}^{j} = \begin{cases} V_{\text{out}}^{+},\ \tau_j \le \tau_{out}, \\ 0,\ \tau_{out} < \tau_j, \end{cases} \tag{7}$$

$$\hat{\alpha}_{i,j} = 1 - \alpha\left(1 - \delta_{ij}\right),\ i = \overline{1,n},\ \ j = \overline{1,m},$$

— and is supplemented by the initial conditions

$$x_{i,j}(0) = \text{rand}[0,\ 1],\ V_{\text{int}}^{j}(0) = 0,\ \tau_j(0) > \max(\tau_r, \tau_{out}),$$

where $\hat{V}_{\text{te}}^{j} = \max(0, \min(V_{\text{te}}^{j},\ V_{\text{te}}^{0}))$ and it is responsible for ensuring that feedback pulses do not contribute to the potential accumulation within the neuron; δ_{ij} — Kronecker delta; $\delta(x)$ — delta-function; $\theta(x)$ — Heaviside function. Eqs. (3) set the memristor model. The function $F_X(I, v, x)$ determines the speed of change of the state variable depending on current (I), voltage (v) and current state (x). The function $F_R(x, v)$ determines the dependence of the memristor resistance on the state and the applied voltage.

Eqs. (4) defines the model for the neuron, which represents a chain parallel to RC (Fig. 4) and connected in series with a resistor (all synapse memristors can be considered as a single resistor at the neuron level).

Eq. (5) implements the time counter mechanism after the last neuron activation. Once the voltage V_{int}^{j} on the capacitor reaches a threshold value V_{th}, the variable τ_j is reset to zero. The same happens in Eq. (4): after the neuron activation, the accumulated potential is reset, while in other neurons it decreases directly proportional to the coefficient α.

Expressions (6) and (7) determine the shape of pulses in the feedback and at the output of the neural network. At the start time, the variable $\tau_j$is selected so as to avoid the premature appearance of pulses in the feedback and output channels.

Eqs. (5)–(7) and the term $\max\limits_{i=\overline{1,m}}[\theta(V_{\text{int}}^{i} - V_{th})(1 - \alpha(1 - \delta_{ij}))]\delta(\prod_{i=1}^{m}(V_{\text{int}}^{i} - V_{th}))V_{\text{int}}^{j}$ in Eq. (4) describe the logic of

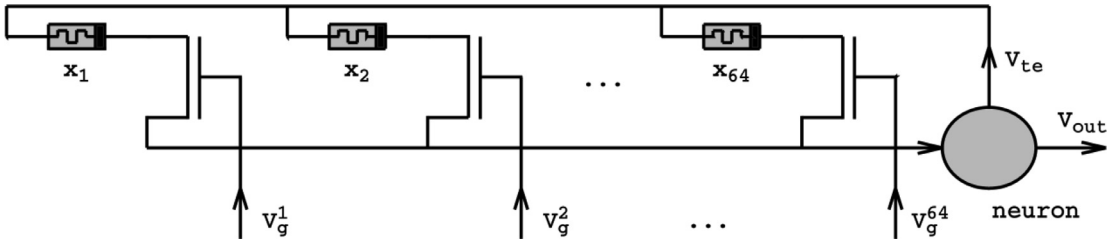

Fig. 6. A neural network consisting of a single neuron.

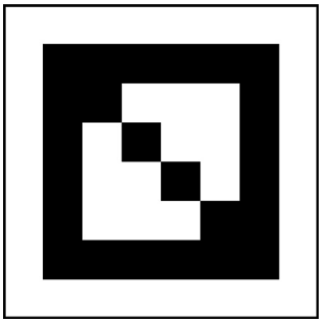

Fig. 7. The template for recognition $V_g = \mathrm{V}$.

the abstract pulse generator G (Fig. 4). At the time when the voltage V_{int}^i in the i - th neuron reaches the threshold value V_{th},V_{int}^i is reset, and the voltages of other neurons decrease by α times (for this, the delta function and the Heaviside function are used).

Note that equations (4) (neuron model) is not connected directly with differential part of equations (3) (memristor model). It allows us to implement various types of memristor switching dynamics models into consideration.

The above mathematical model describes only one neural network layer. To model a multilayer network, it is enough to connect the output of the k-th layer to the input of the $(k+1)$-th layer: $V_{\text{g}}^{i(k+1)}(t) = V_{\text{out}}^{j(k)}(t)$, $i = j$. Thus, the area of application of the described mathematical model is not limited to a single-layer network.

The section 3 presents a rigorous mathematical model of a neuromorphic network with memristive elements as synaptic weights. All the equations that describe a layer of the neuromorphic network are given explicitly. The constructed model is versatile and allows one to implement various models of memristive elements. The authors have developed their own program code that implements this model in C++.

4. Modeling results

When modeling the operation of the schematic design solution, the model (2) is used for the memristor. Accordingly, the functions $F_X(I, v, x)$ and $F_R(x, v)$, being part of the system of equations (3)-(7), are defined as follows:

$$F_X(v,x) = a \cdot v^s \begin{cases} 1-(1-x)^{2round\left(\frac{b}{|v|+c}\right)}, & v \le -v_{thr}, \\ 1-x^{2round\left(\frac{b}{|v|+c}\right)}, & v > v_{thr}, \\ 0, & -v_{thr} < v \le v_{thr}, \end{cases}$$

$$F_R(x,v) = \frac{v}{x^n\beta \sinh(\alpha_M v)+\chi[\exp(\gamma v)-1]}.$$

Let's consider a network consisting of a single neuron ($m = 1$, therefore the index j is omitted further) with 64 synapses (Fig. 6).

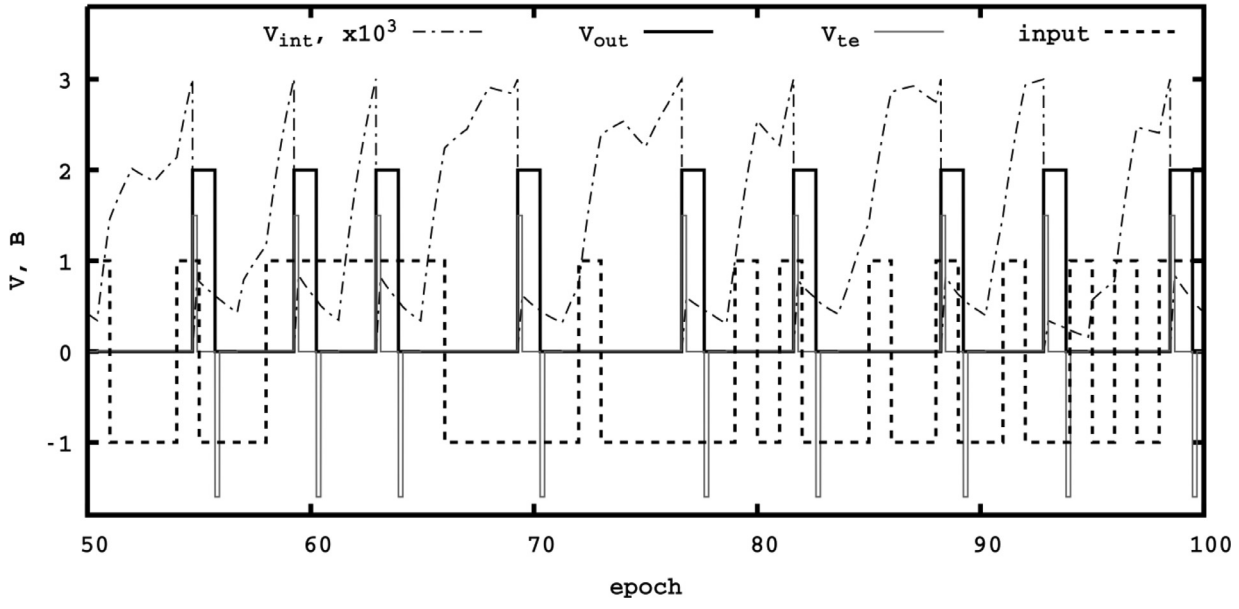

Fig. 8. The process of training a neural network from the 50th to the 100th epoch.

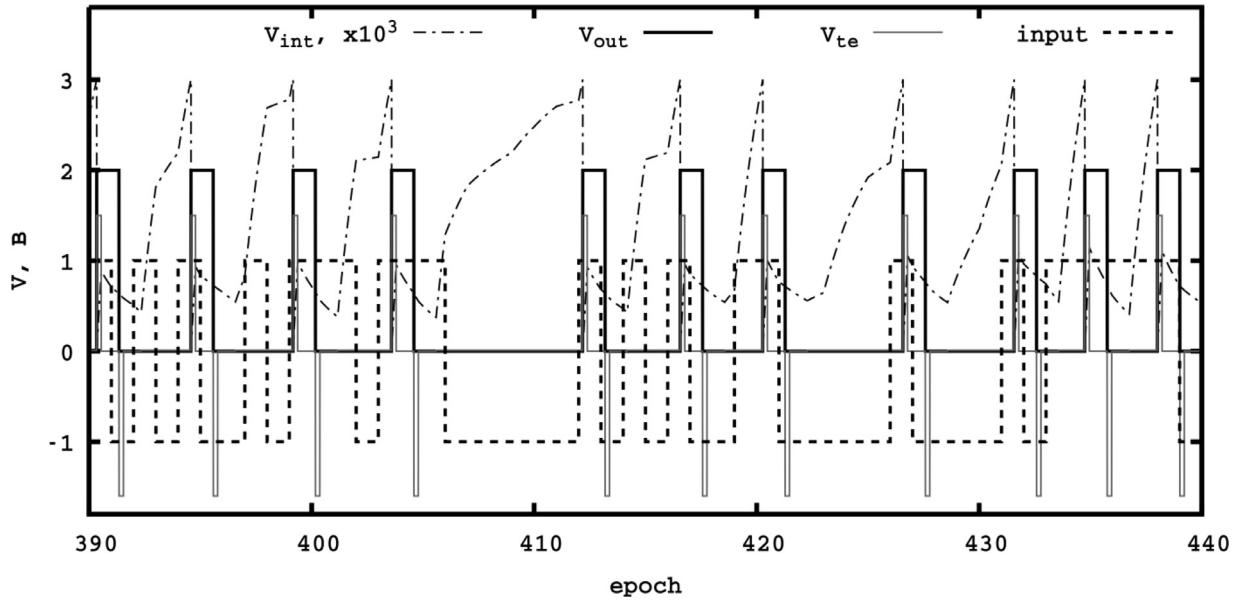

Fig. 9. The process of training the neural network from the 390th to the 440th epoch.

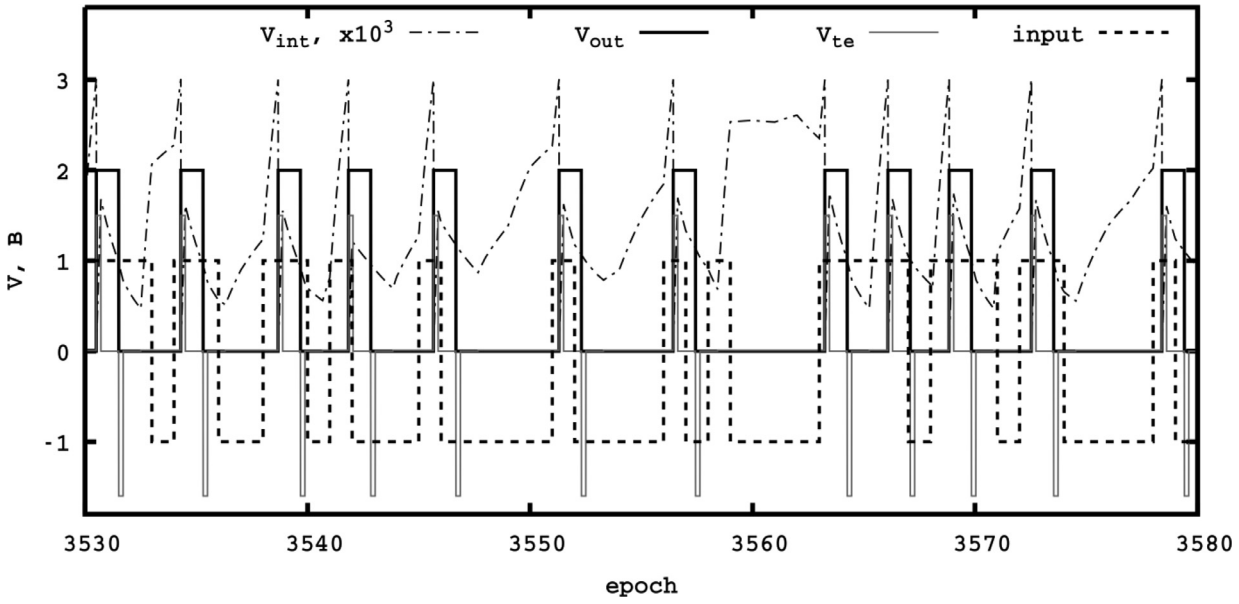

Fig. 10. Neural network operation at the end of training process.

The dynamic system has the following form:

$$\begin{cases} \frac{dx_i}{dt} = \begin{cases} F_X\left(\frac{V_{\text{te}}(\tau)-V_{\text{int}}}{R_i},\ V_{\text{te}}(\tau)-V_{\text{int}}, x_i\right),\ V_{\text{g}}^i(t) > 0, \\ 0,\ V_{\text{g}}^i(t) = 0, \end{cases} \\ \frac{dV_{\text{int}}}{dt} = \frac{1}{C_{\text{int}}}\left[\sum\limits_{\substack{i=1 \\ V_{\text{g}}^i(t)>0}}^{n} \frac{\hat{V}_{\text{te}}(\tau)-V_{\text{int}}}{R_i} - \frac{V_{\text{int}}}{R_{\text{int}}}\right] - \delta(V_{\text{int}}-V_{th})V_{\text{int}}, \\ \frac{d\tau}{dt} = 1-\delta(V_{\text{int}}-V_{th})\tau,\ R_i = F_R(x_i, V_{\text{te}}(\tau)-V_{\text{int}}), \\ x_i(0) = \text{rand}[0,\ 1],\ V_{\text{int}}(0) = 0, \tau(0) > \max(\tau_r, \tau_{out}),\ i = \overline{1,n} \end{cases}$$

And is supplemented by ratios (6) and (7). Parameter values are: $n = 64$, $R_{\text{int}} = 1\ \text{k}\Omega$, $C_{\text{int}} = 45\ \mu\text{F}$, $V_{\text{te}}^{+} = 1.5$ V, $V_{\text{te}}^{-} = -1.6$ V, $V_{\text{te}}^{0} = 10$ mV, $V_{\text{out}}^{+} = 2$ V, $V_{\text{th}} = 3$ mV, $\tau_r = 20$ ms, $\tau_s = 2$ ms, $\tau_{out} = 10$ ms. The values have been chosen so that the relaxation coefficient ($= R \cdot C$) would be equal to 45 ms, which corresponds to the value given in [6].

Table 1
Discrete noise distribution for single-neuron network training.

V_g^i	0 V	2 V
P	0.85	0.15

During the simulation process, vector components every 10 ms can with equal probability be either a random noise (V_g^i has a discrete distribution (see Table 1)) or take certain values which are set according to the recognized template (Fig. 7). For clarity, the vector $V_g(t)$ is recorded here as a matrix:

Below are the graphs of neural network voltage behavior as a function of time (Fig. 8, Fig. 9 and Fig. 10).

One epoch of training equals 10 ms, i.e. the time during which a sample or random noise is demonstrated for the neural network. Narrow peaks correspond to the voltage in neuronal feedback. The

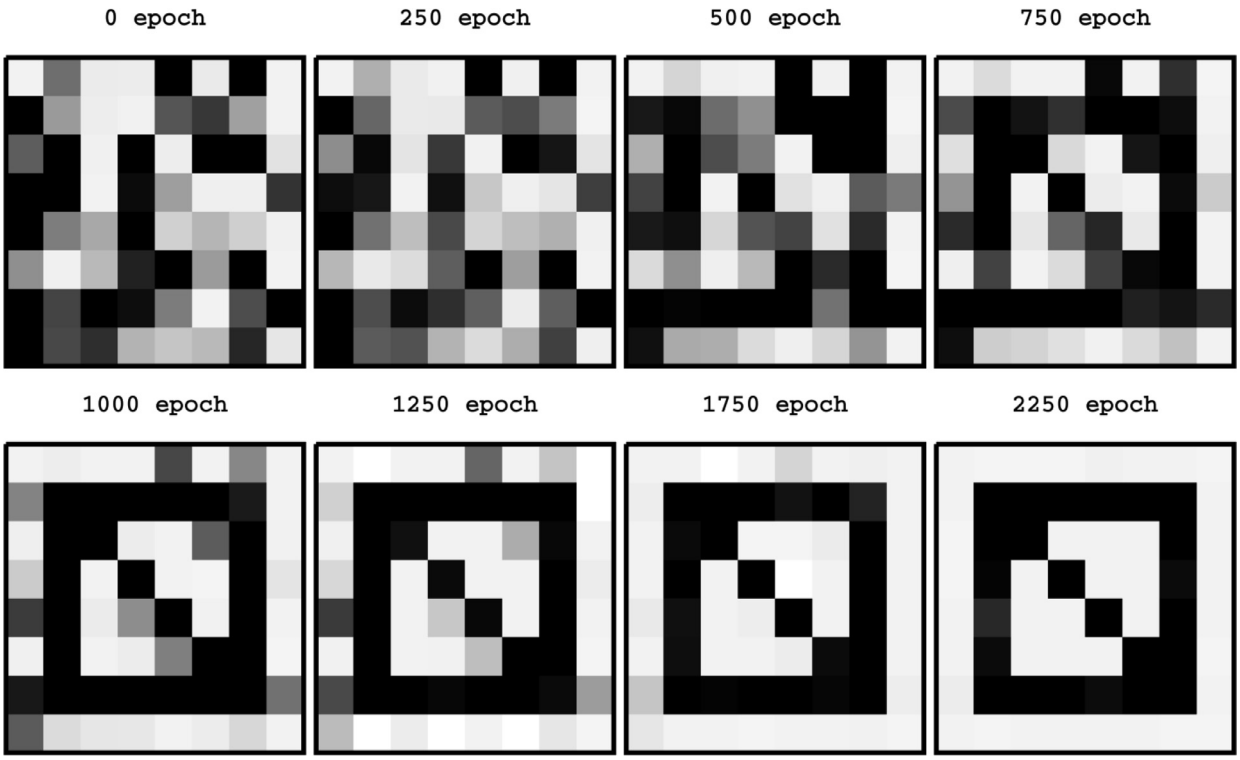

Fig. 11. Change of synaptic weights during training.

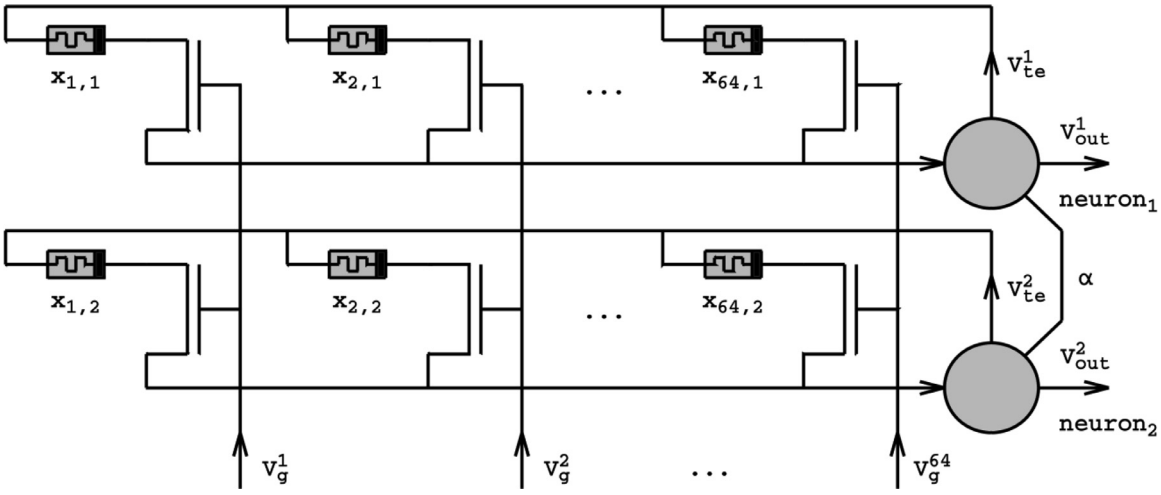

Fig. 12. A neural network consisting of two neurons.

curve line shows the accumulated potential in the neural capacitor. As soon as its value reaches the threshold value (3 mV on the graph; 3V according to the scale), an output pulse and feedback pulses are generated. A dotted line (input) shows what exactly is being shown by the neural network at each moment of time. The value −1 corresponds to noise, and the value 1 corresponds to the sample.

If at the beginning of the training process (Fig. 8) there are numerous template omissions and false responses to noise (fourth and fifth neuron activations), then after a certain number of epochs (Fig. 9) the network almost always reacts to the template and ignores the noise. At the end of the training process (Fig. 10), the network does not respond to the template unless the neuron has been activated before and the training pulses allow the network to operate normally. Note that the potential inside the neuron not only increases but also decreases at the time the network noise is demonstrated.

Fig. 11 shows the process of adaptation of synaptic weights to the sample to be recognized. The shade of gray corresponds to the resistance value of the corresponding memristor: the darker, the conductivity is greater. In the initial state, all weights are initialized by random values and are gradually changed during the network operation. A template that the network learns to recognize begins to be seen from approximately 750th epoch: the information is memorized by the neural network.

The task of recognition of two samples is considered below. The schematic design implementation of the network is shown in Fig. 12. In contrast to the previous example, there is an interaction between neurons. When one neuron is activated, the potential value of the other neuron will decrease directly propor-

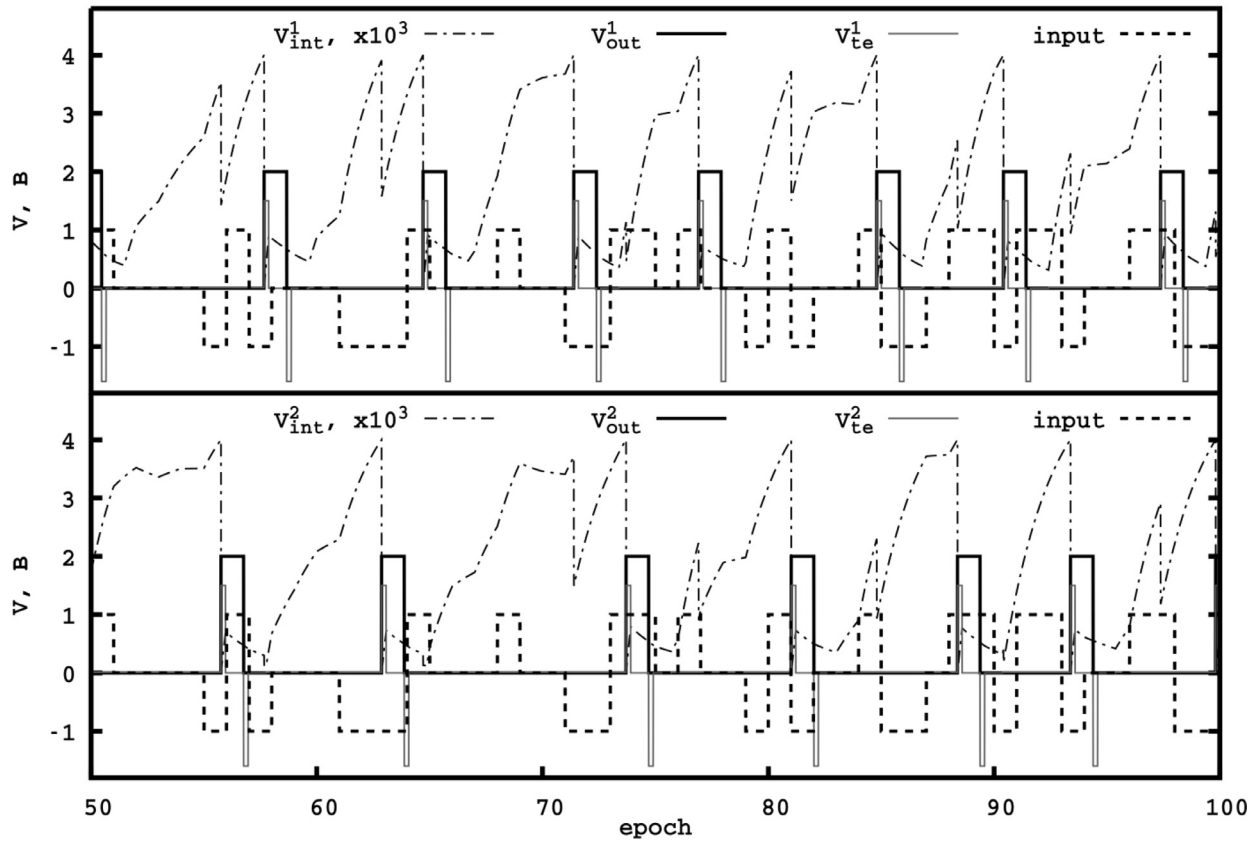

Fig. 13. The process of training a neural network from the 50th to the 100th epoch.

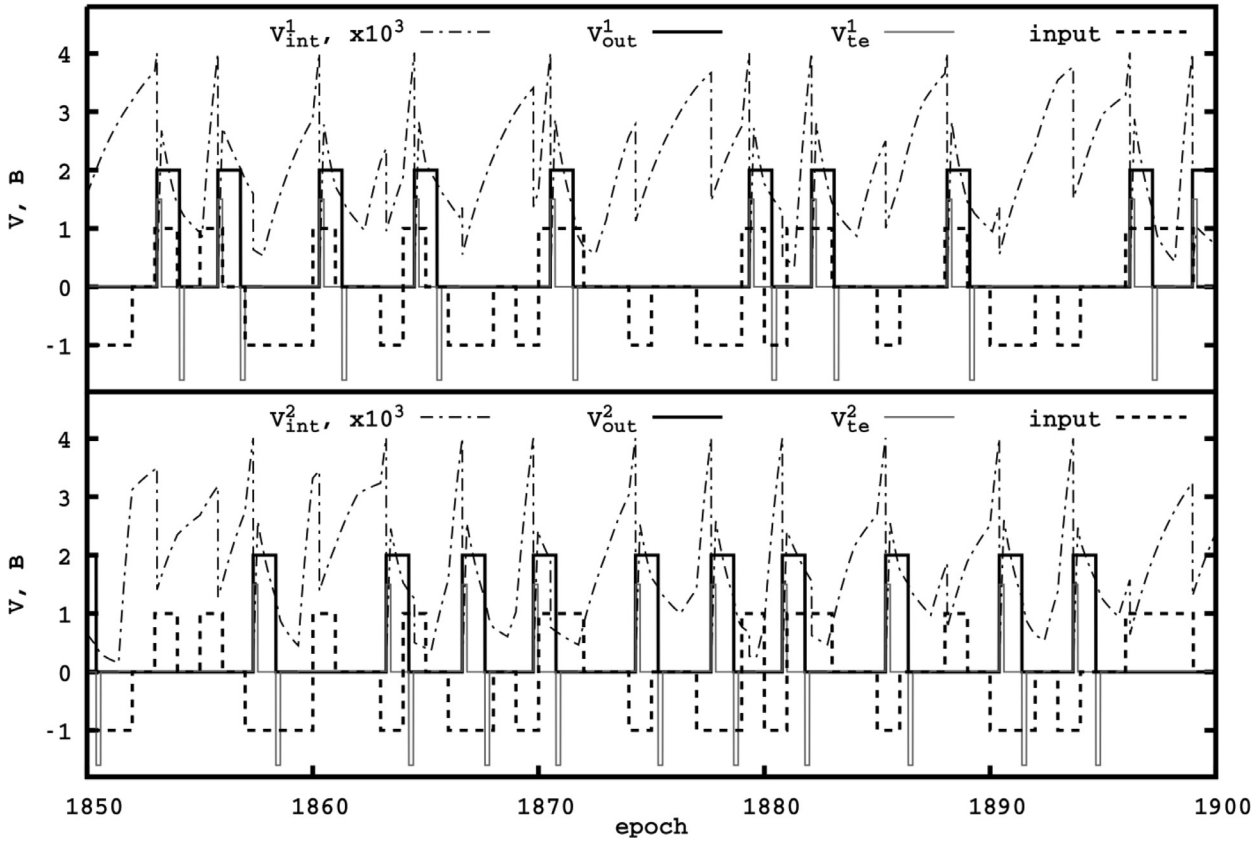

Fig. 14. Neural network operation at the end of training process.

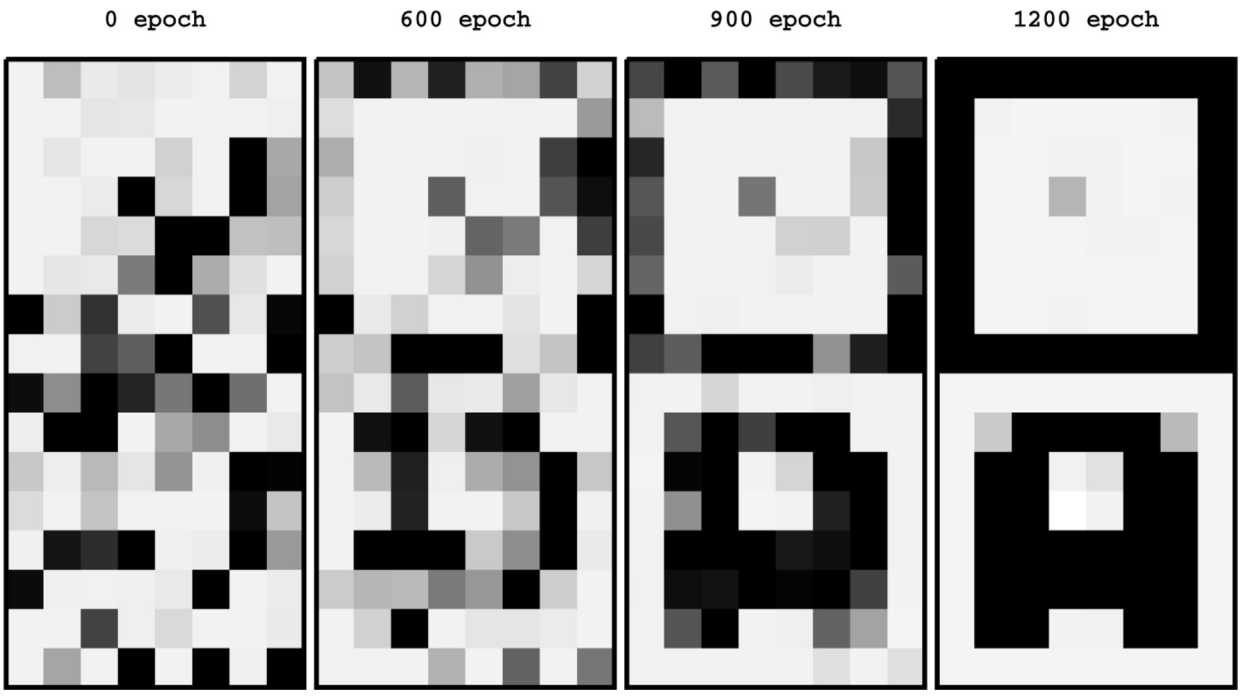

Fig. 15. Change of synaptic weights during training.

Table 2
Discrete noise distribution for training a network with two neurons.

V_g^i	0 V	2 V
P	0.8	0.2

tional to the suppression coefficient $\alpha = 0.4$. The activation threshold $V_{\text{th}} = 4$ mV, and other parameters of the model are the same.

As before, every 10 ms the vector $V_g(t)$ components can be with equal probability either a random noise (V_g^i has a discrete distribution (Table 2)) or take also with equal probability one of two values which are defined by the recognized templates (letter "a" and square).

$$V_g^{(1)} = \begin{pmatrix} 0,0,0,0,0,0,0,0, \\ 0,0,2,2,2,2,0,0, \\ 0,2,2,0,0,2,2,0, \\ 0,2,2,0,0,2,2,0, \\ 0,2,2,2,2,2,2,0, \\ 0,2,2,2,2,2,2,0, \\ 0,2,2,0,0,2,2,0, \\ 0,0,0,0,0,0,0,0 \end{pmatrix}^T \text{V},\ V_g^{(2)} = \begin{pmatrix} 2,2,2,2,2,2,2,2, \\ 2,0,0,0,0,0,0,2, \\ 2,0,0,0,0,0,0,2, \\ 2,0,0,0,0,0,0,2, \\ 2,0,0,0,0,0,0,2, \\ 2,0,0,0,0,0,0,2, \\ 2,0,0,0,0,0,0,2, \\ 2,2,2,2,2,2,2,2 \end{pmatrix}^T \text{V}$$

Fig. 13 and Fig. 14 show the voltage graphs for each neuron individually. Dotted line (input) corresponds to the input of the neural network: value 0 - noise, value -1 - first sample, value 1 - second sample. Sharp jumps are clearly visible on the capacitor potential graphs; it is due to the activation of one of neurons and suppression of another neuron.

From the point of view of recognition quality, chaos is shown at the beginning of the training process (Fig. 13): the same neuron is activated at different templates shown or both neurons ignore templates at all. At the end of the training process (Fig. 14), each neuron only triggers at its specific input signal.

Fig. 15 shows the process of adapting synaptic weights to the recognized templates. In contrast to the previous example, here the templates begin to be viewed from around the 600th epoch.

It should be noted that the results obtained using the developed simulation model are in good agreement with the results published in [31-35].

The work was performed with the support of RFBR grant No. 19-29-03051mk.

5. Conclusions

The article is devoted to modeling neuromorphic networks. The mathematical model of the memristor on the basis of hafnium oxide is considered; the comparison of the model characteristics with experimental data is carried out. The complex mathematical model of the spiking neuromorphic network with the training mechanism according to STDP (Spike-Timing Dependent Plasticity) rule is formulated. Modeling of two neural networks consisting of one neuron with 64 synapses and two neurons with 128 synapses, respectively, is performed. The neural networks have been successfully trained to recognize certain templates in the process of study.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

CRediT authorship contribution statement

Alexander Yu. Morozov: Methodology, Software, Investigation. **Karine K. Abgaryan:** Methodology, Supervision. **Dmitry L. Reviznikov:** Methodology, Investigation, Validation.

References

[1] Wong H-SP, et al. Metal–oxide RRAM. Proceedings of the IEEE 2012;100(6):1951–70. doi:10.1109/JPROC.2012.2190369.
[2] Yang JJ, Strukov DB, Stewart DR. Memristive devices for computing. Nature Nanotechnology 2013;8(1):13–24. doi:10.1038/nnano.2012.240.
[3] Li C, Hu M, Li Y, Jiang H, Ge N, Montgomery E, Zhang J, Song W, Dávila N, Graves CE, Li Z, Strachan JP, Lin P, Wang Z, Barnell M, Wu Q, Williams RS, Yang JJ, Xia Q. Analogue signal and image processing with large memristor crossbars. Nature Electronics 2017;1(1):52–9. doi:10.1038/s41928-017-0002-z.

[4] Hu M, Graves CE, Li C, Li Y, Ge N, Montgomery E, Davila N, Jiang H, Williams RS, Yang JJ, Xia O, Strachan JP. Memristor-Based Analog Computation and Neural Network Classification with a Dot Product Engine. Advanced Materials 2018;30(9). doi:10.1002/adma.201705914.
[5] Diehl P, Cook M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Frontiers in Computational Neuroscience 2015;9:00099. doi:10.3389/fncom.2015.00099.
[6] Ambrogio S, et al. Neuromorphic Learning and Recognition With One-Transistor-One-Resistor Synapses and Bistable Metal Oxide RRAM. IEEE Transactions on Electron Devices 2016;63(4):1508–15. doi:10.1109/TED.2016.2526647.
[7] Guo Y, Wu H, Gao B, Qian H. Unsupervised Learning on Resistive Memory Array Based Spiking Neural Networks. Front Neurosci 2019;13:00812. doi:10.3389/fnins.2019.00812.
[8] Milo V, et al. Resistive switching synapses for unsupervised learning in feed-forward and recurrent neural networks. In: 2018 IEEE International Symposium on Circuits and Systems (ISCAS). Florence; 2018. p. 1–5. doi:10.1109/ISCAS.2018.8351824.
[9] Pedretti G, Bianchi S, Milo V, Calderoni A, Ramaswamy N, Ielmini D. Modeling-based design of brain-inspired spiking neural networks with RRAM learning synapses. 2017 IEEE International Electron Devices Meeting (IEDM) San Francisco, CA; 2017. 28.1.1–28.1.4. doi:10.1109/IEDM.2017.8268467.
[10] Milo V, Ielmini D, Chicca E. Attractor networks and associative memories with STDP learning in RRAM synapses. 2017 IEEE International Electron Devices Meeting (IEDM) San Francisco, CA; 2017. 11.2.1–11.2.4. doi:10.1109/IEDM.2017.8268369.
[11] Strukov DB, Snider GS, Stewart DR, Williams RS. The missing memristor found. Nature 2008;453(7191):80. doi:10.1038/nature06932.
[12] Yang JJ, Pickett MD, Xuema L, Ohlberg DAA, Stewart DR, Williams RS. Memristive switching mechanism for metal/oxide/metal nanodevices. Nature nanotechnology 2008;3(7):429. doi:10.1038/nnano.2008.160.
[13] Pickett MD, Stukov DB, Borghetti JL, Yang JJ, Snider GS, Stewart DR, Williams RS. Switching dynamics in titanium dioxide memristive devices. Journal of Applied Physics 2009;106(7). doi:10.1063/1.3236506.
[14] Joglekar YN, Wolf SJ. The elusive memristor: properties of basic electrical circuits. European Journal of Physics 2009;30(4):661. doi:10.1088/0143-0807/30/4/001.
[15] Biolek Z, Biolek D, Biolkova V. SPICE Model of Memristor with. Nonlinear Dopant Drift. Radioengineering. 2009;18(2).
[16] Prodromakis T, Peh BP, Papavassiliou C, Toumazou C. A versatile memristor model with nonlinear dopant kinetics. IEEE Transactions on Electron Devices 2011;58(9):3099–105. doi:10.1109/TED.2011.2158004.
[17] Zha J, Huang H, Liu Y. A novel window function for memristor model with application in programming analog circuits. IEEE Transactions on Circuits and Systems II: Express Briefs 2015;63(5):423–7. doi:10.1109/TCSII.2015.2505959.
[18] Kvatinsky S, Friedman EG, Kolodny A, Weiser UC. TEAM: Threshold adaptive memristor model. IEEE Transactions on Circuits and Systems I: Regular Papers 2012;60(1):211–21. doi:10.1109/TCSI.2012.2215714.
[19] Kvatinsky S, Ramadan M, Friedman EG, Kolodny A. VTEAM: A general model for voltage-controlled memristors. IEEE Transactions on Circuits and Systems II: Express Briefs 2015;62(8):786–90. doi:10.1109/TCSII.2015.2433536.
[20] Yakopcic C., Taha T. M., Subramanyam G., Pino R. E., Rogers S. A memristor device model. IEEE electron device letters. 2011;32(10):1436–1438. 10.1109/LED.2011.2163292.
[21] Mladenov V. Analysis of Memory Matrices with HfO2 Memristors in a. PSpice Environment. Electronics 2019;8(4). doi:10.3390/electronics8040383.
[22] Teplov GS, Gornev ES. Multilevel bipolar memristor model considering deviations of switching parameters in the Verilog-A language. Russian Microelectronics 2019;48(3):131–42. doi:10.1134/S1063739719030107.
[23] Morozov AY, Reviznikov DL. Adaptive Interpolation Algorithm Based on a kd-Tree for Numerical Integration of Systems of Ordinary Differential Equations with Interval Initial Conditions. Differential Equations, 2018;54(7):945–56. doi:10.1134/S0012266118070121.
[24] Yu Morozov A, L Reviznikov D, Yu Gidaspov V. Adaptive Interpolation Algorithm Based on a KD-Tree for the Problems of Chemical Kinetics with Interval Parameters. Mathematical Models and Computer Simulations 2019;11(4):622–33. doi:10.1134/S2070048219040100.
[25] Agudov NV, Safonov AV, Krichigin AV, Kharcheva AA, Dubkov AA, Valenti D, Guseinov DV, Belov AI, Mikhaylov AN, Carollo A, Spagnolo B. Nonstationary distributions and relaxation times in a stochastic model of memristor. Journal of Statistical Mechanics: Theory and Experiment 2020:024003. doi:10.1088/1742-5468/ab684a.
[26] Mikhaylov AN, Gryaznov EG, Belov AI, Korolev DS, Sharapov AN, Guseinov DV, Tetelbaum DI, Tikhov SV, Malekhonova NV, Bobrov AI, Pavlov DA, Gerasimova SA, Kazantsev VB, Agudov NV, Dubkov AA, Rosário CMM, Sobolev NA, Spagnolo B. Field- and irradiation-induced phenomena in memristive nanomaterials. Phys. Status Solidi C - Current Topics in Solid State Physics 2016;13(10-12):870–81. doi:10.1002/pssc.201600083.
[27] Filatov DO, Vrzheshch DV, Tabakov OV, Novikov AS, Belov AI, Antonov IN, Sharkov VV, Koryazhkina MN, Mikhaylov AN, Gorshkov ON, Dubkov AA, Carollo A, Spagnolo B. Noise-induced resistive switching in a memristor based on ZrO2(Y)/Ta2O5 stack. Journal of Statistical Mechanics: Theory and Experiment 2019(12):124026. doi:10.1088/1742-5468/ab5704.
[28] Spagnolo B, Valenti D, Guarcello C, Carollo A, Persano Adorno D, Spezia S, Pizzolato N, Di Paola B. Noise-induced effects in nonlinear relaxation of condensed matter systems. Chaos, Solitons & Fractals 2015(81):412–24. doi:10.1016/j.chaos.2015.07.023.
[29] Spagnolo B, Guarcello C, Magazzù L, Carollo A, Persano Adorno D, Valenti D. Nonlinear Relaxation Phenomena in Metastable Condensed Matter Systems. Entropy 2017;19(1):20. doi:10.3390/e19010020.
[30] Vasil'ev VA, Chernov PS. Mathematical modeling of memristor in the presence of noise. Matem. Mod. 2014;26(1):122–32.
[31] Ielmini D, Ambrogio S, Milo V, Balatti S, Wang Z. Neuromorphic computing with hybrid memristive/CMOS synapses for real-time learning. In: 2016 IEEE International Symposium on Circuits and Systems (ISCAS); 2016. p. 1386–9. doi:10.1109/ISCAS.2016.7527508.
[32] Milo V, et al. Demonstration of hybrid CMOS/RRAM neural networks with spike time/rate-dependent plasticity 2016 IEEE International Electron Devices Meeting (IEDM) San Francisco, CA; 2016. 16.8.1–16.8.4. doi:10.1109/IEDM.2016.7838435.
[33] Bhattacharya T, Parmar V, Suri M. MASTISK: Simulation Framework For Design Exploration Of Neuromorphic Hardware. In: 2018 International Joint Conference on Neural Networks (IJCNN); 2018. p. 1–9. doi:10.1109/IJCNN.2018.8489277.
[34] Ambrogio S, et al. Novel RRAM-enabled 1T1R synapse capable of low-power STDP via burst-mode communication and real-time unsupervised machine learning. In: 2016 IEEE Symposium on VLSI Technology; 2016. p. 1–2. doi:10.1109/VLSIT.2016.7573432.
[35] Wenger C, et al. Inherent Stochastic Learning in CMOS-Integrated HfO2 Arrays for Neuromorphic Computing. IEEE Electron Device Letters 2019;40(4):639–42. doi:10.1109/LED.2019.2900867.