Level Scaling and Pulse Regulating Methods to Mitigate Cycle-to-cycle Variation in Memristor Based Learning System

Zhiheng Liao

North Dakota State University <https://orcid.org/0000-0002-8225-5044>

Jingyan Fu

North Dakota State University

Jinhui Wang (✉ jwang@southalabama.edu)

University of South Alabama <https://orcid.org/0000-0003-4731-8481>

Article

Keywords: Memristor, cycle-to-cycle variation, neural network, level, pulse regulating, IoT

Posted Date: August 2nd, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-49761/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Level Scaling and Pulse Regulating Methods to Mitigate Cycle-to-cycle Variation in Memristor Based Learning System

Abstract

As a novel non-volatile device, the memristive crossbar array has already delivered many of its promises including low computation complexity, high energy efficiency, and high density for the neuromorphic computing. However, the intrinsic variability of switching behavior has been a major obstacle to their implementation. Here we report a model that experimentally demonstrates the natural stochasticity of cycle-to-cycle variations and quantifies it. In addition, we propose level scaling and pulse regulating methods to mitigate the adverse impact of cycle-to-cycle variations. The relationship of the level of conductance and cycle-to-cycle variation is studied, and experiment results show an optimal number of the levels to mitigate cycle-to-cycle variations in the system. Additionally, the system compresses the number of pulses when the conductance is updated by the pulse stimulus to reduce cycle-to-cycle variations, resulting in the great energy and latency reduction. This work paves the way for the adoption of memristors for more efficient applications for the era of the edge computing and Internet of Things (IoT).

Keywords

Memristor, cycle-to-cycle variation, neural network, level, pulse regulating, IoT

Introduction

The internet of things (IoT) system is a network of devices, sensors, and other items of various functionalities that interact and exchange data electronically¹. Recent years have witnessed significant progress in edge devices and wireless sensor networks, creating unprecedented opportunities to deploy deep learning and artificial intelligence (AI) technologies in IoT, while significantly adding calculation burdens in edges². However, edges consisting of mobile devices and embedded systems usually have limited resources and power, especially when they are used for real-time applications, such resource and power deficiency will results in recognition and prediction accuracy loss in a learning system and even malfunctions in IoT¹⁻⁶.

The conventional complementary metal-oxide-semiconductor transistor (CMOS) technology plateaus the process scaling^{7,8}, which cannot provide satisfied solutions for the emerging edge computing with designated learning systems⁶. Memristors are theoretically postulated by Chua in 1971⁹ and later are physically fabricated by the Hewlett-Packard in 2008¹⁰. The memristor-based crossbar arrays with storage and computing capability show great potential in the neural-network and machine learning applications¹⁰⁻¹⁵. They are characterized with low computational complexity¹⁶, low power consumption¹⁷, fast switching speed¹⁸, high endurance¹⁹, excellent scalability²⁰, and CMOS-compatibility²¹, which are specially appropriate for edge computing in IoT. However,

because of the inherent material property, the intrinsic variation in switching conductance is a major challenge for some applications such as non-volatile memory²². Cycle-to-cycle variation is deviation between target conductance and updated conductance when the same updating signal in different updating cycles is applied in a memristor, even when the initial conductance is the same²³. The memristor-based crossbar arrays suffer from serious cycle-to-cycle variation especially when arrays are used in the neuromorphic computing system that the conductance of memristor needs to be updated innumerable times during the training and testing process²⁴⁻²⁸. In previous works, researchers proposed some solutions including three aspects to mitigate such impact of the cycle-to-cycle variation.

In software-based and algorithm perspective: a conversion algorithm is invented to map arbitrary matrix values appropriately to memristor conductance to reduce computational errors²⁹. The algorithms of the mutual decision between conductance of memristor and Boolean functions are used to tolerate a maximum variation³⁰. A novel off-device neural-network training method is used to improve the performance of the neural network³¹. However, because variation comes from memristor devices - hardware of the neuromorphic computing system, the software-based methods are usually resources-consuming.

In circuit perspective: the smart programming scheme (read the conductance before writing it) and dummy column technologies to eliminate the off-state current are utilized to improve immunity to cycle-to-cycle variations^{32,33}. The experimental result shows the accuracy is improved to 95% from 70%. In addition, a variation-aware training scheme is used to enhance training robustness³⁴. Sophisticated circuits are needed to ensure the quality of conductance switching and either drastically increase the area of circuit and power consumption or bring additional circuit latency.

In device perspective: instead of using a single memristor, the multiple cells technology using several memristors connected in parallel is applied to improve the variation tolerance^{35,36}. But the multiple cells produce area overhead in the system. In addition, the different materials, such as TiO_x as buffer layer³⁷ and $\text{CeO}_2/\text{Ti}/\text{CeO}_2$ tri-layered as active layer³⁸, are proposed and investigated to improve the resistance of ratio between high-resistance state and low-resistance state, enhance the endurance of switching, and reduce the variation of the threshold voltage. Therefore, developing a cycle-to-cycle variation model on memristor is urgent for mitigating that impact so that it is accessible to apply the great potential and advantages of the memristor in practical applications.

Here, we fabricate the $\text{TiO}_2/\text{TiO}_{2-x}$ memristor device and derive a closed model for the cycle-to-cycle variation. Based on this model, we propose and demonstrate level scaling method and pulse regulating methods to mitigate the adverse

impact of the cycle-to-cycle variation. Theoretically, the prediction accuracy is higher if the system utilizes a higher number of the conductance levels^{32,39}. However, this relationship is broken by the cycle-to-cycle variation - sometimes a higher number of the levels yields diminishing accuracy. Thereby, the level scaling method optimizes the level number of conductance. Additionally, the pulse regulating method compresses the number of update pulses to one to mitigate cycle-to-cycle variation when the memristors get update in an IoT

system. In this work, we focus on the relationship of the prediction accuracy and the number of the levels under the cycle-to-cycle variation, and pulse regulating method. And the target of the proposed techniques is to set the appropriate number of the levels and compress number pulses to alleviate and compensate prediction accuracy degradation resulting from the cycle-to-cycle variation, finally optimizing the performance of the neuromorphic computing system in IoT.

Fig. 1 Memristive crossbar arrays and device behavior. **a** An optical image of a wafer with memristive crossbar arrays. **b** Close-up of chip image showing crossbar arrays. **c** Microscope image showing one memristor device. **d** Cross-sectional schematic diagram of the TiO₂/TiO_{2-x} memristor device structure. **e** I-V characteristics from positive and negative voltage sweeping switching (-3 V to 3V) in the memristor. **f** I-V characteristics from consecutive positive voltage pulses sweeps showing a continuous increase in conductance. The width of pulses is 1 ms and the step of voltage is 1.5 mV. **g** Consecutive negative voltage pulses sweeps showing a continuous decrease in conductance. The width of pulses is 1 ms and the step of voltage is 1.5 mV. **h** Hardware implementation of the vector-matrix-multiplication using memristor crossbar. V_i , G_{ij} , and I_j represent the input signal in i^{th} row, the conductance of the memristor in j^{th} column and i^{th} row, and the output current that represent the dot product result of V and G , respectively. **i** Level of conductance with different widths of the updating pulses.

Results

Memristive crossbar array and level of conductance.

We fabricate and test the memristive crossbar arrays in our lab (see Methods for device fabrication and test details). The optical image and geometry of a $\text{TiO}_2/\text{TiO}_{2-x}$ based memristive crossbar arrays used in this work is schematically shown in Fig. 1a. The arrays is composed of 20×20 memristors, as shown in Fig. 1b. Physically, a memristor is a $40 \mu\text{m} \times 40 \mu\text{m}$ two-terminal device formed by two aluminous electrodes sandwiching a thin active layer that is $\text{TiO}_2/\text{TiO}_{2-x}$ material to achieve stable tunable multilevel behavior with a nonlinear current-voltage (IV) relationship, as illustrated in Fig. 1c (see Methods for device fabrication details). Fig. 1d shows the memristor has an $\text{Al}/\text{TiO}_2/\text{TiO}_{2-x}/\text{Al}$ stack in cross-section. In fabrication, a typically memristor presents a very high resistance across its electrodes (an unformed state) and an initial one-time electroforming step is needed²⁷ for multi-level conductance. This can be done by applying voltage or current sweep across the two electrodes until a soft breakdown of the active layer occurs, generating a conductive filament that changes the conductance⁴⁰. This $\text{TiO}_2/\text{TiO}_{2-x}$ memristor displays obvious multilevel behavior as Fig. 1e that shows the current-voltage response of the memristor when the full range voltage sweeps during different cycles. For further investigating this multilevel property, the positive and negative voltage sweeping are separately applied in the same memristor with ten cycles, as shown in Fig. 1f and Fig. 1g. The conductance of the memristor is changed when the voltage achieves the threshold voltage, which is caused by conducting filament formation across the electrodes^{41,42}. Memristive crossbar arrays carry out the vector-matrix multiplication as shown in Fig. 1h. Every row of the crossbar array gets input voltage pulses that are the vector. Each conductance of the device in every cross point composes the matrix. Every column of the crossbar array transmits an output current that is the sum of multiplication by the input signal and conductance in each cross point. To update the conductance of a memristor that has multilevel conductance from the minimum to the maximum, a positive pulse signal is applied to increase the conductance, which is called long-term potentiation (LTP)⁴³. Conversely, long-term depression (LTD) is the process of decreasing the conductance by supplying a negative pulse signal until the conductance gets to the minimum⁴³. Multilevel memristors effectively utilize such multi-value conductance to learn the features of data and realize a neuromorphic computing system^{11,12,28}.

In practice, the width of the pulse signal that is used to update the weight cannot be infinitely narrow and limits the accuracy of the conductance updating. Different widths of the pulses change the different amount of the conductance. Therefore, the widths of different pulses that are used for weight update decide the number of the levels as shown in Fig. 1i. The number of these levels can be expressed qualitatively as equation (1):

$$\lceil (G_{\max} - G_{\min}) / W_{\text{pulse}} \rceil = \text{number of levels} \quad (1)$$

where $G_{\max}$ and $G_{\min}$ are the maximum conductance and the minimum conductance, W_{pulse} is the width of the updating

pulse. Note that although a higher number of the levels gives more precise conductance in the weight update of the memristor, the influence of cycle-to-cycle variation will increase, which is shown in next section.

Cycle-to-cycle variation.

Since the switching mechanism of the memristor conductance is prompted by the applied voltage, a memristor switches its conductance level from one to another when pulse is larger than a threshold voltage for at least the minimum switching time⁴⁴. Simultaneously, cycle-to-cycle variation result in different updated conductance when the same updating signal in different updating cycles is applied in a memristor, even when the initial conductance is the same, as shown in Fig. 2a. For instance, for some given updating pulses, a memristor starting at conductance A and target conductance is B, may end up between C and D, as shown in insert Fig. 2a. Memristors exhibit cycle-to-cycle variation because of the shape of the conductive filament, the oxygen vacancy distribution at and around the filament, and the changing location of the active filament between one cycle to the next. These three mechanisms originate from the coexistence of multiple subfilaments and that the active, current-carrying filament may change from cycle to cycle²³. Thus, cycle-to-cycle variation is a type of inherent randomness associated with the randomness in internal atomic configurations^{26,45,46}. One of the major obstacles for the implementation of redox-based multilevel memristive memory or logic technology is the large cycle-to-cycle variation²³.

Fig. 2b shows the LTP and LTD process with different pulse widths. We can fit these LTP and LTD experimental data with exponential formulas³², as shown in the Fig. 2b fitting curves (Supplementary Figure S1). Because the fitted curve and stochastic behavior of the cycle-to-cycle variation can be approximated with a Gaussian distribution⁴⁷, residual analysis is done by Gaussian distribution fitting after normalization, as illustrated in Fig. 2c (Supplementary Figure S2 and Figure S3). We model cycle-to-cycle variation by Gaussian noise $N(0, \delta)$, where δ is the standard deviation. Our cycle-to-cycle modeling can be defined as

$$\psi_{\text{total}} = N(0, \delta) * \sqrt{n} \quad (2)$$

$$\delta = \alpha * (G_{\max} - G_{\min}) \quad (3)$$

where ψ_{total} is the total cycle-to-cycle variation that is generated for one memristor at one update process, $N(0, \delta)$ is Gaussian noise, n is the number of pulses for this memristor at this update process, α is coefficient that is percentage of difference of the maximum conductance and the minimum conductance, $G_{\max}$ is the maximum conductance, and $G_{\min}$ is the minimum conductance. (Supplementary Note 1)

After Gaussian distribution fitting, we get a distribution of α values with different pulse widths such that the average is 0.03577, as shown in Fig. 2d (Supplementary Table 1). The lines are linear fitting for α values of LTP and LTD. Both slopes are negative. Therefore, it can be concluded that increasing the pulse width does not increase the cycle-to-cycle variation when using the same number of pulses to tune conductance.

Fig. 2 Cycle-to-cycle variation of memristive device. **a** Experimental data of conductance values measured by pulses during 200 potentiation and 200 depression pulses. Cycle-to-cycle variation is the deviation between target conductance and realistic updated conductance by the pulses. **b** The LTP and LTD process with different pulse widths from 520 μ s to 2000 μ s. The black curves are fitted by exponential formula. **c** Residual analysis of fitted Gaussian distribution data after normalization of the deviation from **b** in different pulse widths. **d** extraction of the coefficient that is between 0 to 1 from the standard deviation of the Gaussian distribution fitting from **c** in different pulse widths.

Level scaling method.

In order to evaluate the memristor-based crossbar arrays in the different number of the levels (see Methods for level scaling setting) and to find optimal conductance levels under the cycle-to-cycle variation, the multilayer perceptron platform (MLP platform) is used to emulate the learning classification scenario with Modified National Institute of Standards and Technology (MNIST) handwritten dataset³². We adopt the ANN hardware platform, NeuroSim^{32,48}, to perform handwriting prediction as shown in Fig. 3: Fig. 3a. with the path 1 is for the hardware implementation, Fig. 3b. is for the fully connected networks structure, and Fig. 3c. with the path 1 is for the processing flow chart (see Methods for MLP platform).

To study the relationship between the number of the levels and cycle-to-cycle variation, the different number of the level for the LTP and LTD is set. The ideal circumstances ($\alpha = 0$) with the number of the levels from 10 to 200 and step 10 are set with five algorithms as shown in Fig. 3d-h. When the cycle-to-cycle variation is not involved, with the increasing

number of the level, the accuracy goes up to the high area (bright area) from the low area (dark area), where the highest accuracy appears at the number of the levels = 200 at LTP and LTD (upper right corner). It can be concluded that increasing the number of the levels does increase the prediction accuracy. In bright areas of the figures, the prediction accuracies are around 90% in the lower left corner and higher than 93% in the upper right corner.

As a comparison, realistic cases that are with the cycle-to-cycle variation ($\alpha = 0.03577$) are set to obtain optimization with five algorithms as shown in Fig. 3i-m. When the cycle-to-cycle variation is involved, the values of accuracies are lower than that without cycle-to-cycle variation. Moreover, the bright areas where accuracies are higher than 88% is smaller than the ideal case. Note that, in the top right corner, the accuracies do not go up to the highest with the increasing number of the levels. Even though the boundaries for separating bright and dark areas with different algorithms are different, the boundaries demonstrate the locations of the highest prediction accuracies. Those number of the levels (LTP/LTD)

Fig. 3 Level scaling method for mitigating cycle-to-cycle variation. **a** Hardware implementation of the multilayer perceptron platform. PR represents pulse regulating method. **b** 3-layer fully connected structure of neural network. Input layer, Hidden layer, and Output layer are composed by different neurons. Weight 1 and Weight 2 represent the features of training data. **c** Working flow of the multilayer perceptron platform. Optimized number of the level value and measured cycle-to-cycle variation are set at the circuit parameters configuration step. PR represents pulse regulating method. m equals 8,000 for the path 1 and equals 500 for the path 2. **d-h** Prediction accuracy without cycle-to-cycle variation ($\alpha = 0$) with different LTP and LTD number of the levels (from 10 to 200, step is 10) in 5 algorithms. **i-m** Prediction accuracy with different LTP and LTD number of the level values (from 10 to 200, step is 10) in 5 algorithms under measured cycle-to-cycle variation ($\alpha = 0.03577$).

are 50/40 for the SGD, 60/50 for the Momentum, 60/50 for the AdaGrad, 50/50 for the RMSProp, and 50/40 for the

Adam. Therefore, the best performance of the given memristors based neuromorphic computing system under the related

cycle-to-cycle variation does not occur at the number of the levels = 200 for LTP and LTD as shown in Fig. 3i-m. Level scaling method optimizes the number of the levels so that the system achieves higher prediction accuracy by mitigating the cycle-to-cycle variation.

Pulse regulating method.

During the training process of ANN, the weight change that is calculated by the algorithm is converted to positive/negative pulses that is n in equation (2) to update the conductance of the memristor. According to parameter, n , in equation (2), drastically change in conductance by positive or negative pulses for LTP or LTD process causes more cycle-to-cycle variations in corresponding memristors. Thus, the pulse regulating method is to truncate the number of updating pulse to the one ($n=1$) at each LTP or LTD process as shown in path 2 of Fig. 3a. for the hardware implementation and in path 2 of Fig. 3c. for the processing flow chart. The conventional system originally has writing pulses whose widths are appropriate to tune the conductance of a memristor. Simultaneously, each writing pulse is identical during different update processes. The proposed pulse regulating method, instead of updating the weight by the number of the pulses that are directly converted from the weight change in each iteration, only applies one pulse and keeps the original width of writing pulses,

as shown in golden block (path 2) of Fig. 3a. To verify the effectiveness of the proposed pulse regulating method (see Methods for the pulse regulating strategy), we adopt the ANN hardware platform, NeuroSim^{32,48}, to perform pulse regulating method. The pulse regulating is suitable for all five algorithms that the accuracies are higher than that without the pulse regulating method as shown in Fig. 4a-e. Note that, for evaluating the effect of the pulse regulating method, 100 epochs with 500 images for each epoch are set. The negative impact of the pulse regulating method is reducing the learning speed, which only exists at the several beginning learning epochs and is reflected by the red curves below the blue curves in Fig. 4a-e. Although the learning speed is reduced by the pulse regulating method at the several beginning learning epochs, all prediction accuracies of five algorithms have significant improvement with the pulse regulating method after 100 epochs. In addition, the pulse regulating method effectively produces a smoother convergence of the training process, which reduces the excessive fluctuation of the prediction accuracy. The regressions are carried out by the exponential model to fit the experimental data without and with the pulse regulating method. The Reduced Chi-Sqr values that are represented as χ^2_v with the pulse regulating method are smaller (closer to 1) than that without the pulse regulating method as shown in Fig. 4a-e, which demonstrates that the

a - e

Fig. 4 Pulse regulating method for mitigating cycle-to-cycle variation. **a-e** Prediction accuracy with/without the pulse regulating method in 5 algorithms. 50 and 200 are the number of the levels in different evaluations. χ^2_v is the Reduced Chi-Sqr values with analyzing data. PR represents pulse regulating method. Each curve includes 100 epochs and each epoch includes 500 images.

Table 1 Energy and latency evaluation with and without the pulse regulating method.

Algorithm	Energy (mJ)			Latency (Min)		
	With PR	Without PR	Saved (%)	With PR	Without PR	Saved (%)
SGD	0.313	0.361	13.310	15.059	17.728	15.057
Momentum	0.501	0.575	12.888	23.912	32.340	26.062
AdaGrad	0.642	0.671	4.233	31.333	37.290	15.974
RMSProp	1.451	1.729	16.104	75.962	105.288	27.854
Adam	1.072	1.197	10.394	54.422	68.703	20.787

PR represents pulse regulating method. Each evaluation includes 100 epochs and each epoch includes 500 images with 50 levels of conductance. 32nm technology complementary metal–oxide–semiconductor (CMOS) part is evaluated in this work. For the process of weight update, the voltage of the LTP and LTD is 3.2 V, 2.8 V, respectively. The pulse width of weight update is 600 μ s for both LTP and LTD.

fluctuation of the prediction accuracy is reduced by the pulse regulating method.

Furthermore, because the updating pulses are regulated to one in each iteration, the number of updating pulses has been significantly saved for 100 epochs, taking RMSProp as an example, which effectively saves the energy consumption up to 16.104% and reduces the latency up to 27.854% as shown in Table 1. Every iteration has the designated reading latency since the process of a vector-matrix multiplication is executed using a parallel reading strategy. However, the system updates its weight row by row, which indicates a parallel writing strategy cannot be implemented for all rows at the same time, otherwise, the system will have unacceptable area overhead⁴⁹. Each row's writing latency is determined by the maximum number of writing pulses as a critical path. Thereby, the main latency for crossbar arrays is writing latency that strongly depends on the maximum update pulses of each row. With the pulse regulating method, the maximum number of the writing pulses is one, which reduces the latency of system. Pulse regulating method optimizes the prediction accuracy, improves the energy efficiency, and reduces the system latency by mitigating the cycle-to-cycle variation.

Discussion

For a given memristive crossbar array, the distribution of cycle-to-cycle variation can be modeled by equation (2). At the same time, according to Fig. 1i, a lower number of the levels means larger conductance change between two consecutive pulses, and the system uses wider and fewer updating pulses for the same weight change that is calculated through any machine learning algorithm. According to experiment results and equation (2), the wider pulse does not increase the cycle-to-cycle variation and fewer updating pulses corresponds to a smaller n , which reduces the cycle-to-cycle variation. Therefore, level scaling is an effective method to mitigate cycle-to-cycle variation. Note that, an extremely low number of levels will influence the accuracy of the conductance, which means some desired values of conductance cannot be achieved as shown in Fig. 1i, so reducing the precision

of the system. This influence also is reflected by the low prediction accuracy as shown in the low (dark) number of the levels area of Fig. 3d-m. Thereby, for multilevel memristive crossbar arrays that are used in machine learning systems, the highest prediction accuracy of the system occurs when the memristor uses an optimized number of levels rather than the highest number of levels.

In further comparison of Fig. 3d-h and Fig. 3i-m, some accuracies with cycle-to-cycle variations and with certain LTP/LTD levels are higher than that without cycle-to-cycle variation (ideal circumstance) (Supplementary Figure S4). This is because only integer number of pulses are generated in circuit. As for the mechanism of the updating process, the amount of conductance that is increased or decreased will be calculated by the algorithm, then the accurate number of pulses is gotten accordingly. However, in the circuit level (hardware), only the integer number of pulses are available to update the conductance. Hence, the truncation function for an integer number of pulses is employed for hardware implementation. The updated weight gets the deviation by an integer number of pulses. But when the cycle-to-cycle variation is involved in every weight update, in some cases, they make the updated weight achieve closer to the accurate weight that algorithm requires and then the system gets even higher prediction accuracy (Supplementary Figures S5-S7). Thereby, with the truncation function, the cycle-to-cycle variation compensates the decrease of the accuracy by truncation of number of update pulses. That is why some accuracies with cycle-to-cycle variation and with certain LTP/LTD levels in Fig. 3i-m. are higher than that without cycle-to-cycle variation in Fig. 3d-h.

According to the mechanism of the cycle-to-cycle variation, the pulse regulating method efficiently reduces the cycle-to-cycle variation by compressing the number of update pulse to one with n parameter in equation (2). For every updating, the cycle-to-cycle variation is limited with one pulse's impact, which minimizes the cycle-to-cycle variation for the system. Note that, the prediction accuracies have significant improvement with the pulse regulating method as shown in Fig. 4a-e. The reasons are two aspects that includes the pulse

regulating method minimizes the cycle-to-cycle variation and each update step uses at most one pulse to tune conductance. One pulse to tune conductance means smaller steps is achieved in the direction of convergence, while a big step will make the learning jump over minimum point of weight⁵⁰. What's more, energy consumption and system latency are correspondingly reduced when the pulse regulating method is adopted in the system by compressing the number of update pulse to one.

In summary, we propose the level scaling method and the pulse regulating method that are simple and feasible universal methods to effectively mitigate the impact of cycle-to-cycle variation. Under cycle-to-cycle variation, the prediction accuracy in the maximum number of the levels is not optimal for the real device. As for different materials based multilevel memristors, using the same analysis method, the level scaling method can be used to optimize the neuromorphic computing system through selecting appropriately the number of the levels. Similarly, the pulse regulating method mitigates the impact of cycle-to-cycle variation by compressing the number of updating pulses to one and improves energy efficiency and system timing. Furthermore, both methods can be implemented at edge computing, which paves the way for the adoption of memristors for more efficient applications for the era of the IoT.

Methods

Device fabrication. We used Si wafers that have 100 nm thermally grown SiO₂ on top as the substrates. For the 40μm x 40μm memristive device, the bottom electrodes were patterned by ultraviolet photolithography. After that, a 100 nm thick Al bottom electrode was deposited in a Kurt Leaker CMS-18 Sputterer, followed by a lift-off process in acetone. A 100 nm-TiO₂/100 nm-TiO_{2-x} active layer was prepared by sputtering from Ti target (power for TiO₂/TiO_{2-x}: 262 W). Top electrodes were defined by a photolithography step, deposition of a 100 nm Al using sputtering (650 W) and lift-off. The exposure of bottom electrodes was done by etching the active layer through HF.

Electrical characterization. The I-V characteristics from positive and negative voltage sweeping were carried out using a Keysight B1500a semiconductor parameter analyzer in a voltage-sweep and voltage-pulse mode. The wafer was set on the Micromanipulator probe station and the pads were contacted by probe tips (Supplementary Figure S8).

Dataset. For the MNIST dataset, feature vectors are the unrolled grayscale pixel values of the handwritten digital two-dimensional images. The original images are 28 pixels by 28 pixels. Since the edges of the images are not the most informative, one handwritten digit is cropped into 20 x 20 to match the size of the input layer in the platform.

MLP platform. The multilayer perceptron platform⁴⁸ (MLP platform) is used to emulate the level scaling method and the pulse regulating method. The crossbar array architecture with memristors had been proposed for on-chip implementation of weighted sum and weight update in the training process of learning algorithms⁵¹. This platform contains a three-layer with 400 neurons for input layer, 100 neurons for hidden layer, and 10 neurons for output layer, as shown

in Fig. 3b. The perceptron neural network is simulated that bases on memristive crossbar arrays that refers to a special subset of the memristor that can tune the conductance by voltage pulse stimulus. The desired weight update for each layer is calculated in software⁵², then applied to the crossbar by the system as illustrated in Fig. 3a for the hardware implementation block diagram, and in Fig. 3c for the processing flow chart. For the level scaling method, each evaluation trains 125 epochs, and every epoch randomly selects 8,000 images from 60,000 training images. For the pulse regulating method, each evaluation trains 100 epochs, and every epoch randomly selects 500 images from 60,000 training images. A different set of 10,000 images are included in the testing dataset. Note that, the networks will continually learn the feature of an input data after the last epoch since this platform is online learning networks⁴⁸. In this platform, parameters of memristor come from the measurement results of our fabricated devices. In summary, this MLP platform is a standalone functional platform that is able to evaluate the learning accuracy and device-level performance during the learning process.

Level scaling setting. A number of the levels is set in the circuit parameter configuration step according to the working flow of Fig. 3c. That means there is a certain number of the levels that conductance of memristor can be obtained between the maximum and the minimum conductance. The number of the levels will map to the width of the pulse that is generated from the pulse generator in hardware implementation. The higher number of the levels corresponding to the narrower pulses. Theoretically, the system can achieve higher precision of weight for the desired value of conductance. Simultaneously, however, a higher number of the levels introduces larger cycle-to-cycle variation when the system updates the conductance of memristor arrays, because the pulse generator produces more pulses to tune the conductance when the algorithm calculates the same Δweight with that system has a lower number of the levels. Therefore, the level scaling method is applying to appropriately reduce the number of the levels. In this work, the number of the levels is a parameter and set from 10 to 200 and the step is 10.

Pulse regulating strategy. One multiplexer is used to compress the value of the Δweight signal for generating one updating pulse whenever a conductance of a memristor needs to tune according to the hardware implementation diagram in path 2 (golden block) of Fig. 3a. The pulse regulating method only applies one pulse and keeps the original width of writing pulses. The decoder gets a signal from an arithmetic logic unit (ALU) for selecting one row to update. At the same time, the registers get the values of Δweight that are calculated by an ALU. Then these values are transmitted to multiplexers as control signals. Multiplexers select one writing pulse that comes from a pulse generator as output when control signals are enabled. The enabled signal means that the corresponding memristor needs to be updated and that corresponding Δweight value is greater than or equals weight change by one pulse. In this way, the pulse regulating method directly affects every weight update and minimizes the number of pulses and then avoids cycle-to-cycle variation as much as possible.

Data availability. The data that support the findings of this study are available from the corresponding author upon request.

References

1. Weber, R. H. & Weber, R. *Internet of Things*, Vol. 12. (Springer, 2010).

2. Fayyazi, A., Ansari, M., Kamal, M., Afzali-Kusha, A. & Pedram, M. An ultra low-power memristive neuromorphic circuit for internet of things smart sensors. *IEEE Internet Things J.* 5, 1011-1022 (2018).
3. Deng, Y. Deep learning on mobile devices: a review. In *Mobile Multimedia/Image Processing, Security, and Applications 2019*. 109930A (International Society for Optics and Photonics).
4. Lane, N. D. et al. Deepix: A software accelerator for low-power deep learning inference on mobile devices. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 1-12 (IEEE, Vienna, Austria, 2016).
5. Wang, J. et al. Not Just Privacy: Improving Performance of Private Deep Learning in Mobile Cloud. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2407-2416 (ACM, London, United Kingdom, 2018).
6. Gubbi, J., Buyya, R., Marusic, S. & Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* 29, 1645-1660 (2013).
7. Williams, R. S. What's Next? [The end of Moore's law]. *Comput. Sci. Eng.* 19, 7-13 (2017).
8. Waldrop, M. M. The chips are down for Moore's law. *Nature News* 530, 144 (2016).
9. Chua, L. Memristor-the missing circuit element. *IEEE Trans. Circuit Theory* 18, 507-519 (1971).
10. Strukov, D. B., Snider, G. S., Stewart, D. R. & Williams, R. S. The missing memristor found. *Nature* 453, 80-83, doi:10.1038/nature06932 (2008).
11. Wang, Z. et al. Fully memristive neural networks for pattern classification with unsupervised learning. *Nat. Electron* 1, 137 (2018).
12. Zidan, M. A., Strachan, J. P. & Lu, W. D. The future of electronics based on memristive systems. *Nat. Electron* 1, 22 (2018).
13. Xia, L. et al. Switched by input: Power efficient structure for RRAM-based convolutional neural network. In *2016 53rd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 1-6 (IEEE, Austin, TX, USA, 2016).
14. Fu, J., Liao, Z., Gong, N. & Wang, J. Mitigating Nonlinear Effect of Memristive Synaptic Device for Neuromorphic Computing. *IEEE Trans. Emerg. Sel. Topics Circuits Syst.* (2019).
15. Fu, J., Liao, Z. & Wang, J. Memristor-Based Neuromorphic Hardware Improvement for Privacy-Preserving ANN. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* (2019).
16. Kvatinsky, S., Friedman, E. G., Kolodny, A. & Weiser, U. C. TEAM: Threshold adaptive memristor model. *IEEE Trans. Circuits Syst. I, Reg. Papers* 60, 211-221 (2012).
17. Pickett, M. D. & Williams, R. S. Sub-100 fJ and sub-nanosecond thermally driven threshold switching in niobium oxide crosspoint nanodevices. *Nanotechnology* 23, 215202 (2012).
18. Torrezan, A. C., Strachan, J. P., Medeiros-Ribeiro, G. & Williams, R. S. Sub-nanosecond switching of a tantalum oxide memristor. *Nanotechnology* 22, 485203 (2011).
19. Lee, M.-J. et al. A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta₂O₅-x/TaO₂-x bilayer structures. *Nat. Mater.* 10, 625 (2011).
20. Pi, S., Lin, P. & Xia, Q. Cross point arrays of 8 nm× 8 nm memristive devices fabricated with nanoimprint lithography. *J. Vac. Sci. Technol. B, Nanotechnology and Microelectronics: Materials, Processing, Measurement, and Phenomena* 31, 06FA02 (2013).
21. Xia, Q. et al. Memristor- CMOS hybrid integrated circuits for reconfigurable logic. *Nano lett.* 9, 3640-3645 (2009).
22. Guan, X., Yu, S. & Wong, H.-S. P. On the switching parameter variation of metal-oxide RRAM—Part I: Physical modeling and simulation methodology. *IEEE Trans. Electron Devices* 59, 1172-1182 (2012).
23. Baeumer, C. et al. Subfilamentary networks cause cycle-to-cycle variability in memristive devices. *ACS Nano* 11, 6921-6929 (2017).
24. Zhu, R. et al. A Versatile and Accurate Compact Model of Memristor With Equivalent Resistor Topology. *IEEE Electron Device Lett.* 38, 1367-1370, doi:10.1109/led.2017.2736006 (2017).
25. Choi, S., Sheridan, P. & Lu, W. D. Data Clustering using Memristor Networks. *Sci. Rep.* 5, 10492, doi:10.1038/srep10492 (2015).
26. Lee, J.-H., Lim, D.-H., Jeong, H., Ma, H. & Shi, L. Exploring Cycle-to-Cycle and Device-to-Device Variation Tolerance in MLC Storage-Based Neural Network Training. *IEEE Trans. Electron Devices* 66, 2172-2178 (2019).
27. Lastras-Montañó, M. A. & Cheng, K.-T. Resistive random-access memory based on ratioed memristors. *Nat. Electron* 1, 466-472, doi:10.1038/s41928-018-0115-z (2018).
28. Kim, S., Lim, M., Kim, Y., Kim, H.-D. & Choi, S.-J. Impact of synaptic device variations on pattern recognition accuracy in a hardware neural network. *Sci. Rep.* 8, 2638 (2018).
29. Hu, M. et al. Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication. In *Proceedings of the 53rd annual design automation conference (DAC)*, 19 (ACM, Austin, TX, USA, 2016).
30. Rajendran, J., Maenm, H., Karri, R. & Rose, G. S. An approach to tolerate process related variations in memristor-based applications. In *2011 24th International Conference on VLSI Design (VLSI Design)*, 18-23 (IEEE, Chennai, India, 2011).
31. Chen, L. et al. Accelerator-friendly neural-network training: Learning variations and defects in RRAM crossbar. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 19-24. (IEEE, Swiss Tech, Lausanne, Switzerland, 2017).
32. Chen, P.-Y. et al. Mitigating effects of non-ideal synaptic device characteristics for on-chip learning. In *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 194-199 (IEEE Press, Austin, TX, USA, 2015).
33. Kim, K. M. et al. Voltage divider effect for the improvement of variability and endurance of TaO_x memristor. *Sci. Rep.* 6, 20085 (2016).
34. Liu, B. et al. Vortex: variation-aware training for memristor x-bar. In *Proceedings of the 52nd Annual Design Automation Conference (DAC)*, 15 (ACM, San Francisco, CA, USA, 2015).
35. Rajendran, J., Karri, R. & Rose, G. S. Improving tolerance to variations in memristor-based applications using parallel memristors. *IEEE Trans. Comput.* 64, 733-746 (2014).
36. Rajendran, J., Karri, R. & Rose, G. S. Parallel memristors: Improving variation tolerance in memristive digital circuits. In *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, 2241-2244 (IEEE, Rio de Janeiro, Brazil, 2011).
37. Li, Y. et al. Improvement of resistive switching characteristics in ZrO₂ film by embedding a thin TiO_x layer. *Nanotechnology* 22, 254028 (2011).
38. Rana, A. M. et al. Endurance and cycle-to-cycle uniformity improvement in tri-layered CeO₂/Ti/CeO₂ resistive switching devices by changing top electrode material. *Sci. Rep.* 7, 39539 (2017).
39. Tang, T. et al. Spiking neural network with rram: Can we use it for real-world application? In *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 860-865 (IEEE, Grenoble, France, 2015).

40. Switching, R. From Fundamentals of Nanoionic Redox Processes to Memristive Device Applications. PhD diss., ed. D. Ielmini and R. Waser, John Wiley & Sons, 2015 Search PubMed, 2015.
41. Yang, Y. et al. Observation of conducting filament growth in nanoscale resistive memories. *Nat. Commun.* 3, 732 (2012).
42. Chang, T. et al. Synaptic behaviors and modeling of a metal oxide memristive device. *Appl. Phys. A-Mater.* 102, 857-863 (2011).
43. Williamson, A. et al. Synaptic behavior and STDP of asymmetric nanoscale memristors in biohybrid systems. *Nanoscale* 5, 7297-7303 (2013).
44. Uddin, M., Hasan, M. S. & Rose, G. S. On the Theoretical Analysis of Memristor based True Random Number Generator. In *Proceedings of the 2019 on Great Lakes Symposium on VLSI*, 21-26 (ACM, Tysons Corner, VA, USA, 2019).
45. Tuma, T., Pantazi, A., Le Gallo, M., Sebastian, A. & Eleftheriou, E. Stochastic phase-change neurons. *Nat. Nanotechnol.* 11, 693 (2016).
46. Jiang, H. et al. A novel true random number generator based on a stochastic diffusive memristor. *Nat. Commun.* 8, 882, doi:10.1038/s41467-017-00869-x (2017).
47. Ye, C. et al. Physical mechanism and performance factors of metal oxide based resistive switching memory: a review. *J. Mater. Sci. Technol.* 32, 1-11 (2016).
48. Chen, P.-Y., Peng, X. & Yu, S. NeuroSim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures. In *2017 IEEE International Electron Devices Meeting (IEDM)*, 6.1. 1-6.1. 4 (IEEE, San Francisco, CA, USA, 2017).
49. Gao, L. et al. Fully parallel write/read in resistive synaptic array for accelerating on-chip learning. *Nanotechnology* 26, 455204 (2015).
50. Buduma, N. & Locascio, N. Fundamentals of deep learning: Designing next-generation machine intelligence algorithms. ("O'Reilly Media, Inc.", 2017).
51. Chen, P.-Y., Peng, X. & Yu, S. NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* 37, 3067-3080 (2018).
52. Ponti, M. A., Ribeiro, L. S. F., Nazare, T. S., Bui, T. & Collo-mosse, J. Everything you wanted to know about deep learning for computer vision but were afraid to ask. In *2017 30th SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T)*. 17-41 (IEEE, 2017).

Figures

Figure 1

Memristive crossbar arrays and device behavior. a An optical image of a wafer with memristive crossbar arrays. b Close-up of chip image showing crossbar arrays. c Microscope image showing one memristor device. d Cross-sectional schematic diagram of the TiO₂/TiO_{2-x} memristor device structure. e I-V characteristics from positive and negative voltage sweeping switching (-3 V to 3V) in the memristor. f I-V characteristics from consecutive positive voltage pulses sweeps showing a continuous increase in conductance. The width of pulses is 1 ms and the step of voltage is 1.5 mV. g Consecutive negative voltage pulses sweeps showing a continuous decrease in conductance. The width of pulses is 1 ms and the step of voltage is 1.5 mV. h Hardware implementation of the vector-matrix-multiplication using

memristor crossbar. V_i , G_{ij} , and I_j represent the input signal in i th row, the conductance of the memristor in j th column and i th row, and the output current that represent the dot product result of V and G , respectively. i Level of conductance with different widths of the updating pulses.

Figure 2

Cycle-to-cycle variation of memristive device. a Experimental data of conductance values measured by pulses during 200 potentiation and 200 depression pulses. Cycle-to-cycle variation is the deviation between target conductance and realistic updated conductance by the pulses. b The LTP and LTD process with different pulse widths from 520 μ s to 2000 μ s. The black curves are fitted by exponential formula. c Residual analysis of fitted Gaussian distribution data after normalization of the deviation from b in different pulse widths. d extraction of the coefficient that is between 0 to 1 from the standard deviation of the Gaussian distribution fitting from c in different pulse widths.

Figure 3

Level scaling method for mitigating cycle-to-cycle variation. a Hardware implementation of the multilayer perceptron platform. PR represents pulse regulating method. b 3-layer fully connected structure of neural network. Input layer, Hidden layer, and Output layer are composed by different neurons. Weight 1 and Weight 2 represent the features of training data. c Working flow of the multilayer perceptron platform. Optimized number of the level value and measured cycle-to-cycle variation are set at the circuit parameters con-figuration step. PR represents pulse regulating method. m equals 8,000 for the path 1 and equals 500 for the path 2. d-h Prediction accuracy without cycle-to-cycle variation ($\alpha = 0$) with different LTP and LTD number of the levels (from 10 to 200, step is 10) in 5 algorithms. i-m Prediction accuracy with different LTP and LTD number of the level values (from 10 to 200, step is 10) in 5 algorithms under measured cycle-to-cycle variation ($\alpha = 0.03577$).

Figure 4

(A-E) Pulse regulating method for mitigating cycle-to-cycle variation. a-e Prediction accuracy with/without the pulse regulating method in 5 algorithms. 50 and 200 are the number of the levels in different evaluations. χ^2 is the Reduced Chi-Sqr values with analyzing data. PR represents pulse regulating method. Each curve includes 100 epochs and each epoch includes 500 images.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SupplementaryFile.docx](#)