Pulse Truncation Enabled High Performance and Low Energy Memristor-based Accelerator

Zhiheng Liao, Jingyan Fu
Electrical and Computer Engineering
North Dakota State University
Fargo, USA
{zhiheng.liao, jingyan.fu}@ndsu.edu

Caiwen Ding
Computer Science and Engineering
University of Connecticut
Storrs, USA
caiwen.ding@uconn.edu

Jinhui Wang
Electrical and Computer Engineering
University of South Alabama
Mobile, USA
jwang@southalabama.edu

SoutheastCon 2022 | 978-1-6654-0652-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SoutheastCon48659.2022.9764063

***Abstract*—Energy consumption and system latency in memristive crossbar arrays become increasingly significant, especially for the ultra-high density memristor based DNN accelerator. A solution is presented in this paper for improving energy efficiency, meanwhile heightening the performance of the DNN accelerator. Specifically, a pulse truncation (PT) method is proposed to reduce number of pulses and not change the original pulse width in every weight update. The DNN accelerator with the PT method is implemented and evaluated based on the fabricated memristor with the active layer - Silver (Ag) and Silicon (Si) and its tested current-pulse characteristics. Different DNN algorithms with various architectures are employed. The experimental results indicate that the PT method cannot only effectively avoid uneven pulse distributions, but also save the writing energy of crossbar array by 8.29%-26.87% and reduce the writing latency by 30%-48%. Finally, considering non-ideal features of memristors, it concludes that even with the significant nonlinearity, many variations, failure rates, and aging effect, the PT method is still much effective.**

***Index Terms*—DNN accelerator, memristor, energy, latency, accuracy**

I. Introduction

Memristor is a emerging device with a simple three-layer structure that can achieve analog operations to exploit multi-level conductance states by external incentive [2]. Therefore, memristors are suitable for the hardware design [3] to enable a Deep Learning Neural Networks (DNNs) [4]. However, same with the CMOS (Complementary metal–oxide–semiconductor) circuit [1], [5], [6], [32]–[34], the high-performance functionality of memristor-based DNN accelerator translates into high energy density and reduced reliability. What is more, according to the main mechanism of the change of resistance [7], the writing process of a memristor consumes much more energy than the reading process, thereby becoming dominant in the total energy consumption during the training process [8]–[10]. In order to investigate the energy consumption for memristor arrays in the DNN accelerator, the physical modeling of memristor cells, and the energy effects induced by the self-heating effect have been investigated [9]–[14]. With some practical guides given for optimization of energy consumption Sun et al. [11] and Wang et al. [12], [14] propose improved cross array structures. In [13] a software-based writing re-arrangement algorithm is proposed to implement the parity rearrangement coding scheme to alleviate the influence of energy consumption and utilize it at the hardware level for different applications.

In this paper, we deal with the energy efficiency problem of memristor-based DNN accelerators by a pulse truncation (PT) method, that compresses the number of pulses without computation overhead to give a feasible solution in practice. Specifically, this paper makes the following contributions: 1) The PT method makes the DNN accelerator more energy-efficient and faster. 2) The PT method compresses the number of pulses for each update step. Hence, in each writing of weight updating, the PT method avoids intensive pulses. 3) By applying the PT method, the writing latency decided by the maximum number of pulses is significantly reduced. 4) The nonlinearities, variations, failures, and aging effect of the memristor are considered in experiments to evaluate the proposed PT method.

This work was supported in part by the National Science Foundation under Grant 1953544 and Grant 1855646.

II. Background

The memristor can enable DNN accelerator through the efficient vector-matrix multiplications. The conductance of a memristor with multi-level [15], [17], is increased when it is stimulated by a positive pulse. This increasing process is named as long-term potentiation (LTP) [30], [32]. Conversely, the long-term depression (LTD) is to decrease the conductance by a negative pulse [30], [32].

III. Methodology

A. Pulse truncation (PT) method

In the memristor-based DNN accelerator, the weight change that is calculated by algorithms is translated into number of pulses to update the conductance of a memristor. At the beginning of the training, a drastic change in conductance consumes large energy in corresponding memristors. Also, only the corresponding memristors will update in the training. Inevitably, this will lead to uneven pulse distributions in an entire crossbar array. Additionally, the maximum number of pulses decides the writing latency in the update stage in one iteration. In order to save energy, reduce writing latency, and organize the timing, a universal PT method in the

memristor-based DNN accelerator is proposed in this paper. The traditional system originally has writing pulses whose widths are appropriate and identical. The number of pulses in each update process is directly converted from algorithms. The proposed PT method, instead of using the pulses that are directly converted from algorithms in each iteration, only applies 1 pulse and keeps the original width of writing pulses [32]. As shown in Fig. 1, the multiplexers get the values of weight change that are calculated by arithmetic logic units (ALU) as control signals. Then multiplexers select reference voltage to transmit signals to pulse generators for generating 1 writing pulse when control signals are enabled. The enabled signal means the corresponding memristor needs to be updated no matter how large the weight change is. Therefore, the PT method compresses the number of the pulses to one at each updating process so that the update time in different iterations is the same. Note that, with the PT method, the learning slows down a little bit at the beginning of the training. This is because more truncation happen at the beginning of the training and the PT method compresses the number of pulses to 1. But, the weight update still keeps in the direction of the algorithm convergence [16], [17]. What's more, according to the given feature of stochastic gradient descent (SGD) algorithms, the accurate and large update without the PT method at the beginning of the training will make the learning jump over minimum [17]. Therefore, the PT method seems to slow down the learning for every weight update, but in fact it has an advantage for decreasing the overall system latency by effectively producing a smoother convergence of the training and reducing the entire training latency that is discussed in Section IV-B.

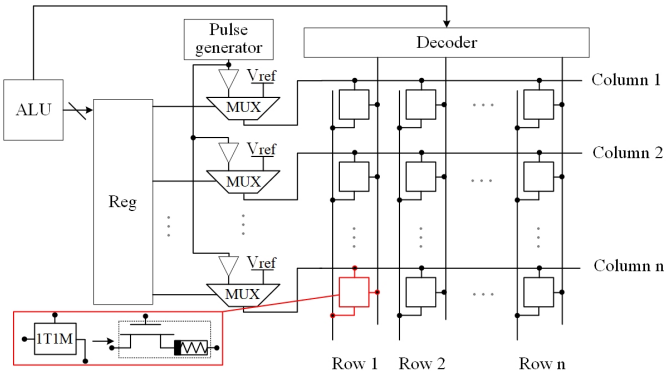

Fig. 1. Circuit design of the PT method.

B. Evaluation and working flow

In order to verify the proposed PT method, the fabricated memristor with Silver (Ag) and Silicon (Si) structure and tested current-pulse characteristics are utilized. As shown in Fig. 2, the curves indicates that the memristor is programmed by consecutive 100 identical positive pulses followed by consecutive 100 identical negative pulses [22]. The conductance is measured at 1 V just after each programmed pulse and the read current is plotted. Also, NeuroSim platform [17] is

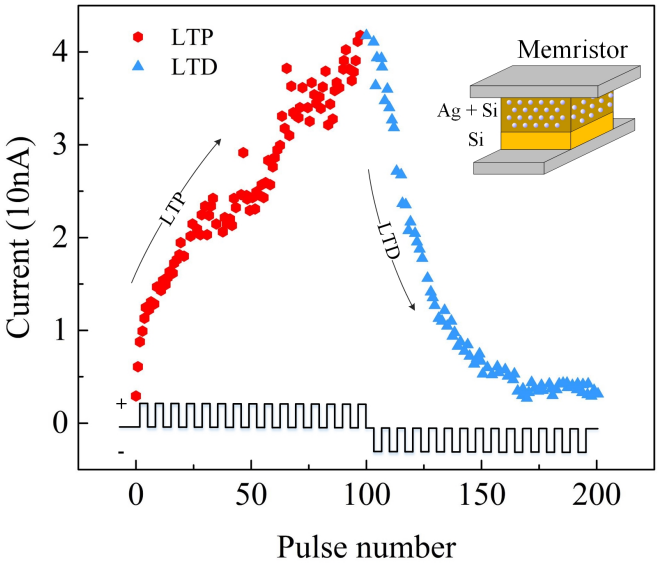

Fig. 2. Memristors response to pulse stimulates.

performed to emulate recognition scenario with the Modified National Institute of Standards and Technology (MNIST) handwritten database [18] and CIFAR-10 database. In this platform, each training runs up to 125 epochs. Due to the online learning mechanism [18], the DNN accelerator can learn the feature from input data at each epoch. The hardware working flow of the PT method for one epoch is shown in Fig. 3. 1) Before the training, all weights are initialized to randomly distribute the conductance of untrained memristors. 2) The DNN accelerator randomly selects one image from the database to do forward propagation and back propagation, and then gets weight change information (Δweight). 3) The PT method is applied to truncate the number of pulses to 1. 4) The DNN accelerator generates 1 updating pulse for weight updating. 5) The DNN accelerator uses 1 pulse to update the conductance of the memristor. 6) The 2-5 steps are repeated until the DNN accelerator trains 8,000 images and it runs the test process. Finally, the above procedures except for the step 1 will repeat 125 times that is 125 epochs including 1,000,000 training times. Note that, the PT method only adds multiplexers into the original DNN accelerator, as shown in Fig. 1.

IV. Results and Discussions

In order to verify the proposed PT method to reduce the energy, decrease latency, and heighten the reliability of a DNN accelerator in a hardware implementation, a comprehensive experiments has been performed. Five different algorithms including SGD, Momentum, Adaptive Gradient (AdaGrad), Root Mean Square Prop (RMSProp), and Adaptive Moment Estimation (Adam) [16] are used in experiments, where the pulse number of the PT method is truncated to 1.

A. Energy consumption

As for the DNN accelerator, the energy consumption is mainly dynamic energy (i.e., the current flow through mem-

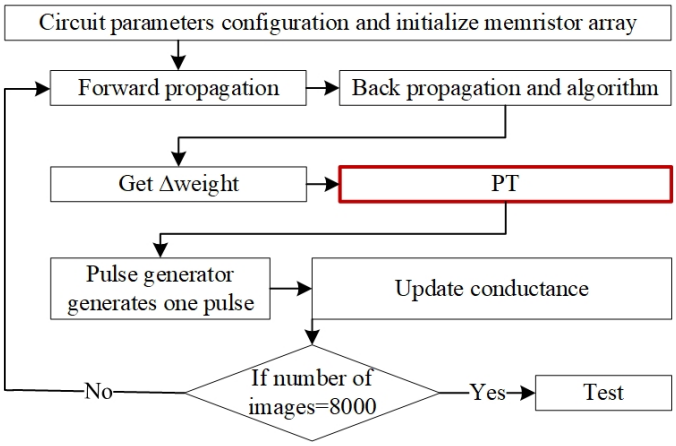

Fig. 3. Working flow of system with the PT method in one epoch.

ristors). The energy consumption on the selected memristor at the weight increase/decrease process is calculated as [18]

$$E_{cell} = U_W^2 N T_p / R$$

U_W and R are the write voltage for weight increase/decrease and the resistance of a memristor. N and T_p are the number of applied write pulses and the pulse width. Besides memristors, the dynamic energy consumption on the metal wire is also calculated and included. Then, the total energy consumption for a memristor-based DNN accelerator can be estimated as the sum of the energy consumption of memristor arrays and sub-circuit modules.

What is more, in the DNN accelerator, the total energy consumption includes reading and writing energy consumption. The reading energy is determined by the size of the crossbar array and the number of total iterations. According to a given crossbar array in our experiments including 41,000 memristors, the reading energy keeps the same - 0.42 nJ for each iteration. Also, the reading energy is usually much smaller than writing energy. It has two reasons: 1) the number of pulses used for the reading is less than the writing; and 2) voltage of reading pulse is much lower than the voltage of writing pulse [8]. Thus, as the result shown in Table I, the DNN accelerator consumes less writing energy with the PT method than that without the PT method. The writing energy reduction is from 8.29% to 26.87%. Furthermore, RMSProp realizes maximum energy saving as 26.87%.

TABLE I
WRITING ENERGY OF DIFFERENT ALGORITHMS

Algorithm	Without PT (mJ)	With PT (mJ)	Energy Saved (%)
SGD	6.20	5.35	13.71
Momentum	6.29	5.28	16.06
AdaGrad	3.86	3.55	8.29
RMSProp	17.12	12.52	26.87
Adam	11.75	9.39	20.08

Table II shows inference accuracies of five algorithms. Because the systems are evaluated based on real device involved platform, all accuracies range from 91% to 95%, which are typical accuracies reported in [18], [19], [22]. Four of five algorithms realize the increased accuracy. Only the AdaGrad algorithm induces accuracy drop, but it is just 1.07%, which can be tolerant. Those results prove that the proposed method can effectively reduce energy consumption during the training process in the DNN accelerator without much accuracy loss.

TABLE II
INFERENCE ACCURACY OF DIFFERENT ALGORITHMS (%)

Algorithm	Without PT	With PT	Fluctuation
SGD	91.94	92.82	+0.96
Momentum	93.13	93.65	+0.56
AdaGrad	93.29	92.29	-1.07
RMSProp	93.63	94.48	+0.91
Adam	94.22	94.73	+0.54

TABLE III
INFERENCE ACCURACY OF DIFFERENT ALGORITHMS FOR 1ST IMAGE AND 1ST EPOCH (%)

Algorithm	1st image without/with PT	1st epoch without/with PT
SGD	14.83/14.75	70.40/71.78
Momentum	14.83/14.75	76.89/72.44
AdaGrad	12.90/14.74	70.08/84.02
RMSProp	11.28/14.74	79.80/82.95
Adam	12.48/14.76	83.41/83.41

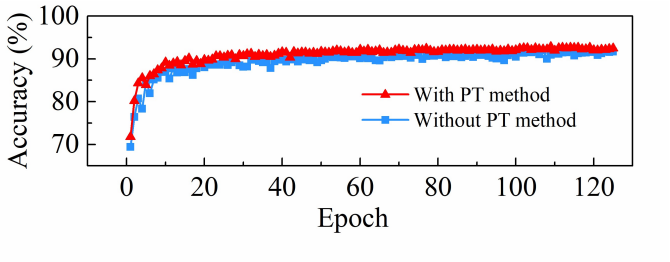

Fig. 4. Inference accuracy with epoch numbers.

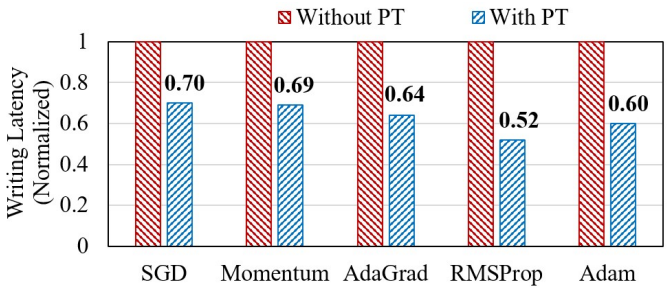

Fig. 5. Comparsion for writing latency.

B. Latency of writing process

As for the training stage, the latency of the memristor-based DNN accelerator includes reading and writing latency. For a given DNN structure, every iteration has stable reading

TABLE IV
WRITING ENERGY AND INFERENCE ACCURACY WITH NONLINEARITY

NL[a] (LTP/LTD)	Writing energy without PT (mJ)	Writing energy with PT (mJ)	Energy saved (%)	Inference accuracy without PT (%)	Inference accuracy with PT (%)
0 / 0	3.86	3.55	7.98	93.29	92.29
1 / -1	4.65	4.03	13.28	92.19	91.96
2 / -2	4.62	4.20	9.11	89.08	88.29
3 / -3	4.75	4.05	14.72	84.63	86.73

[a]NL represents the value of nonlinearity.

latency since the process of matrix-vector multiplication is performed using a parallel reading strategy. However, the system writes its weights row by row. Writing latency at each row is determined by the maximum number of writing pulses. For example, assumed the writing latency is 5 pulses without the PT method for the selected row, but it is only 1 pulse with the PT method, reducing the latency of pulses by 80%. In some extreme cases, suppose the 1 pulse change 1 unit of conductance for a memristor, and the maximum conductance is 200, theoretically, the maximum number of the needed writing pulses without the PT method is 200. However, with the PT method, the maximum number of writing pulses is still 1, reducing the latency of pulses up to 99.5%. The total normalized writing latency after 125 epochs without/with the PT method is shown in Fig. 5. They are decreased by 30%-48% for five algorithms, respectively. Thus, the PT method effectively reduces writing latency. Additionally, because of the PT method, every iteration has the same number of writing pules, the timing regularity of the system and the reliability of the system is significantly improved.

C. Nonlinearity and variation of memristors

If memristor is an ideal device, the conductance of a memristor will update proportionally to the number of input pulses. However, in reality, such a change is nonlinear [19]. In our experiments, LTP and LTD are labeled from +3 to -3 [19], [20] for nonlinearity metrics, which indicates the curve deviates from the ideal device LTP=LTD=0. The + and - signs are merely to label LTP and LTD, respectively. Taking the SGD algorithm as an example, the total writing energy without/with the PT method is listed in Table IV. The inference accuracies nearly keep same. But, writing energy without the PT method is higher than that with the PT method. Energy reductions are up to 14.72%. Thus, even with the nonlinear property of memristors, the PT method still effectively reduces writing energy.

What is more, variations for ON/OFF ratio, minimum and maximum conductance, cycle-to-cycle, and device-to-device always exist in the memristor-based DNN accelerator. To further verify the PT method, AdaGrad algorithm is taken as an example and investigated with these variations following standard/Gaussian distribution N (μ, σ). For Variations 1 and 2 in Table V, ON/OFF ratios are configured as 17 and 15. σ of the minimum conductance, maximum conductance, device-to-device (subjects to N (NL, σ) distribution), and cycle-to-cycle variation are set to 5.0%, 5.0%, 0.5, 1.0%, and 15.0%, 15.0%, 1.4, 2.5%, respectively [20]. Table V lists results of experiments under different circumstances. From Table V, it concludes that the PT method is still efficient to reduce energy consumption and writing latency in the DNN accelerator, even with many variations.

TABLE V
ENERGY, ACCURACY, AND LATENCY WITH VARIATIONS

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

TABLE VI
INFERENCE ACCURACY AND STANDARD DEVIATION WITH DIFFERENT FAILURE RATES

	Mean[a]		Standard deviation[a]	
Failure Rate	Without PT	With PT	Without PT	With PT
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 in 500 random cases.

D. Failure and aging effect

Typically, the failure rate $<$10% is required in manufacture [24]. In order to evaluate the DNN accelerator with the influence of failure, 5%-15% of the fault in the crossbar array are considered, as shown in Fig. 6 [25]. Fault memristors are placed at random positions in the crossbar array. The fault ratio of the stuck at 0 and 1 is 1:5.2 [24]. Taking the SGD algorithm as an example, Table VI lists the accuracy with the failure. From results of the mean and standard deviation that are obtained from 500 random cases of each failure rate, the DNN accelerator with the TP method still has accuracy improvement as compared with that without the TP method.

Aging effect also exists in memristor array. After a given times of programming, the tunability of conductance in a memristor deviates from the expected state, which is named aging effect, and it limits the lifetime of the DNN accelerator [26]. The conductance is assumed to drift towards different final states, or randomly drift, based on different various drift rates, which are equivalent to conductance drifts different amounts over 10 years, respectively [27]. Taking the SGD algorithm as an example, the accuracies with the aging effect

is shown in Fig. 7. The parameters regarding precision (P), recall (R), and F1 score are listed in Table VII.

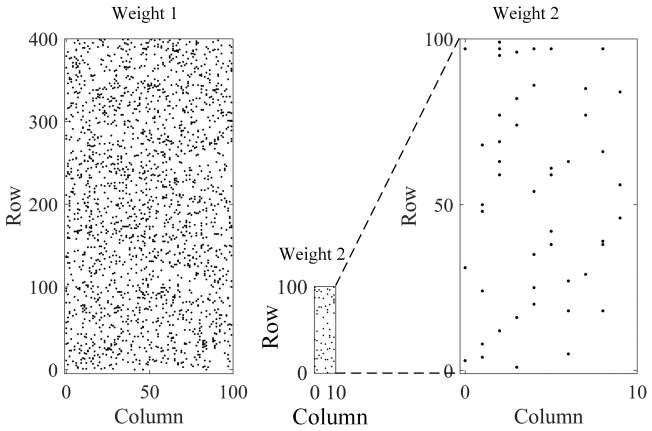

Fig. 6. 5% Fault memristors in crossbar array.

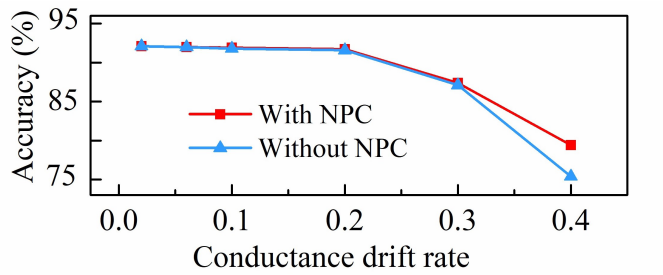

Fig. 7. Inference accuracy with different drift ratios

TABLE VII
PARAMETERS REGARDING PRECISION (P), RECALL (R), AND F1

	Without PT			With PT		
Class	**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
Avg.	0.773	0.741	0.737	0.804	0.794	0.793

[a]0.4 conductance drift ratios.

In addition, the endurance of a memristor is one limitation for high frequency writing in a DNN accelerator [28], [29]. The PT method extremely saves the number of writing pulses, as shown in Fig. 5.

Therefore, the proposed PT method is still effective with failure and aging circumstances and benefits the cycling endurance performance of a memristor.

E. PT method with different architectures and database

The PT method for different architectures and database is also considered. Taking SGD algorithm as an example, Fig. 8 shows different hidden layers of DNNs. As expected, the inference accuracy with the PT method is higher than that without the PT method. However, the leakage power is increased when enlarging hidden layer. The leakage power with the PT method is a little higher (<10%) than that without the PT method because multiplexors are added. Furthermore, VGG-8 architecture and CIFAR-10 database is also used to verify the PT method as listed in Table VIII. The accuracy difference without/with the PT method is as small as 0.8%. Therefore, the PT method does not much hurt inference accuracy [23]. However, as expected, the PT method respectively reduces latency up to 46.00% and energy consumption up to 16.67% (Latency is normalized to that without the PT method).

TABLE VIII
ENERGY, ACCURACY, AND LATENCY WITH VGG-8 AND CIFAR-10

	Accuracy	**Latency**	**Energy**
With TP method	90.30%	0.54	0.25J
Without TP method	91.10%	1.00	0.30J
Difference	0.80%	46.00%	16.67%

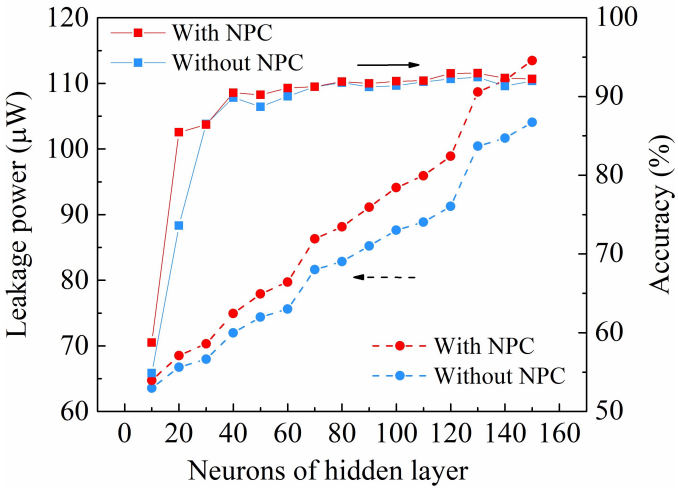

Fig. 8. Inference accuracy and leakage power with different hidden layers.

F. Comparison with the-state-of-art

The proposed PT method is a efficient method for the high performance and low energy online learning hardware design. As listed in Table IX, as compared with the state-of-art, the PT method does not need additional materials and algorithms to save energy.

TABLE IX
COMPARISON WITH THE STATE-OF-ART

Items	**[9]**	**[11]**	**[12]**	**[13]**	**[21]**	This work
I[a]	×	×	×	√	×	√
I[b]	√	√	√	×	√	√

[a]Without new material or structure.[b]Without extra algorithm.

CONCLUSION

In this paper, the PT method is proposed to improve energy efficiency and timing regularity of the memristor-based DNN accelerator, and it is verified using the fabricated device and NeuroSim platform (Device and NeromSim are detailed in Section III-B). Different with the traditional algorithm-based technology, the PT method combines hardware and algorithm implementation to optimize the pulse distributions and energy consumption in the system, avoiding additional complex peripheral circuits. Specifically, it significantly reduce number of pulses, but not change the original pulse width in every weight update. DNN accelerator with different architectures and algorithms under nonlinearities, variations, failures, and aging effect have been evaluated. It concludes: 1) The PT method realizes low energy consumption. 2) Since the pulse number for each weight update is truncated to 1, the PT method effectively reduces writing latency. 3) The PT method also improves timing regularity. Furthermore, the PT method is a general and adaptive method for any memristor-based DNN accelerator.

REFERENCES

[1] J. Edstrom, Y. Gong, A. A. Haidous, B. Humphrey, M. E. McCourt, Y. Xu, J. Wang, and N. Gong, "Content-Adaptive Memory for Viewer-Aware Energy-Quality Scalable Mobile Video Systems," IEEE Access, vol. 7, pp. 47479-47493, 2019.

[2] D. Strukov, G. Snider, D. Stewart, and R. Williams, "The missing memristor found," Nature, vol. 453, no. 7191, pp. 80-3, 2008.

[3] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on memristive systems," Nature Electronics, vol. 1, no. 1, pp. 22-29, 2018.

[4] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves, Z. Li, J. P. Strachan, P. Lin, Z. Wang, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, "Analogue signal and image processing with large memristor crossbars," Nature Electronics, vol. 1, no. 1, pp. 52-59, 2017.

[5] J. J. Kim, B. Cho, K. S. Kim, T. Lee, and G. Y. Jung, "Electrical characterization of unipolar organic resistive memory devices scaled down by a direct metal-transfer method," Advanced Materials, vol. 23, no. 18, pp. 2104-2107, 2011.

[6] 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 Transactions on Big Data, vol. 5, no. 1, pp. 95-105, 2019.

[7] D. Ielmini, "Modeling the Universal Set/Reset Characteristics of Bipolar RRAM by Field- and Temperature-Driven Filament Growth," IEEE Transactions on Electron Devices, vol. 58, no. 12, pp. 4309-4317, 2011.

[8] 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 Transactions on Electron Devices, vol. 56, no. 2, pp. 193-200, 2009.

[9] S. Li, W. Chen, Y. Luo, J. Hu, P. Gao, J. Ye, K. Kang, H. Chen, E. Li, and W.-Y. Yin, "Fully Coupled Multiphysics Simulation of Crosstalk Effect in Bipolar Resistive Random Access Memory," IEEE Transactions on Electron Devices, vol. 64, no. 9, pp. 3647-3653, 2017.

[10] S. Kim, S. J. Kim, K. M. Kim, S. R. Lee, M. Chang, E. Cho, Y. B. Kim, C. J. Kim, U. Chung, and I. K. Yoo, "Physical electrothermal model of resistive switching in bi-layered resistance-change memory," Scientific Reports, vol. 3, pp. 1680, 2013.

[11] Y. Luo, W. Chen, M. Cheng, and W.-Y. Yin, "Electrothermal Characterization in 3-D Resistive Ran-dom Access Memory Arrays," IEEE Transactions on Electron Devices, vol. 63, no. 12, pp. 4720-4728, 2016.

[12] D. Wang et al., "Fully Coupled Electrothermal Simulation of Large RRAM Arrays in the "Thermal-House"," IEEE Access, vol. 7, pp. 3897-3908, 2019.

[13] Y. Li, H.-H. Shen, C. Li, and F. Zhang, "An efficient parity rearrangement coding scheme for RRAM thermal crosstalk effects," in 2017 IEEE 12th Inter-national Conference on ASIC (ASICON), 2017, pp. 20-23.

[14] Z. Wang, S. Joshi, S. Savel'ev, W. Song, R. Midya, Y. Li, M. Rao, P. Yan, S. Asapu, and Y. Zhuo, "Fully memristive neural networks for pattern classification with unsupervised learning," Nature Electronics, vol. 1, no. 2, pp. 137, 2018.

[15] J. Fu, Z. Liao, and J. Wang, "Memristor-Based Neuromorphic Hardware Improvement for Privacy-Preserving ANN," IEEE Transactions on Very Large Scale Integration Systems, vol. 27, no. 12, pp. 2745-2754, 2019.

[16] M. Ponti, L. Ribeiro, T. Nazare, T. Bui, and J. Collomosse, "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), 2017, pp. 17-41.

[17] N. Buduma, and N. Locascio, "Fundamentals of deep learning: Designing next-generation machine intelligence algorithms," O'Reilly Media, Inc, 2017

[18] P. Y. Chen, X. Peng, and S. Yu, "NeuroSim: A Circuit-Level Macro Model for Benchmarking Neuro-Inspired Architectures in Online Learning," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 12, pp. 3067-3080, 2018.

[19] J. Fu, Z. Liao, N. Gong, and J. Wang, "Mitigating Nonlinear Effect of Memristive Synaptic Device for Neuromorphic Computing," IEEE Journal on Emerging and Selected Topics in Circuits and System, vol. 9, no. 2, pp. 377-387, June 2019.

[20] J. Fu, Z. Liao, N. Gong, and J. Wang, "Linear Opti-mization for Memristive Device in Neuromorphic Hardware," In 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2019, pp. 453-458.

[21] P. Sun, N. Lu, L. Li, Y. Li, H. Wang, H. Lv, Q. Liu, S. Long, S. Liu, and M. Liu, "Thermal crosstalk in 3-dimensional RRAM cross-bar array," Scientific Reports, vol. 5, no. 1, pp. 1-9, 2015.

[22] S. Jo, T. Chang, I. Ebong, B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," Nano Letters, vol. 10, no. 4, pp. 1297-1301, 2010.

[23] X. Peng, S. Huang, Y. Luo, X. Sun, and S. Yu, "DNN+NeuroSim: An End-to-End Benchmarking Framework for Compute-in-Memory Accelerators with Versatile Device Technologies," 2019 IEEE International Electron Devices Meeting (IEDM), 2019, pp. 32.5.1-32.5.4.

[24] C.-Y. Chen, H.-C. Shih, C.-W. Wu, C.-H. Lin, P.-F. Chiu, S.-S. Sheu, and F. T. Chen, "RRAM defect modeling and failure anal-ysis based on march test and a novel squeeze-search scheme," IEEE Transactions on Computers, vol. 64, no. 1, pp. 180-190, 2014.

[25] 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 Transactions on Big Data, vol. 5, no. 1, pp. 95-105, 2017.

[26] A. Irmanova, A. Maan, A. James, and L. Chua, "Analog Self-timed Programming circuits for Aging Memristors," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no.4, pp. 1133-1137, 2020.

[27] P.-Y. Chen and S. Yu, "Reliability perspective of resistive synap-tic devices on the neuromorphic system performance," IEEE International Reliability Physics Symposium (IRPS), 2018, pp. 5C-4.

[28] C. Liu, M. Hu, J. P. Strachan, and H. Li, "Rescuing memristor-based neuromorphic design with high defects," 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), 2017, pp. 1-6.

[29] 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," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 40, no. 11, pp. 2306-2319, 2021.

[30] J. Fu, Z. Liao and J. Wang, "Cycle-to-cycle Variation Enabled Energy Efficient Privacy Preserving Technology in ANN," 2020 IEEE 33rd International System-on-Chip Conference (SOCC), 2020, pp. 66-71.

[31] Z. Liao, J. Fu, and J. Wang, "Ameliorate Performance of Memristor Based ANNs in Edge Computing," IEEE Transactions on Computers, vol. 70, no. 8, pp. 1299-1310, 2021.

[32] D. Chen, J. Edstrom, X. Chen, W. Jin, J. Wang, and N. Gong. "Data-Driven Low-Cost On-Chip Memory with Adaptive Power-Quality Trade-off for Mobile Video Streaming," 2016 International Symposium on Low Power Electronics and Design (ISLPED). 2016, pp. 188-193.

[33] S. A. Pourbakhsh et al., "Sizing-priority based low-power embedded memory for mobile video applications," 2016 17th International Symposium on Quality Electronic Design (ISQED), 2016, pp. 1-5.

[34] D. Chen et al., "Viewer-Aware Intelligent Efficient Mobile Video Embedded Memory," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 4, pp. 684-696, April 2018.