Bull. Nov. Comp. Center, Comp. Science, 39 (2016), 69–75

The memristor crossbar-based WTA neural network

M.S. Tarkov, M.I. Osipov

Abstract. The problems of programming memristor arrays (memristor crossbars) are considered. An estimate for the pulse width to set the desired memristor resistance (memristance) value is obtained. The implementation of the Winner-Take-All (WTA) neural network on the memristor crossbar and the NMOS transistors for binary images recognition is proposed. The proposed WTA network implementation by simulation on the LTspice IV software was approved.

1. Introduction

An artificial neural network, typically, uses a weighting coefficients matrix to represent a set of neurons layer synapses. Accordingly, the computation of the layer neurons activations can be considered as the weighting matrix multiplication by the input layer signals vector. The neural network hardware implementation requires large volume of memory for storing the weights matrix of the neurons layer and is expensive. Solving this problem is simplified by using a device called "memristor" as a memory cell. The memristor was theoretically predicted in 1971 by Leon Chua [1]. First, the memristor physical implementation in 2008 was demonstrated by a Hewlett Packard laboratory as a thin film structure TiO_2 [2]. In Russia the first memristor based on TiO_2 was developed in Tyumen State University [3] in 2012. The memristor has many advantages such as non-volatile storage media, low power consumption, high density integration and excellent scalability. A unique ability to retain traces of the device excitation makes it an ideal means for the implementation of electronic synapses in neural networks [4].

2. Memristor crossbar

The memristor behaves like a synapse: it "remembers" the total electric charge passed through it [5]. The memristor memory can reach very high degree of integration of 100 Gbit/cm^2, several times higher than that based on the flash memory technology [6]. These unique properties make the memristor a promising device for creating massively parallel neuromorphic systems [7–9].

A memristance (the memristor resistance) (Figure 1) can be represented as (cf. [10])

$$M(p) = pR_{\mathrm{on}} + (1-p)R_{\mathrm{off}}, \tag{1}$$

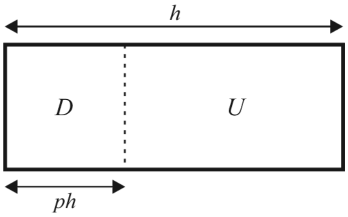


Figure 1. Memristor: D is a low resistance zone, U is a high resistance zone, $p \in [0, 1]$ is the doping front position, h is the total $\mathrm{TiO_2}$ film thickness

where $0 \leq p \leq 1$ is the doping front position relative to the total film thickness h of $\mathrm{TiO_2}$, R_{on} is the memristor minimum resistance, R_{off} is the memristor maximum resistance.

When a voltage V above a certain threshold V_{th} is applied to the memristor, its memristance decreases due to the expansion of the doped band D having a low resistance and reducing the zone U of pure oxide having a high resistance. Accordingly, the memristance increases by applying a voltage V lower than $-V_{\mathrm{th}}$ due to the zone D reduction and the zone U expansion. After the voltage switches off, the current memristance is preserved.

The velocity of the doping front motion is defined as follows:

$$\text{at } V(t) > V_{\mathrm{th}}: \qquad \frac{dp}{dt} = \mu_\nu \frac{R_{\mathrm{on}}}{h^2} \frac{V(t) - V_{\mathrm{th}}}{M(p)}, \tag{2}$$

$$\text{at } V(t) < -V_{\mathrm{th}}: \qquad \frac{dp}{dt} = \mu_\nu \frac{R_{\mathrm{on}}}{h^2} \frac{V(t) + V_{\mathrm{th}}}{M(p)}. \tag{3}$$

Here μ_ν is the average ion mobility and $V(t)$ is the current voltage value on the memristor.

Setting the memristor to a desired level of the memristance M_d depends on the ratio between M_d and the initial memristance value M_0. The memristance adjustment is made by applying to the memristor a constant voltage $V > V_{\mathrm{th}}$ for $M_0 > M_d$ or a voltage $V < -V_{\mathrm{th}}$ for $M_0 < M_d$ due to some time τ. Solving equations (1)–(3) yields the desired time

$$\tau = \begin{cases} \dfrac{M_0^2 - M_d^2}{2k(V - V_{\mathrm{th}})}, & V > V_{\mathrm{th}}, \\[2ex] \dfrac{M_0^2 - M_d^2}{2k(V + V_{\mathrm{th}})}, & V < -V_{\mathrm{th}}, \end{cases} \tag{4}$$

where $k = \mu_\nu \frac{R_{\mathrm{off}}}{h^2}(R_{\mathrm{off}} - R_{\mathrm{on}})$.

Figure 2 shows a layer of neurons (neurons are indicated by triangles) with a matrix of weighting coefficients realized on memristors. The matrix is called a memristor crossbar which is a typical memristor memory structure. It contains a memristor at each intersection of the horizontal and vertical wires. The vertical wires are the neuron layer inputs. The horizontal wires realize the weighted input summations, i.e. calculate the neuron activations. The weighting factors are set by the memristor conductivities located at the intersection of wires.

In this paper, a model of the WTA neural network hardware implementation is proposed. This network implements the crossbar-based associative memory. The network training procedure is implemented in the hardware. The memristor synapse includes the NMOS transistor to control the current through the memristor.

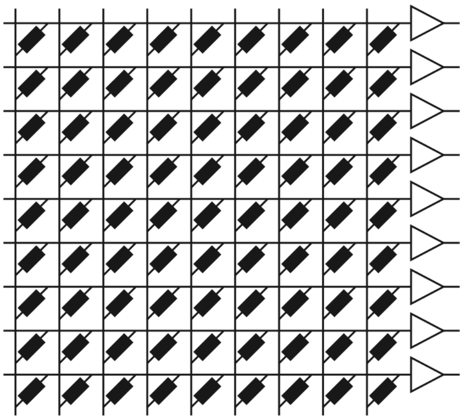


Figure 2. Example of neural layer based on a memristor crossbar

3. WTA neural network

The WTA network functioning is described by equation (5), where w_i is a weight vector of the ith neuron, $i = 1, \dots, P$, f is an activation function, x is the input vector, y_i is an output vector of the ith neuron. We suppose that x_j, $j = 1, \dots, N$, are signals produced by an input pattern, and $x_0 = -1$ is the threshold signal. Then

$$y_i = f(a_i), \quad a_i = \sum_{j=0}^{N} w_{ij} x_j. \tag{5}$$

The input vector x belongs to a class i when $a_i > a_j$, $j = 1, \dots, P$, $j \neq i$ (WTA principle). Let

$$f(a) = \begin{cases} 1, & a > 0, \\ 0, & a \leq 0. \end{cases} \tag{6}$$

Let the neural layer receive a set of pairwise distinct objects $x^i = (x_1^i, \dots, x_N^i)$, $x_j^i = \pm 1$, $i = 1, \dots, P$, i.e. $x^i \neq x^k$ for $i \neq k$. Such objects could be binary images. Let each object x^i contain m components equal to 1 (“white”), and n components of -1 (“black”), $m + n = N$, i.e. objects differ from each other by permutation of white and black pixels (Figure 3).

We associate a weight $w_{ij} = W_{\max}$ with the white pixel and a weight $w_{ij} = W_{\min}$ with the black pixel, where $W_{\max} > W_{\min}$ are specified values. Let $w_{i0} = W_0$, $i = 1, \dots, P$, be threshold weights. The ith neuron wins the competition on the input vector x^i, because

$$a_i = (x^i, w_i) = mW_{\max} - nW_{\min} - W_0, \tag{7}$$

$$a_j = (x^i, w_j) < (m-1)W_{\max} - (n-1)W_{\min} - W_0 < a_i, \quad j \neq i, \tag{8}$$

which corresponds to the WTA principle.

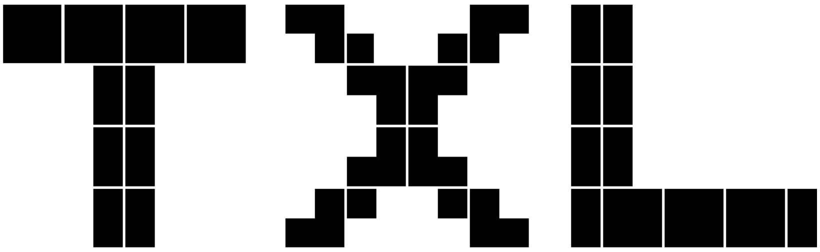
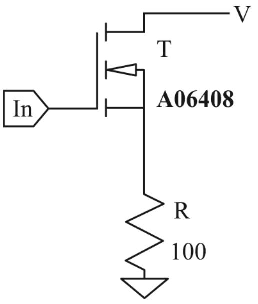


Figure 3. Symbols “T”, “X”, “L”: 8 × 8 pixel pattern

Assuming $W_0 = (m-1)W_{\max} - (n-1)W_{\min}$, we obtain from (5)–(8)

$$a_i > 0, \quad f(a_i) = 1; \qquad a_j < 0, \quad f(a_j) = 0, \quad j \neq i.$$



A mapping of the proposed WTA network onto the memristor crossbar is reduced to the network weights assignment as the crossbar memristors conductivities. The activation function (6) can be implemented using the NMOS transistor (Figure 4) [11].

Figure 4. Neuron activation function realization: the function value is a voltage on the resistor “R”, “In” is the function input, “V” is the supply voltage

4. A hardware model of the WTA network

In the experiments we use the SPICE model of memristor [11–15]. Before setting each memristor, its resistance is evaluated by measuring the current across it when the voltage pulse is applied. After evaluating the memristances, for given voltages $V_j, i = 1, \ldots, 65$, we calculate, in accord with (4), the unipolar pulses durations for setting the memristances to the specified values $M_{\min}$ and $M_{\max}$.

The memristance settings are made successively for the crossbar rows (Figure 5). First, the voltage $hold_1$ is applied to the transistor gates of the first row of synapses which corresponds to the first memorized pattern. In this case, the memristances of other rows do not change because their voltages $hold_2$ and $hold_3$ are zero. Further, the same procedure is realized for all other rows of the crossbar.

The voltage sources $hold_i$, $i = 1, 2, 3$, are used for opening the transistors in synapses, and the voltages V_j, $j = 1, \ldots, 65$, are used for the memristances setting (see Figure 5). During the memristances setting, the crossbar horizontal wires are grounded for removing parasitic currents. After the

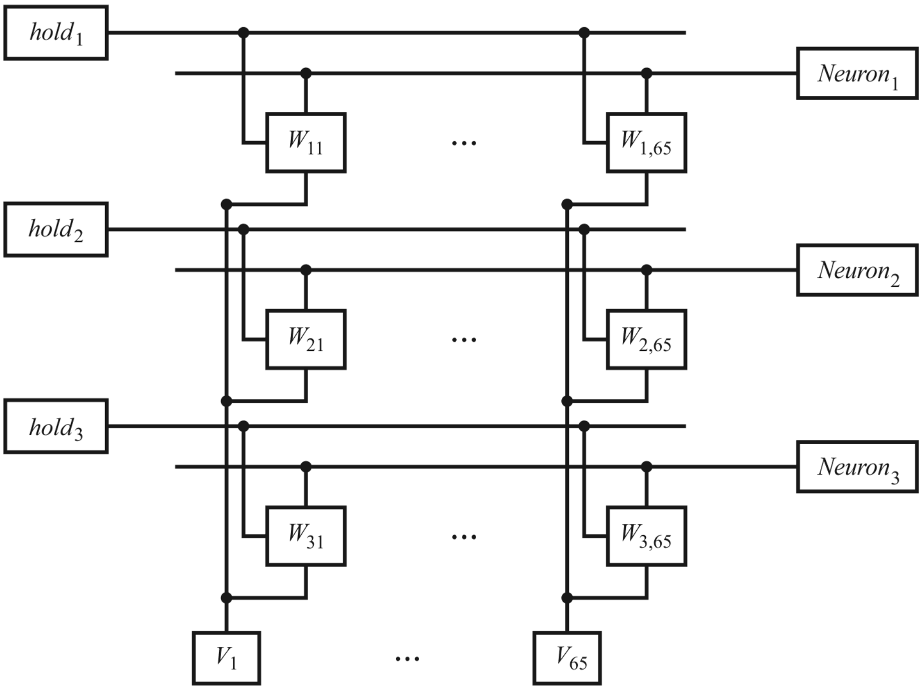


Figure 5. WTA network: W_{ij} are synapses (Figure 6), $Neuron_i$ are the activation functions (see Figure 4), $hold_i$ are the voltages for synapses control, $i = 1, 2, 3$, $j = 1, \dots, 65$; V_j, $j = 1, \dots, 64$, are the input signals, V_{65} is the threshold signal

setting procedure completion the memristor states are saved in a file. These states are read as above.

The synapses W_{ij}, $i = 1, 2, 3$, $j = 1, \dots,$ 64, conductivities are

$$W_{\max} = \frac{1}{M_{\min}} = \frac{1}{3 \cdot 10^3} \text{ ohm}^{-1},$$
$$W_{\min} = \frac{1}{M_{\max}} = \frac{1}{6 \cdot 10^3} \text{ ohm}^{-1}.$$

The threshold $W_{i,65}$, $i = 1, 2, 3$, conductivities are equal to

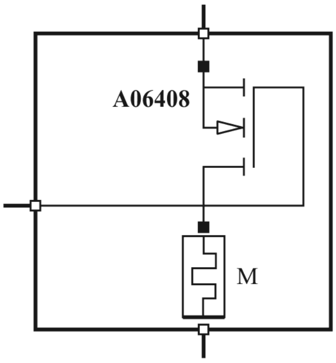


Figure 6. Synapse construction: M is a memristor

$$W_0 = (m-1)W_{\max} - (n-1)W_{\min} = 35W_{\max} - 27W_{\min} = \frac{43}{6 \cdot 10^3} \text{ ohm}^{-1}.$$

White (“1”) and black (“-1”) pixels of the images are simulated by the bipolar pulses (Figures 7a and 7b, respectively). The use of such bipolar signals allows us to save the memristor conductivity unchanged after the signal exposure.

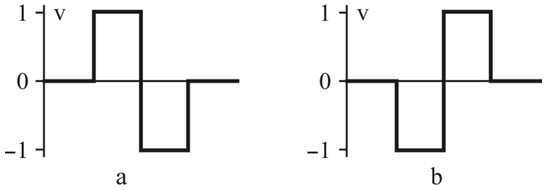


Figure 7. Signals

Each row in Figure 5 forms an adaptive adder that computes the neuron activation (9) using the first half of the bipolar pulses. Neurons 1, 2, and 3 recognize the symbols "L", "T", and "X", respectively. The ith neuron is formed by the synapses W_{ij} (see Figure 6), $j = 1, \ldots, 65$, and the activation function $Neuron_i$ (see Figure 4). The voltages on resistors of the activation functions are considered to be output signals of the neurons. Input signals are given by the source voltages V_j, $i = 1, \ldots, 64$, and V_{65} is the threshold voltage.

In the table, the voltages $3 \cdot 10^{-4}$ volts and $2 \cdot 10^{-9}$ volts correspond, respectively, to values 1 and 0 of the activation function. The table shows that the scheme proposed can be successfully used for image recognition based on the WTA principle.

Output signals of the WTA network (in volts)

Output	L	T	X
y_1	$3 \cdot 10^{-4}$	$2 \cdot 10^{-9}$	$2 \cdot 10^{-9}$
y_2	$2 \cdot 10^{-9}$	$3 \cdot 10^{-4}$	$2 \cdot 10^{-9}$
y_3	$2 \cdot 10^{-9}$	$2 \cdot 10^{-9}$	$3 \cdot 10^{-4}$

5. Conclusion

The memristor is a promising element for the hardware synapses implementation. The memristor array (memristor crossbar) programming problems are considered. The pulse width estimate for setting a desired memristor resistance (memristance) value is obtained. The WTA neural network implementation on the memristor crossbar and the NMOS transistors is proposed for the binary images recognition. The proposed WTA network implementation was approved by the simulation in the LTspice IV software. The results can be used both in mathematical modeling, and in the physical implementation of neural networks with interneuronal connections realized by the memristors.

References

[1] Chua L. Memristor — the missing circuit element // IEEE Transactions on Circuit Theory. — 1971. — Vol. 18. — P. 507–519.

[2] Strukov D.B., Snider G.S., Stewart D.R., Williams R.S. The missing memristor found // Nature. — 2008. — Vol. 453. — P. 80–83.

[3] Tyumen Associative Systems Association (TASO) blog. — http://www.taso.pro/author/admin/ (Usage date: 16.12.2015).

[4] Pershin Y., Di Ventra M. Experimental demonstration of associative memory with memristive neural networks // Neural Networks. — 2010. — Vol. 23, No. 7. — P. 881–886.

[5] Chua L. Resistance switching memories are memristors // Applied Physics A: Materials Science & Processing. — 2011. — Vol. 102, No. 4. — P. 765–783.

[6] Ho Y., Huang G.M., Li P. Nonvolatile memristor memory: device characteristics and design applications // Proc. IEEE Int. Conf. on Computer-Aided Design (ICCAD). — IEEE Publisher, 2009. — P. 485–490.

[7] Jo S.H., Chang T., Ebong I., et al. Nanoscale memristor device as synapse in neuromorphic systems // Nanoletters. — 2010. — Vol. 10, No. 4. — P. 1297–1301.

[8] Kavehei O. Memristive Devices and Circuits for Computing, Memory, and Neuromorphic Applications: PhD thesis. — Australia: The University of Adelaida, 2011.

[9] Lehtonen E. Memristive Computing. — Finland: University of Turku, 2012.

[10] Wu Q., Liu B., Chen Y., Li H., et al. Bio-inspired computing with resistive memories — models, architectures and applications // Proc. 2014 IEEE Int. Symposium on Circuits and Systems (ISCAS). — IEEE Publisher, 2014. — P. 834–837.

[11] Volodin V.Ya. Computer Modeling of Electronic Circuits. — S.-Petersburg: BHV-Peterburg, 2010 (In Russian).

[12] Wagner P. The LTspice2Matlab Function. — http://www.mathworks.com/matlabcen-tral/fileexchange/23394-fast-import-of-compressed-binary-raw-files-created-with-ltspice-circuit-simulator.

[13] Dorran D. The runLTspice Function. — https://dadorran.wordpress.com/tag/run-ltspice-matlab/.

[14] Biolek Z., Biolek D., Biolkova V. SPICE model of memristor with nonlinear dopant drift // Radioengineering. — Vol. 18, No. 2. — P. 210–214.

[15] Falatic M. Memristor simulation with LTspice — a practical example! — http://www.falatic.com/index.php/69.