Ameliorate Performance of Memristor-Based ANNs in Edge Computing

Zhiheng Liao, Jingyan Fu, *Student Member, IEEE*, and Jinhui Wang, *Senior Member, IEEE*

Abstract—Energy efficiency and delay time in the Internet of Things (IoT) system are becoming increasingly significant, especially for the emerging memristor-based crossbar arrays for smart edge computing. This article aims to find a solution for increasing energy efficiency and reducing the delay time, thereby improving the performance of ANNs in edge computing systems. The Number of Pulses Compression (NPC) method is proposed to optimize pulse distribution, energy consumption, and latency by compressing the number of pulses in every weight update step. The NPC method is implemented and verified in a memristor-based hardware simulator based on the MNIST and CIFAR-10 dataset under different circumstances of variations, failure rates, aging effects, architectures, and algorithms. The experimental results show that the NPC method can not only alleviate the uneven distribution of writing pulses but also save the writing energy of the crossbar array by 7.7–26.9 percent and reduce the writing latency by 30.0–50.0 percent. Additionally, the timing regularity of the system is enhanced by the NPC method.

Index Terms—Artificial neural networks (ANNs), memristor, weight update, energy consumption, latency, compression, edge computing, Internet of Things (IoT)

1 Introduction

Under the explosive development of the internet of things (IoT), the edge computing is facing unprecedented challenges including the limited energy budget, large system latency, and shortage of storage and computing resource. Especially for the edge AI (Artificial Intelligence), the required real-time response and online-learning in various scenarios are translated to urgent demands on high-speed and power-efficiency hardware components [1], [2].

As complementary metal-oxide semiconductor transistor (CMOS) technology approaches the end of process scaling, the issues of the energy consumption and speed impact the development of the IoT [3], [4]. Memristor device is a promising candidate to complement and/or replace traditional CMOS (at least in some applications) due to advantages including read/write latencies in the order of 1's to 100's of nanoseconds, and energy dissipation of few pJ per bit [5], [6], [7]. Memristors were theoretically postulated by Chua in 1971 [8] and later were physically manufactured by Hewlett-Packard in 2008 [9]. A memristor is a device with three simple layers that can not only achieve desirable device properties such as sub-10 nm feature sizes, sub-nanosecond switching speed long write-erase endurance, and nano amperes programming energy, but also exploit multilevel conductance states by external incentive [8], [9], [10], [11], [12]. Because of these metrics, memristors are suitable for biologically inspired computing [13], [14] - neuromorphic computing, for example, where programmable memristive crossbar arrays have been considered as a critical enabler to achieve a small area footprint and high-density structure in different types of neural networks in the edge AI, such as convolutional neural network (CNN) and deep learning neural network (DNN) [10], [11], [12]. Memristive crossbar arrays could break the bottleneck of the speed and energy efficiency in vector-matrix multiplication which is very resource-consuming and satisfy the high desire for IoT applications by achieving orders-of-magnitude improvement in energy efficiency and performance [1], [2], [3], [10], [11], [12].

Similar to CMOS circuits, the high-performance functionality of a memristive crossbar array that is utilized in neural network translates into high power densities, high operating temperatures, and low reliability [10], [12], [15], [16]. If the memristive crossbar array runs at the edge for an IoT system, overpower and overheat will reduce the effectiveness and lifespan of components. Furthermore, learning algorithms, usually routinely considering accuracy nowadays, rarely pay attention to energy efficiency and latency. In fact, uneven weight updates for different memristors caused by algorithms inevitably leads to local overlarge energy consumption. Also, the writing process of a memristor generates much more energy than the reading process. Therefore, most of the energy during the training process of a memristive crossbar array comes from the writing for the weight update [17]. Consequently, the energy consumption becomes a significant contributor to decline system reliability, deteriorates memristors' retention and endurance, and causes severe timing uncertainty [18], [19], [20]. Therefore, investigating the weight update patterns and the related energy consumption in the memristor arrays is critical to ensure high performance in the edge computing.

- *Zhiheng Liao and Jingyan Fu are with the Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND 58102 USA. E-mail: zhiheng.liao@ndsu.edu, jingyan.fu@ ndsu.edu.*
- *Jinhui Wang is with the Department of Electrical and Computer Engineering, University of South Alabama, Mobile, AL 36688 USA. E-mail: jwang@southalabama.edu.*

Manuscript received 1 Sept. 2020; revised 26 Apr. 2021; accepted 1 May 2021. Date of publication 19 May 2021; date of current version 9 July 2021. (Corresponding author: Jinhui Wang.) Recommended for acceptance by A. Rahimi, L. Benini, S. Benatti, and T. Jang. Digital Object Identifier no. 10.1109/TC.2021.3081985

Recently, researchers proposed several techniques to improve energy efficiency and system latency from different levels. Dual-element memristors are used to achieve low-power memory design [21]. A memristor-based predictor is designed to reduce the energy consumption comparing to the digital counterpart [22]. The hybrid crossbar architecture for improving the performance of energy efficiency and system latency is studied in [23]. In [24], the error correcting code is proposed to relax the Bit Error Rate requirement of a single memory to improve the write energy consumption and latency for both the CMOS based and cross-point based memristor resistive random-access memory (ReRAM) designs.

In this paper, we focus on improving the energy efficiency and system latency for the memristor arrays in multilayer perceptron (MLP) and convolutional neural network (CNN) by compressing pulses number for weight updates. An overview of the pulse distribution of the memristor-based crossbar array and their relation to the energy consumption is provided. We propose a method that reduces the energy consumption during the memristors' weight updating process, reduces the writing latency, and improves timing regularity, thereby enhancing the performance of the edge computing in IoT systems. Specifically, this paper makes the following contributions:

1) A low cost method to reduce the writing energy in the memristor-based online learning for the edge computing: An effective hardware-based method is proposed that we call Number of Pulses Compression (NPC) with low circuit overhead. The proposed method will compress the number of pulses during updating the conductance of a memristor.
2) A mechanism for smoothing learning process, thereby avoiding extra energy consumption and writing latency in the edge computing: By applying the NPC method, the weight updating fluctuation is reduced. The writing latency that is decided by the maximum number of the pulses is drastically reduced by the compressing mechanism of the NPC. Additionally, the timing regularity of the system is improved.
3) The nonlinearity and variations analysis: To investigate the effectiveness of the NPC method based on the actual memristive devices, the nonlinear property of memristor and four variations that include device-to-device variation, cycle-to-cycle variation, maximum/minimum conductance variation, and on/off ratio variation are considered in experiments.
4) Thorough evaluation: We evaluate the proposed method based on the standard image classification tasks [25] and the hardware-based online learning simulator, NeuroSim+ [26], enabling a model under three failure rates, different network architecture, and aging effect.

2 Background

2.1 ANNs and Weight Update

ANNs transform inputs to desired outputs by feedforward neural networks that comprise many layers. Each node in the network is a neuron that takes a weighted sum of the outputs of the prior layer, and then transmits the sum to the next layer. The main work of a training ANN is to learn the feature that is represented by weight from a large volume of training data. During the training process, weight is updated in each iteration by weight change that is calculated based on an algorithm.

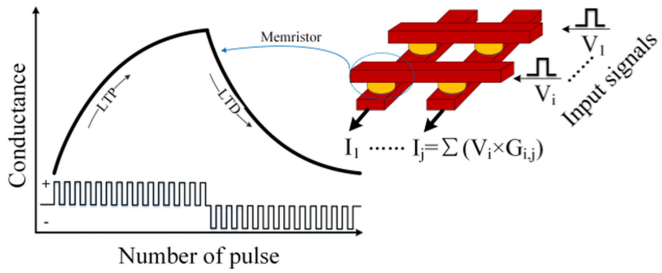


Fig. 1. Hardware implementation of neural networks using memristor crossbar. V_i, G_{ij}, and I_j represent the input signal in ith row, the conductance of the memristor in jth column and ith row, and the output current that represent the dot product result of V and G, respectively. Conductance of memristor is regulated with the number of pulses. The LTP/LTD is triggered by positive/negative pulses [28].

2.2 Memristive Crossbar Array

The property of the memristive device enables it to implement ANNs [9], [27]. Memristive crossbar array carries out the vector-matrix multiplication and learns the feature of data by updating each memristor's conductance [13], as shown in Fig. 1. Every row gets input voltage signal which is the vector. Each conductance of a memristor in every cross point composes the matrix. Every column transmits an output current which is the sum of the product by the input signal and conductance in the same column. Due to the efficient implementations on vector-matrix multiplication operations, memristive crossbar array is suitable and efficient hardware for ANNs in edges.

The conductance of a memristor with multilevel as shown in Fig. 1 [28], is increased by supplying a positive pulse until the conductance gets to the maximum. This increasing process is long-term potentiation (LTP). Conversely, long-term depression (LTD) is the process of decreasing the conductance by supplying a negative pulse until the conductance gets to the minimum. Simultaneously, the memristor can store the information even when the power supply is turned off, because of its non-volatile property. The memristor, therefore, is a device that combines learning, storage, and computing, making it essential in hardware for ANNs, especially for edges with the limited resource, but real-time response in IoT systems.

3 Methodology

As for the conventional method to update the conductance of a memristor, according to the value of weight change that is calculated by the algorithm, the control circuit will generate corresponding signals to control the pulse generator and update time for producing positive/negative pulses and tuning the conductance of the memristor. Note that a memristor has the characteristics of the finite conductance states, specific switching time, and fixed threshold voltage. Thus, some small delta weights cannot be converted to pulses. Therefore, the precision of conductance during the update process is limited. Besides, the drastic change occurs in conductance by these positive or negative pulses at the beginning stage of the training process, which causes more energy consumption in corresponding

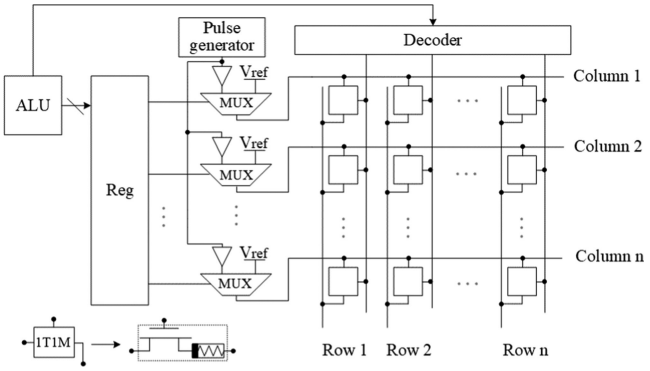


Fig. 2. Circuit level design of the NPC method. Each cell includes one selection transistor forming the one-transistor one-memristor (1T1M) array to avoid sneak path current problems.

memristors. Because specific features are various for different given training data, and only corresponding conductance of memristors will be updated to record feature in one iteration, inevitably, this will lead to uneven pulse distribution in a crossbar array. Additionally, the maximum number of the pulses determines the writing latency of the update stage in one iteration as a critical path. Note that, such maximum number for updating weight in different iterations are different due to the presence of the different training data and status of the current conductance.

In a system using multilevel memristor, to save energy, reduce writing latency, and organize the update timing, we propose a universal NPC method in this paper. The traditional system originally has writing pulses whose widths are appropriate, on the one side, to regulate the conductance of a memristor, on the other side, not to damage the device. Simultaneously, each writing pulse is identical during different update processes. The proposed NPC method, instead of updating the weight by the number of the pulses that are directly converted from the value of weight change in each iteration, only applies the minimum number of the pulses and keeps the original width of writing pulses, as shown in Fig. 2. 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 weight change that are calculated by an ALU. Then these values are transmitted to multiplexers as control signals. Multiplexers select the minimum number of writing pulses that come from a pulse generator as output when control signals are enabled. The enabled signal means the corresponding memristor needs to be updated. The minimum number of writing pulses is determined by the system that has a minimum number of pulses to change the conductance (update conductance). For instance, in the simulator (Section 4.2), one writing pulse with a certain width and amplitude is enough to change a conductance of a memristor. Therefore, the minimum number of writing pulses is one for this work. Some other systems with different types of analog memristor need at least more than one writing pulse (for example: two) to make the conductance change, such a minimum number (two) of pulses is specified and can be applied by the NPC method. Thus, NPC compresses all updating number of the pulses to the minimum, so that the update time in different iterations are the same.

For the system with the NPC method, the system reduces the number of pulses to one at every single update. Some weights need to be updated for several times to reach the certain value during entire training process according to a global minimum of loss function that is calculated by algorithm. Thus, multiple pulses and updates are implemented at entire train process rather than within once update. Such necessary multiple updates consume indispensable energy. Thus, the energy saving is limited. But as for latency, because NPC method reduces the number of pulses to one at every single update, although sometimes several updates are needed for a memristor, each update can be parallel with the update for the other memristor in the same row, which is actually all memristors in a row shared the update time without additional latency. Therefore, the latency reduction is more notable.

This NPC method reduces energy consumption by compressing the number of pulses, especially at several beginning iterations that have a dramatic change of weight, and effectively reduces the writing latency by reorganizing the update timing. Additionally, the NPC method is a general method. Besides the edge AI in IoT systems, it can be adapted to any other pulse incentive multilevel memristor-based ANNs for online learning systems.

4 Application on Digits-Image Recognition

In order to verify the effectiveness of the proposed NPC method in the edge AI, the MLP simulator is used to emulate the learning classification scenario with the Modified National Institute of Standards and Technology (MNIST) handwritten dataset [29]. The memristive crossbar has energy efficiency and areas superiority compared with CMOS synapses in the very large scale integrated (VLSI) circuit in online learning and can be necessary to overcome the effect of device variability and alternate current paths [30]. 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 algorithms [29]. We adopt the online learning hardware platform, NeuroSim+ [26], [29], which is based on a ReRAM with Silver (Ag) and Silicon (Si) as active layer [14], that needs to constantly write the crossbar array to perform handwriting recognition. The networks in this simulator contain a three-layer with 400 neurons, 100 neurons, and 10 neurons, respectively and base on memristors that can tune the conductance by voltage pulse stimulus [29]. Since edges of the images are not the most informative, one handwritten digit is cropped into 20 x 20. The recognitions of networks are ten digits. Thus, the input layer is 400 neurons and the output layer is 10 neurons. Note that, the availability of the NPC method is not constrained by the number of hidden layers and the dimensions of each hidden layer and therefore can be adapt to any architecture and dimension in a given network and realize performance improvement. Parameters that are set in the simulator come from the results of the real memristor measurement [14]. Therefore, this MLP simulator is a standalone functional simulator that is able to evaluate the learning accuracy and device-level performance during the learning process.

4.1 Algorithms

The Stochastic Gradient Descent (SGD) algorithm is one possible solution to accelerate the gradient descent process to use approximate methods that goes through the data in

samples composed of random examples drawn from the original dataset [31], which is the major algorithm used in this simulator. In fact, SGD is a rough approximation, producing a non-smooth convergence. Because of that, variants were proposed to compensate, such as the Momentum, Adaptive Gradient (AdaGrad), Root Mean Square Prop (RMSProp), and Adaptive Moment Estimation (Adam) [31], which are included in this simulator.

4.2 Hardware Structure

In this simulator, each simulation trains up to 125 epochs including 1000000 images, and every epoch randomly selects 8000 images from 60000 training images. The testing dataset has 10000 images and the system runs test after each training epoch. The networks will continually learn the feature of input data after 125 epochs since this simulator is online learning network [29]. The metrics are evaluated with 125 epochs in this work. As for the flow of the training and testing, firstly, at the beginning of the training process, all weights are initialized to simulate the conductance of untrained memristors. Secondly, the system randomly selects one image from the dataset and follows the ANN algorithms to process forward propagation and backpropagation. Thirdly, the system gets delta weight that will be converted to the number of the pulses to be applied for weight updating. Fourthly, the NPC method is applied to compress the number of the pulses to one in this system. Using one pulse that is processed by the NPC method to update the corresponding conductance of a memristor. Fifthly, the second to fourth steps are repeated until the system trains 8000 images and the system runs the test process. Finally, the above procedures except for the first step will repeat 125 times.

Specifically, the hardware implementation block diagram of the NPC method for one image training is shown in Fig. 3. The system follows ANN algorithms by forwarding propagation to get recognition results, which includes vector-matrix multiplication operations. The "label" data of the training dataset is involved by backpropagation to get delta-weight for each memristor that is the value of weight needs to be updated, as shown in Fig. 3. Then the NPC method is implemented to compress the number of pulses. In our case, the system compresses the number of pulses to one. Therefore, when the delta-weight is larger than that corresponding to one pulse, the multiplexer will generate the signal that selects only one pulse to update the memristor, as shown in Fig. 2. The basic peripheral and internal circuits that are included in this platform such as MUX, Adder, and MUL and so on are explained in [26] and [29].

4.3 Metrics Estimation

In the simulation, the reading voltage is 0.5 V and the reading pulse's width is 5 ns. For the writing process, the voltage of the LTP and LTD is 3.2 V, -2.8 V, respectively. The pulse width of writing is 300 μs for both LTP and LTD [14], [26]. Reading and writing energy are determined by both the operations of the periphery circuit and the reading/writing within the crossbar array. In terms of the reading energy, it includes the operation energy of the periphery circuit - decoder, multiplexer, register, and analog-to-digital converter, etc., and the energy within the crossbar array - word lines, bit lines, and memristors. As for writing energy, it includes the operation energy of the periphery circuit – decoder, pre-charger, etc., and the energy within the crossbar array - word lines, bit lines, and memristors that are selected to update [26].

As for the training process, the latency of the memristor-based crossbar array includes reading and writing latency that is determined by both the operations of the periphery circuit and the reading/writing within the crossbar array. In terms of the reading latency, it includes the operation latency of the periphery circuit - decoder, multiplexer, register, and analog-to-digital converter, etc., and the latency within the crossbar array that is the width of the reading pulse. As for writing latency, it includes the operation latency of the periphery - decoder and pre-charger circuit, etc., and the latency within the crossbar array - both in the LTP and LTD that is calculated by multiplication of the number of update pulses and width of pulses. Note that, for each row, the latency of the writing is determined by both the latency of the maximum LTP process and the maximum LTD process, as shown in:

$$L = w_{pulse} * \sum_{1}^{n} (max(p_1,\ p_2,\ \dots p_m) + max(d_1,\ d_2,\ \dots d_m)), \tag{1}$$

where L is the latency of the writing within the whole crossbar array at a certain iteration, w_{pulse} is the width of writing pulse, n is the number of rows, p is a latency of the LTP process for one memristor, d is a latency of the LTD process in one memristor, and m is the number of the corresponding columns [26].

5 Results and Discussion

A comprehensive suite of the digit recognition simulations has been conducted to explore the proposed NPC method for performance improvement in a hardware implementation as the edge AI. We perform five groups of simulations to explore the NPC method with five different algorithms, where the number of pulses is compressed to one.

5.1 Energy Consumption

As for memristive crossbar array, reading energy and writing energy constitute the total energy consumption, as listed in Table 1. The reading energy for different algorithms is determined by the size of the crossbar array and the number of the total iterations. According to the process of a vector-matrix multiplication in a given memristive crossbar array including 41000 memristors in our experiments, the reading energy is always the same - 0.4 nJ for each iteration. However, the reading energy is much smaller than writing energy, this is because 1) voltage of reading pulse is lower than the voltage of writing pulse [17], as mentioned in Section 4.3; and 2) the number of the pulses for the reading is usually much less than the writing. As shown in Fig. 4, the writing energy for crossbar array changes with the number of epochs without and with the NPC method. It demonstrates the system consumes less writing energy with the NPC method than that without the NPC method, and the energy-saving is increased with the increased number of

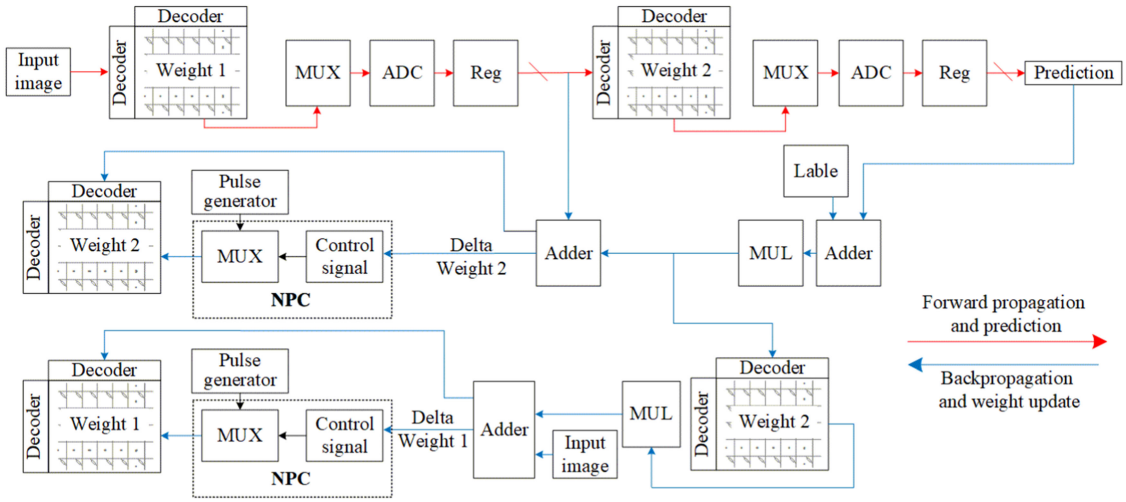


Fig. 3. Hardware implementation block diagram of the NPC method. The red path is forward propagation and prediction. The blue path is backpropagation and weight update.

epochs. Furthermore, the details of numerical writing energy are shown in Fig. 5. The red and blue bars represent the writing energy after 125 epochs without and with the NPC method with five algorithms. Because less pulse is used in weight updating following the NPC method, the writing energy saving is from 7.7 percent to 26.9 percent. Furthermore, the AdaGrad consumes the least writing energy in five algorithms that is respectively 3.9 mJ and 3.6 mJ without and with the NPC method, which realizes 7.7 percent writing energy saving. Meanwhile, as listed in Table 1, the total energy of the AdaGrad is respectively 4.3 mJ and 4.0 mJ without and with the NPC method, which realizes 7.0 percent total energy saving. Additionally, the RMSProp consumes the largest energy in five algorithms. It consumes respectively 17.1 mJ and 12.5 mJ writing energy without and with the NPC method, which realizes 26.9 percent writing energy saving. The total energy is respectively 17.5 mJ and 12.9 mJ without and with the NPC method, which realizes 26.3 percent total energy saving. Such energy saving makes the proposed NPC method especially suitable for the edge AI in IoT systems with the serious energy constrain.

Table 2 shows the recognition accuracy of the five algorithms after the first image, first epoch, and 125th epoch training, respectively. All accuracy is higher than 92.3 percent after 125 epochs training. The difference without and with the NPC method is smaller than 1.0 percent. Additionally, the accuracy is limited by the number of bits of input data and hardware based constraint that includes ADC precision and circuit noise in this platform [26]. Therefore, the NPC method does not much hurt recognition accuracy.

In addition, the NPC method effectively produces a smoother convergence of the training process, which reduces the excessive fluctuation of the recognition accuracy. Taking SGD as an example, Fig. 6 shows the recognition accuracy with increasing epochs. The regressions are carried out by the Nelder model to fit the experimental data

TABLE 1
Total Energy and Saving Percentage of Neural Network With/Without NPC Method With Five Algorithms

	Algorithm	Without NPC (mJ)	With NPC (mJ)	Energy Saved (%)
Total Energy	SGD	6.6	5.8	12.1
	Momentum	6.7	5.7	14.9
	AdaGrad	4.3	4.0	7.0
	RMSProp	17.5	12.9	26.3
	Adam	12.2	9.8	19.7

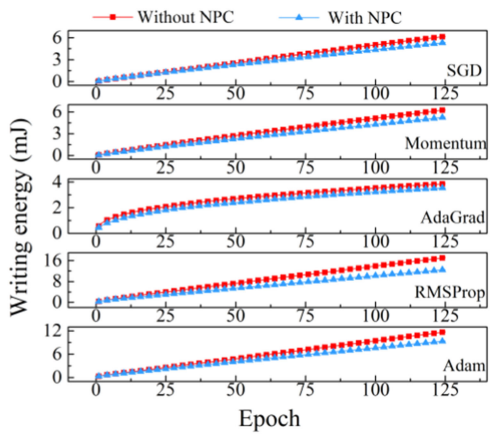


Fig. 4. Writing energy as a function of epoch with five algorithms. Red and blue lines represent without and with the NPC method.

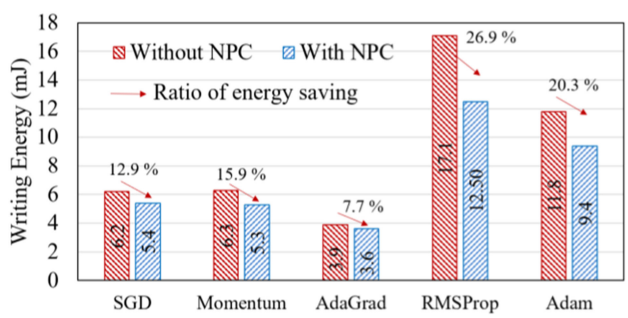


Fig. 5. Writing energy with five algorithms. The red and blue bars represent the writing energy after 125 epochs without and with the NPC method.

TABLE 2
Recognition Accuracy of Neural Network With Different Training Stage (%)

Algorithm	1st image without/ with NPC	1st epoch without/ with NPC	125th epoch without/ with NPC	Fluctuation (after 125 epoch)
SGD	14.8/14.8	70.4/71.8	92.7/92.8	+0.1
Momentum	14.8 / 14.8	76.9/72.4	93.1/93.7	+0.6
AdaGrad	12.9/14.7	70.1/84.0	93.3/92.3	−1.0
RMSProp	11.3/14.7	79.8/83.0	93.6/94.5	+0.9
Adam	12.5/14.8	83.4/83.4	94.2/94.7	+0.5

TABLE 3
Number of the Pulses and Power Consumption for Weight Update

Layer	Pulse number without NPC	Pulse number with NPC	Pulse saved (%)
Weight 1	70250859	37498805	46.6%
Weight 2	16541627	10283640	37.8%
Power (nW)	Without NPC 354	With NPC 586	

of the simulation without and with the NPC method. The Reduced Chi-Sqr with the NPC method, 0.4, is higher than that without one, 0.2, which demonstrates that the fluctuation of the recognition accuracy is reduced by the NPC method [32]. Therefore, a smooth convergence of the training process is another reason that the total writing energy is lower than that without the NPC method. Although the NPC method makes the system learn slowly at the beginning of the first epoch, this disadvantage disappears after 1250 images learning as shown in the inset of Fig. 6. Therefore, the drawback of the NPC method can be neglected for the whole system.

Finally, we calculate the sum of the number of the updating pulses required for 125 epochs, taking SGD as an example, without and with the NPC method as shown in Table 3. Note that the power (energy/latency) with the NPC method is higher than that without the NPC method. The reason is that the saved latency with the NPC method is more significant than the saved energy with the NPC method. However, the total number of the pulses is saved 46.6 percent and 37.8 percent for weight 1 and weight 2 layer, respectively. The fewer pulses are utilized, the less energy is consumed by the pulse generator. Those results further prove that the proposed method can effectively reduce energy consumption during the training process in ANN.

5.2 Pulse Distribution

The energy of the pulses in the reading and writing process will generate thermal power. The more pulses are generated for updating conductance, the more heat crossbar array generates. Since the reading pulse is evenly distributed, now we only analyze the writing pulse distribution. Additionally, the accuracy of the recognition at the beginning stage is very low since the weight is randomly initialized before training. Therefore, the weight change is larger at the beginning stage than later, which can be reflected by the difference of accuracies, as shown in Table 2 after training of the first image and first epoch without and with the NPC method in five algorithms. All the accuracies are lower than 15.0 percent after training of the first image and higher than 70.0 percent after training of the first epoch. The increment of the accuracy is more than 55.0 percent. Thereby, it indicates to need more pulses at the beginning stage of the training for large weight updating.

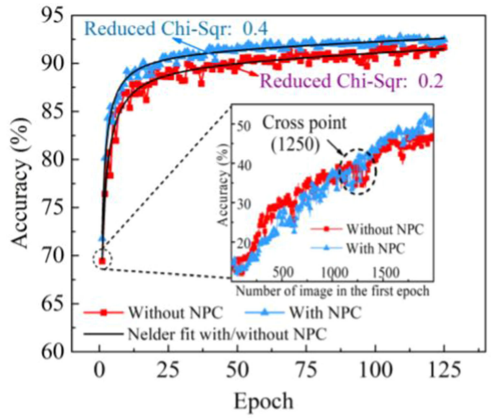


Fig. 6. Nelder model fit without and with the NPC method.

Because of more writing pulses at the beginning stage of the training, we extracted weight update's pulse distribution at the 1st and 1,000th iteration at the first epoch with the AdaGrad algorithm as an example. Each iteration will update weight 1 and 2 layers. Figs. 7a and 7c represent the weight 1 layer with 400 input and 100 output for the 1st and 1,000th iteration, and Figs. 7b and 7d represent the weight 2 layer with 100 input and 10 output for the 1st and 1,000th iteration. The Z-axis is the number of the pulses for weight update that includes LTP and LTD. As shown in Figs. 7a and 7b, for the 1st iteration without the NPC method, in the weight 1 layer, the maximum number of the pulses is 52 that is in the 91st column covering the related 125 rows; in the weight 2 layer, the maximum number of the pulses is 30 that is in the 60st row 6th column and the 91th row 6th column. Similarly, as shown in Figs. 7c and 7d, for the 1,000th iteration without the NPC method, in the weight 1 layer, the maximum number of the pulses is 5 that is in the 75th column covering the related 90 rows; in the weight 2 layer, the maximum number of the pulses is 10 that is in the 2nd column covering 7 rows. With the increasing iterations, the weight is closer to the global minimum. Therefore, the number of the pulses decreases with an increasing number of iterations. It is concluded that without the NPC method, extremely uneven heat distribution is caused by pulses uneven distribution. Figs. 7a′, 7b′, 7c′, and 7d′ show the distribution of the pulses with the NPC method at the same update stage. All of the numbers of the pulses are compressed to one. Note that, the position of the pulses that are used to update at 1,000th iteration in Figs. 7c and 7c′ are different, and pulse distribution in Fig. 7c′ cannot be obtained directly by compressing all pulses in Fig. 7c to one. The reason is that after weight updating based on the first image without and with the NPC method, the following weight updating between both of that is totally different since the current pulse distribution in Figs. 7c and 7c′ only bases on

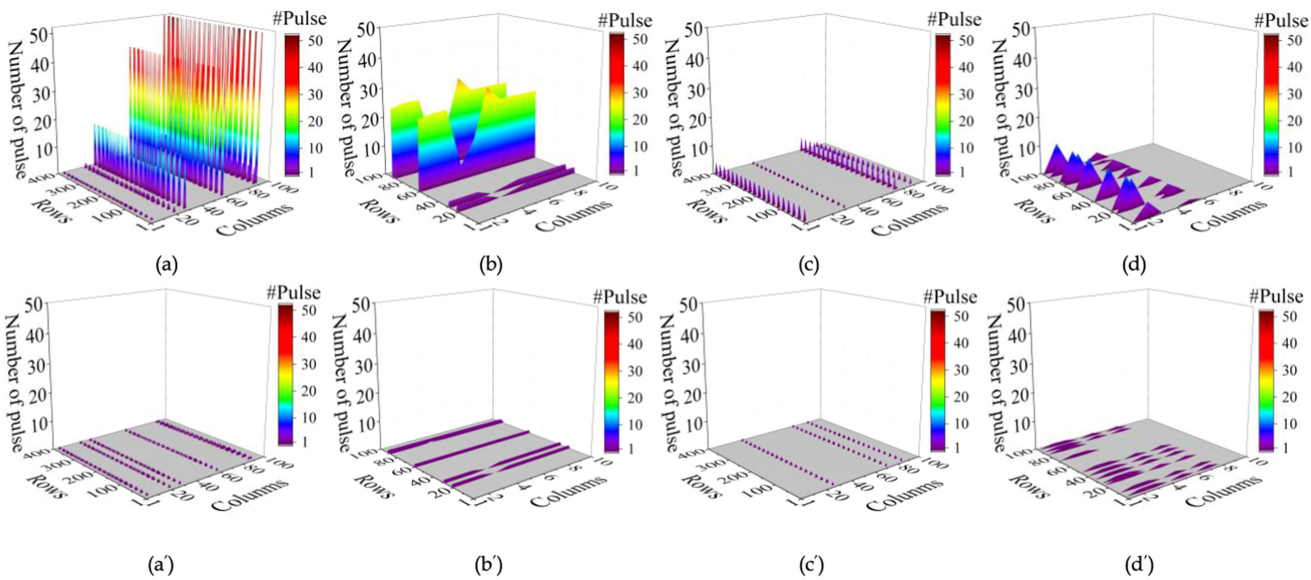


Fig. 7. Pulse distribution of crossbar array in 1st and 1 000th iteration without and with the NPC method.

the present image and current weight. It is the same reason for different pulses distribution in Figs. 7d and 7d′. For further analysis, Table 4 shows the mean and standard deviation of those number of update pulses without and with the NPC method. With the NPC method, the mean of the number of the pulses decreases by 18.8, 9.5, 1.6, and 4.5 for weight layers at 1st and 1,000th iteration, respectively. All the standard deviation of the number of update pulses with the NPC method is 0, but that is 20.0, 11.2, 1.6, and 3.6 without the NPC method, respectively. Thus, it is verified that even pulse distribution is achieved using the NPC method in the ANN system.

5.3 Latency

For a given ANN structure in edges, every iteration has stable 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. Each row's writing latency is determined by the maximum number of writing pulses as a critical path. Thereby, the main latency for crossbar array is writing latency that strongly depends on the maximum update pulses of each row. For example, as shown in Fig. 8, suppose the writing latency is four pulses without the NPC method for the selected row, but it is only one pulse with the NPC method, reducing the latency of the pulses by 75.0 percent. In some extreme cases, suppose the one change is from the minimum conductance to the maximum conductance, which has 100 levels (default in simulator [26]), theoretically, the maximum number of the needed writing pulses without the NPC method is 100. However, with the NPC method, the maximum number of the writing pulses is still one, since the number of the writing pulses is compressed to one, reducing the latency of the pulses by up to 99.0 percent. Therefore, with the NPC method, Equation (1) can be improved to:

$$L = w_{pulse} * ((\ NO.\ of\ p\ ) + (\ NO.\ of\ d\ )), \tag{2}$$

where L is the latency of the writing within the whole crossbar array at a certain iteration, w_{pulse} is the width of writing pulse, $NO.\ of\ p$ and $NO.\ of\ d$ are the total number of rows that are selected for LTP and LTD process. Latency schematic diagram of writing process is reduced by the NPC method as shown in Fig. 9. In each iteration, the system will read memristor that is forward propagation, then calculate delta weights that include backpropagation, and finally write memristor. When applying the NPC method, the time of writing will be decreased to minimum by compressing to one pulse.

Indeed, the NPC method theoretically impacts the learning speed, but this impact only occurs at the beginning of the first epoch, which can be neglected comparing 125 epochs, as shown in the inserted figure in Fig. 6. Only before the cross point - 1250, the accuracy without the NPC method is higher than that with the NPC method.

TABLE 4
Statistics of Number of the Pulses With Different Iteration[a]

	1st iteration without/ with NPC	1000th iteration without/with NPC
M of weight 1	19.8 / 1.0	2.6 / 1.0
M of weight 2	10.5 / 1.0	5.5 / 1.0
SD of weight 1	20.0 / 0	1.6 / 0
SD of weight 2	11.2 / 0	3.6 / 0

[a] *M represents mean. SD represents standard deviation.*

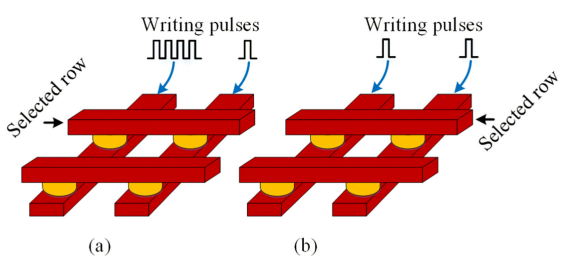


Fig. 8. Weight update by pulse signal in selected row. (a) without the NPC method and (b) with the NPC method.

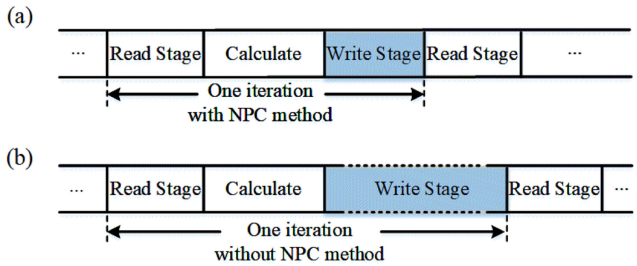


Fig. 9. Latency schematic diagram. (a) The latency of write stage is reduced by the NPC method compared to original system. (b) Latency diagram of original system. The latency of write stage may different, which depends on the max number of pulses for LTP and LTD processes.

Fig. 10 shows the total writing latency that is normalized after 125 epochs without and with the NPC method. The total writing latency is decreased by 30.0 percent-50.0 percent for five algorithms, respectively. Therefore, the NPC method extremely effective in reducing writing latency, which is preferred by the edge AI with real-time requirements. Additionally, because of the NPC method, every iteration has the same number of the writing pules, the timing regularity of the system and the reliability of the system is greatly improved.

5.4 Nonlinear Property of Memristors

Ideally, when LTP or LTD occurs, the change in the conductance of an ideal synapse device is proportional to the number of writing pulses. However, in reality, such change mismatches the writing pulses due to the nonlinearity of memristors [33], [34], [35]. In our simulation, the actual conductance curve is labeled with a nonlinearity value from +3 to -3 [33], [34], which represents the extent to the curve deviates from the ideal linear device.

Taking the AdaGrad algorithm as an example, Table 5 shows the total writing energy without and with the NPC method, under the significant nonlinear property. The recognition accuracy does not have significant fluctuation. The accuracy recovery is done by piecewise linear method [33] that regains accuracy over 90 percent under 3/-3 circumstance. All of the total writing energy with the NPC method is lower than without the NPC method. Energy saved is from 7.7 percent to 13.0 percent, respectively. Thus, the NPC method is proved to effectively reduce writing energy even with the nonlinear property of a memristor.

TABLE 5
Writing Energy and Recognition Accuracy of Neural Network with Nonlinearity

NL[a] (LTP/LTD)	Writing energy without NPC (mJ)	Writing energy with NPC (mJ)	Energy saved (%)
0 / 0	3.9	3.6	7.7
1 / -1	4.6	4.0	13.0
2 / -2	4.6	4.2	8.7
3 / -3	4.7	4.1	12.8

NL (LTP/LTD)	Recognition accuracy without NPC (%)	Recognition accuracy with NPC (%)
0 / 0	93.3	92.3
1 / -1	92.2	92.0
2 / -2	89.1	88.3
3 / -3	84.6	86.7

[a] *NL represents the value of nonlinearity.*

5.5 Variations of Memristors

Because of physical limitations of a memristor, minimum conductance variation (G_{min}), maximum conductance variation (G_{max}), ON/OFF ratio variation (G_{max}/G_{min}), cycle-to-cycle variation (CtoC), and device-to-device variation (DtoD) [13] exist in the application of memristor-based hardware implementation, as shown in Fig. 11. To explore the effectiveness of the NPC method, we take AdaGrad algorithm as an example and investigate these variations following standard/Gaussian distribution N (μ, σ) into consideration. In our experiments, minimum conductance subjects to N (G_{min}, $\sigma \times G_{min}$), and maximum conductance subjects to N (G_{max}, $\sigma \times G_{max}$). Device-to-device variation subjects to N (NL, σ) distribution. Cycle-to-cycle variation that subjects to N (0, $\sigma \times (G_{max}\text{-}G_{min})^2$) represents conductance deviations in each weight update [34].

Above, NL, Gmax, and Gmin are fixed parameters for each simulation. ON/OFF ratios are configured as 17 in variation 1 and 15 in variation 2. For Variations 1 and 2 in Table 6, we set σ of the minimum conductance, maximum conductance, device-to-device, and cycle-to-cycle variation as 5.0 percent, 5.0 percent, 0.5, 1.0 percent, and 15.0 percent, 15.0 percent, 1.4, 2.5 percent, respectively [33], [34]. Table 6 shows the result of simulations under different circumstances.

In Table 6, for different circumstances without the NPC method, the total writing energy is 3.9 mJ and 3.8 mJ. After utilizing the NPC method, the total writing energy is saved

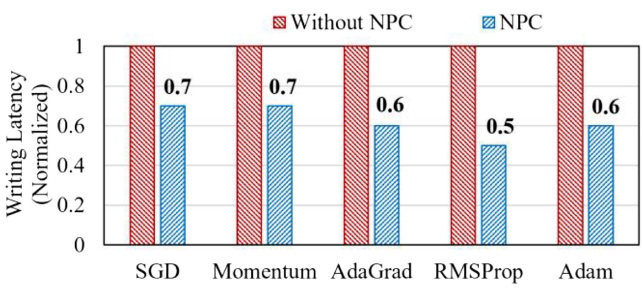


Fig. 10. Writing latency of crossbar array without and with the NPC method.

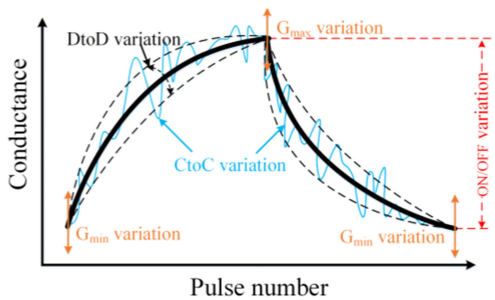


Fig. 11. Variations of memristor. DtoD, CtoC, Gmax and Gmin represent device to deice, cycle to cycle, maximum conductance, and minimum conductance, respectively [26].

TABLE 6
Experimental Results of Neural Network With Variations

	Variation 1 without/with NPC	Variation 2 without/with NPC
Writing Energy (mJ)	3.9 / 3.5	3.8 / 3.2
Recognition accuracy (%)	91.9 / 90.6	86.1 / 82.4
Writing Latency (normalized)	1 / 0.7	1 / 0.6

by 10.4 percent and 15.3 percent, respectively. Additionally, recognition accuracy does not have big degradation, and writing latency is reduced by 30.0 percent and 40.0 percent. Thus, even with many variations, the NPC method is still efficient to reduce energy consumption and writing latency of the crossbar array.

5.6 Failure Rate, Endurance, and Aging

A typical manufacturing process typically seeks a failure rate of <10 percent [36]. To evaluate the influence of failure rate in a crossbar array for the edge AI, 5 percent, 10 percent, and 15 percent of the fault in the crossbar array are simulated, as shown in Fig. 12 [37]. The random positions of the fault memristors are chosen in the crossbar array. The ratio of the stuck at 0 and 1 is 1: 5.2 [36]. Taking the SGD algorithm as an example, Table 7 shows the accuracy under the influence of the failure rate in this network. The neural network with the NPC method still has accuracy improvement as compared to that without the NPC method according to results of the mean and standard deviation that are obtained from 500 random cases of each failure rate. When the failure rate is 15 percent, the accuracies of the neural network without and with the NPC method both reduce about 3 percent.

Memristors can only be programmed reliably for a given number of times. Afterward, the conductance tunability of the memristor deviates from the initial state, which is called aging, and it limits the lifetime of memristor-based crossbars in the edge computing system [38]. The conductance is assumed to drift towards different final states, or randomly drift, based on different various drift rates, which are equivalent to conductance drift different amounts over 10 years, respectively [39]. Taking the SGD algorithm as an example, Fig. 13 shows the accuracies under the influence of the aging

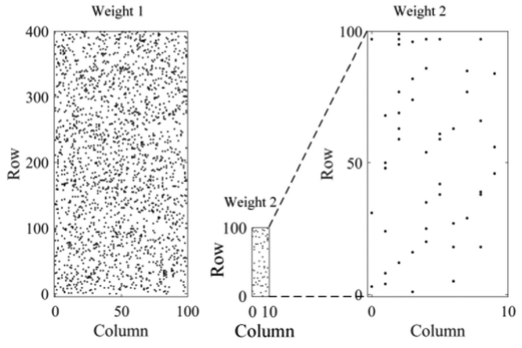


Fig. 12. Random positions of the failure memristors in crossbar array with 5 percent failure rate.

TABLE 7
Recognition Accuracy and Standard Deviation With Different Failure Rates

Failure rate	Mean[a]		Standard deviation[a]	
	Without NPC	With NPC	Without NPC	With NPC
5%	91.7%	92.0%	0.0050	0.0039
10%	91.0%	91.4%	0.0058	0.0044
15%	88.5%	89.6%	0.0071	0.0047

[a] *Mean and Standard deviation are obtained including 500 random cases of each failure rate.*

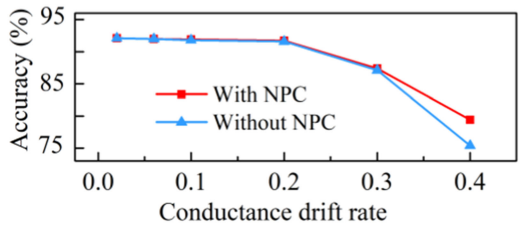


Fig. 13. Recognition accuracy with different conductance drift ratios that are 0.02, 0.06, 0.10, 0.20, 0.30, and 0.40.

in this network. The precision, recall, and F1 score are shown in Table 8. The restoration of accuracy can be completed by retaining and remapping method [40]. The NPC method is still effective by comparing to the accuracy that is without the NPC method.

In addition, the endurance of a memristor is one limitation for high frequency writing in an ANN system [14], [41]. The NPC method extremely saves the number of writing pulses, as shown in Table 3. Therefore, this method is still effective with failure and aging circumstances and benefits the cycling endurance performance of a memristor.

5.7 NPC Method With Different Architecture and Database

To verify the NPC method in different architecture, different hidden layers are simulated as shown in Fig. 14. As expected, the recognition accuracy for using the NPC method is higher than that without the NPC method.

TABLE 8
Classification Report[a]

Class	Without NPC method			With NPC method		
	P	R	F1	P	R	F1
0	0.742	0.956	0.836	0.812	0.945	0.874
1	0.628	0.990	0.768	0.753	0.990	0.856
2	0.727	0.806	0.765	0.740	0.850	0.791
3	0.805	0.784	0.794	0.781	0.665	0.719
4	0.630	0.764	0.690	0.811	0.797	0.804
5	0.840	0.577	0.684	0.789	0.609	0.687
6	0.770	0.863	0.814	0.923	0.864	0.893
7	0.890	0.714	0.792	0.832	0.791	0.811
8	0.887	0.507	0.645	0.843	0.633	0.723
9	0.810	0.449	0.578	0.757	0.798	0.777
Micro_avg	0.773	0.741	0.737	0.804	0.794	0.793

[a] *The result with 0.4 conductance drift ratios. P, R, and F1 represent precision, recall, and F1-score, respectively.*

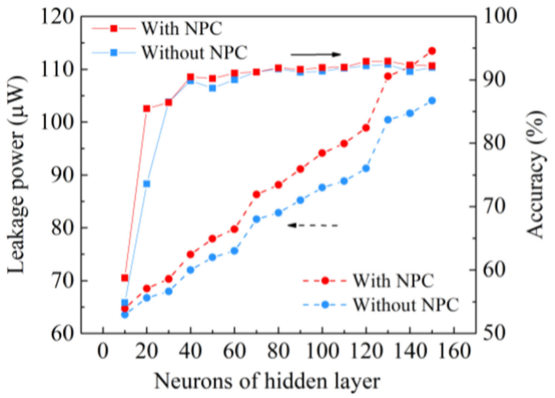


Fig. 14. Leakage power with different number of neurons of hidden layer. Recognition accuracy with different number of neurons of hidden layer.

However, the leakage power is increasing when increasing the neurons of the hidden layer as shown in Fig. 14. The leakage power with the NPC method is a little higher (<10 percent) than that without the NPC method because multiplexors are added. Thus, the NPC method is effective with different architecture.

Furthermore, the NPC method in VGG-8 with memristive crossbar array architecture and CIFAR-10 database are evaluated as shown in Table 9. The accuracy difference without and with the NPC method is smaller than 1.0 percent. Therefore, the NPC method does not much hurt recognition accuracy. Additionally, the accuracy is limited by the hardware based setting that includes variations and nonlinearity in this platform [42]. As expected, the NPC method respectively reduces latency by 46.00 percent and energy consumption by 16.67 percent in the system.

5.8 NPC Method and Other Works

PRCoder as an algorithm was proposed for different RRAM applications [15]. The cycle-rehabilitate technique was used to alleviate thermal crosstalk [16]. At the same time, increasing the size of insulator or utilizing new materials with higher thermal conductivity for improving performance were proposed in [43], [44]. A new structure, thermal-house, was presented to optimize the thermal management [45]. However, those new algorithm or new material/structure of device based solutions inevitably increase the complexity of peripheral circuit or the difficulty of manufacture process, even increasing the latency.

The NPC method is a simple and feasible method for the edge computing in IoT systems. As shown in Table 10, as compared with the state-of-art, the NPC method does not need to use special structure and material. Simultaneously, it does not need to add an extra algorithm to alleviate uneven pulse distribution. What's more, with the NPC method, the energy consumption is effectively reduced by 7.5 percent and the writing latency is averagely reduced 38.0 percent.

TABLE 9
Experimental Results of Neural Network with VGG-8 and CIFAR-10

	Accuracy	Latency[a]	Energy
With NPC method	90.3%	0.54	0.25 J
Without NPC method	91.1%	1.00	0.30 J
Difference	0.88%	46.00%	16.67%

[a] *Latency values are normalized.*

TABLE 10
Comparison of the State-of-the-art

	[34]	This work
Energy consumption	6.7 mJ	6.2 mJ
Write latency (Normalized)	1	0.62

	[15]	[16]	[43]	[44]	[45]	This work
Without new material or structure	√	√	×	×	×	√
Without adding extra algorithm	×	×	√	√	√	√

6 Conclusion

In this paper, we propose the Number of Pulses Compression (NPC) method to reduce energy consumption, decrease writing latency, and improve timing regularity of memristor-based smart edge computing in IoT systems. The NPC method is verified to be effective based on devices to algorithms architectures. Instead of modifying the traditional algorithm-based technology, the NPC method that only needs to add a multiplexer circuit before every writing operation in the weight-updating process, to optimize the performance of the system. ANNs in five algorithms with nonlinearity from (0/0) to (3/-3), different failure rates (5 percent, 10 percent, and 15 percent), two variations conditions, different architectures, and aging effect have been evaluated to investigate the effectiveness of the NPC method in the edge computing. The results indicate that it saves the writing energy of crossbar array by 7.7 percent-26.9 percent and reduces the writing latency by 30.0 percent-50.0 percent. It concludes: 1) The proposed NPC method enables low energy consumption and even pulse distribution to reduce the heat resulted from intensive pulses. 2) Because the number of the pulses for weight update is compressed to one, the NPC method effectively reduces the writing latency and improves timing regularity. 3) The NPC method is still effective under different nonlinearity, failure rates, aged devices, architectures, and variation circumstances in memristor-based ANN for paving the way for the further development of the edge computing in IoT systems.

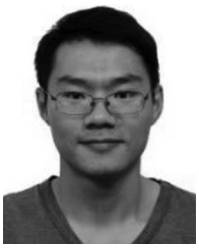
Acknowledgments

This work made use of computing resources at the Center for Computationally Assisted Science and Technology (CCAST), North Dakota State University. This work was supported in part by the National Science Foundation under Grant 1953544 and Grant 1855646.

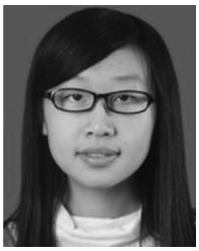
References

[1] G. Yuan *et al.*, "Memristor crossbar-based ultra-efficient next-generation baseband processors," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst.*, 2017, pp. 1121–1124.

[2] R. Cai, A. Ren, Y. Wang, and B. Yuan, "Memristor-based discrete fourier transform for improving performance and energy efficiency," in *Proc. IEEE Comput. Soc. Annu. Symp.*, 2016, pp. 643–648.
[3] C. Li *et al.*, "Analogue signal and image processing with large memristor crossbars," *Nat. Electron.*, vol. 1, no. 1, pp. 52–59, 2018.
[4] M. M. Waldrop, "The chips are down for moore's law," *Nat. News*, vol. 530, no. 7589, pp. 144–147, 2016.
[5] S. C. Bartling, S. Khanna, M. P. Clinton, S. R. Summerfelt, J. A. Rodriguez, and H. P. McAdams, "An 8MHz 75μA/MHz zero-leakage non-volatile logic-based cortex-m0 MCU SoC exhibiting 100% digital state retention at v DD = 0V with< 400ns wakeup and sleep transitions," in *Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2013, pp. 432–433.
[6] N. Sakimura *et al.*, "10.5 A 90nm 20MHz fully nonvolatile microcontroller for standby-power-critical applications," in *Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2014, pp. 184–185.
[7] P.-F. Chiu *et al.*, "Low store energy, low VDDmin, 8T2R nonvolatile latch and SRAM with vertical-stacked resistive memory (memristor) devices for low power mobile applications," *IEEE J. Solid-State Circuits*, vol. 47, no. 6, pp. 1483–1496, Jun. 2012.
[8] L. Chua, "Memristor-the missing circuit element," *IEEE Trans. Circuit Theory*, vol. 18, no. 5, pp. 507–519, Sep. 1971.
[9] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.
[10] A. P. James, "Introduction to neuro-memristive systems," in *Deep Learning Classifiers With Memristive Networks*. Cham, Switzerland: Springer, 2020, pp. 3–12.
[11] A. P. James, "An overview of memristive cryptography," *Eur. Phys. J. Special Topics*, vol. 228, no. pp. 10 pp. 2301–2312, 2019.
[12] O. Krestinskaya, A. P. James, and L. Chua, "Neuromemristive circuits for edge computing: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 4–23, Jan. 2020.
[13] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on memristive systems," *Nat. Electron.*, vol. 1, no. 1, pp. 22–29, 2018.
[14] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett*, vol. 10, no. 4, pp. 1297–1301, 2010.
[15] Y. Li, H.-H. Shen, C. Li, and F. Zhang, "An efficient parity rearrangement coding scheme for RRAM thermal crosstalk effects," in *Proc. IEEE 12th Int. Conf. ASIC*, 2017, pp. 20–23.
[16] P. Sun *et al.*, "Thermal crosstalk in 3-dimensional RRAM crossbar array," *Sci. Rep.*, vol. 5, no. 1, pp. 1–9, 2015.
[17] Y. V. Pershin, and M. Di Ventra, "Practical approach to programmable analog circuits with memristors," *IEEE Trans. Circuits Syst. I Reg. Papers*, vol. 57, no. 8, pp. 1857–1864, Aug. 2010.
[18] Y. Y. Chen *et al.*, "Improvement of data retention in HfO 2/Hf 1T1R RRAM cell under low operating current," in *Proc. IEEE Int. Electron Dev. Meeting*, 2013, pp. 10.1.1–10.1.4.
[19] B. Chen *et al.*, "Endurance degradation in metal oxide-based resistive memory induced by oxygen ion loss effect," *IEEE Electron Dev. Lett.*, vol. 34, no. 10, pp. 1292–1294, Oct. 2013.
[20] U. Russo, D. Ielmini, C. Cagli, and A. L. Lacaita, "Self-accelerated thermal dissolution model for reset programming in unipolar resistive-switching memory (RRAM) devices," *IEEE Trans. Electron Dev.*, vol. 56, no. 2, pp. 193–200, Feb. 2009.
[21] D. Niu, Y. Chen, and Y. Xie, "Low-power dual-element memristor based memory design," in *Proc. 16th ACM/IEEE Int. Symp. Low Power Electron. Des.*, 2010, pp. 25–30.
[22] J. Wang, Y. Tim, W.-F. Wong, and H. H. Li, "A practical low-power memristor-based analog neural branch predictor," in *Proc. IEEE Int. Symp. Low Power Electron. Des.*, 2013, pp. 175–180.
[23] H. Saadeldeen *et al.*, "Memristors for neural branch prediction: A case study in strict latency and write endurance challenges," in *Proc. ACM Int. Conf. Comput. Front.*, 2013, pp. 1–10.
[24] D. Niu, Y. Xiao, and Y. Xie, "Low power memristor-based ReRAM design with error correcting code," in *Proc. IEEE 17th Asia South Pacific Des. Automat. Conf.*, 2012, pp. 79–84.
[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[26] P.-Y. Chen, X. Peng, and S. Yu, "NeuroSim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures," in *Proc. IEEE Int. Electron Dev. Meeting*, 2017, pp. 6.1.1–6.1.4.
[27] Z. Wang *et al.*, "Fully memristive neural networks for pattern classification with unsupervised learning," *Nat. Electron.*, vol. 1, no. 2, pp. 137–145, 2018.
[28] J. Fu, Z. Liao, and J. Wang, "Memristor-based neuromorphic hardware improvement for privacy-preserving ANN," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 12, pp. 2745–2754, Dec. 2019.
[29] P.-Y. Chen, X. Peng, and S. Yu, "NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 12, pp. 3067–3080, Dec. 2018.
[30] S. Choi, P. Sheridan, and W. D. Lu, "Data clustering using memristor networks," *Sci. Rep.*, vol. 5, 2015, pp. Art. no. 10492.
[31] M. A. Ponti, L. S. F. Ribeiro, T. S. Nazare, T. Bui, and J. Collomosse, "Everything you wanted to know about deep learning for computer vision but were afraid to ask," in *Proc. IEEE 30th SIBGRAPI Conf. Graph., Patterns Images Tut.*, 2017, pp. 17–41.
[32] D. W. Hosmer, and S. Lemesbow, "Goodness of fit tests for the multiple logistic regression model," *Commun. Statist.-Theory Methods*, vol. 9, no. 10, pp. 1043–1069, 1980.
[33] J. Fu, Z. Liao, N. Gong, and J. Wang, "Mitigating nonlinear effect of memristive synaptic device for neuromorphic computing," *IEEE Trans. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 2, pp. 377–387, Jun. 2019.
[34] J. Fu, Z. Liao, N. Gong, and J. Wang, "Linear optimization for memristive device in neuromorphic hardware," in *Proc. IEEE Comput. Soc. Ann. Symp.*, 2019, pp. 453–458.
[35] O. Krestinskaya, A. Irmanova, and A. P. James, "Memristive non-idealities: Is there any practical implications for designing neural network chips?," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2019, pp. 1–5.
[36] C.-Y. Chen *et al.*, "RRAM defect modeling and failure analysis based on march test and a novel squeeze-search scheme," *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 180–190, Jan. 2015.
[37] J. Edstrom, D. Chen, Y. Gong, J. Wang, and N. Gong, "Data-pattern enabled self-recovery low-power storage system for big video data," *IEEE Trans. Big Data*, vol. 5, no. 1, pp. 95–105, Mar. 2019.
[38] A. Irmanova, A. Maan, A. James, and L. Chua, "Analog self-timed programming circuits for aging memristors," *IEEE Trans. Circuits Syst., II Exp. Briefs*, vol. 68, no. 4, pp. 1133–1137, Apr. 2021.
[39] P.-Y. Chen and S. Yu, "Reliability perspective of resistive synaptic devices on the neuromorphic system performance," in *Proc. IEEE Int. Rel. Phys. Symp.*, 2018, pp. 5C.4–5C.1–5C.4–4.
[40] C. Liu, M. Hu, J. P. Strachan, and H. Li, "Rescuing memristor-based neuromorphic design with high defects," in *Proc. 54th ACM/EDAC/IEEE Des. Automat. Conf.*, 2017, pp. 1–6.
[42] X. Peng, S. Huang, H. Jiang, A. Lu, and S. Yu, "DNN+ neurosim V2.0: An end-to-end benchmarking framework for compute-in-memory accelerators for on-chip training," 2020, *arXiv:2003.06471*.
[42] K. M. Kim *et al.*, "Voltage divider effect for the improvement of variability and endurance of TaOx memristor," *Sci. Rep.*, vol. 6, 2016, Art. no. 20085.
[43] S. Li *et al.*, "Fully coupled multiphysics simulation of crosstalk effect in bipolar resistive random access memory," *IEEE Trans. Electron Dev.*, vol. 64, no. 9, pp. 3647–3653, Sep. 2017.
[44] Y. Luo, W. Chen, M. Cheng, and W.-Y. Yin, "Electrothermal characterization in 3-D resistive random access memory arrays," *IEEE Trans. Electron Dev.*, vol. 63, no. 12, pp. 4720–4728, Dec. 2016.
[45] D.-W. Wang *et al.*, "Fully coupled electrothermal simulation of large RRAM arrays in the 'Thermal-House,'" *IEEE Access*, vol. 7, pp. 3897–3908, 2018.



Zhiheng Liao received the BE and MS degree in electrical engineering from the Beijing University of Technology, China, in 2014 and 2017, respectively. He is currently working toward the PhD degree with North Dakota State University, Fargo, ND, USA. His research focuses on the emerging device and algorithm optimization for neuromorphic computing and artificial intelligence technology.



Jingyan Fu (Student Member, IEEE) received the BE and MS degree in electrical engineering from the Beijing University of Technology, China, in 2014 and 2017, respectively. She is currently working toward the PhD degree with the North Dakota State University, Fargo, ND, USA. Her research focuses on hardware design and algorithm optimization for neuromorphic computing, and artificial intelligence technology.

Jinhui Wang (Senior Member, IEEE) received the BE degree in electrical engineering from Hebei University, Hebei, China, and the PhD degree from the Beijing University of Technology, Beijing, China. He was a postdoc fellow with the University of Rochester, NY, USA, a visiting professor with the State University of New York at Buffalo, NY, USA, and a visiting scholar with IMEC, Leuven, Belgium. He is currently an associate professor with the Department of Electrical and Computer Engineering, University of South Alabama, Mobile, AL, USA. He has authored or coauthored more than 130 papers and 20 patents in the emerging semiconductor technologies. His research interests include neuromorphic computing, artificial intelligence technology, low-power, high-performance, and variation-tolerant IC design, 3D IC, and thermal solution in VLSI.