Insights and Optimizations on IR-drop Induced Sneak-Path for RRAM Crossbar-based Convolutions

Yujie Zhu, Xue Zhao, Keni Qiu
College of Information Engineering, Capital Normal University, Beijing, China

Abstract— RRAM crossbar structure has been proposed to accelerate the convolution computation neural networks because its current-mode weighted summation operation intrinsically matches the dominant multiplication-and-accumulation (MAC) operations. However, there is an inevitable IR-drop problem with the RRAM crossbar, which may introduce sneak-path and thus reduce the accuracy of neural network algorithms and the system reliability. This work addresses the sneak-path problem caused by the IR-drop in a RRAM crossbar. We first present the characteristics of variation distribution of the sneak-path through numerous experiments, taking into account RRAM cell resistance, input voltage, and cell location in a crossbar. Then we propose optimization strategies from the hardware and software perspectives respectively to mitigate the variations resulting from sneak-path. The experimental results show that the proposed methods can compensate the accuracy of algorithms.

I. Introduction

As the core driving force of the new industrial transformation, artificial intelligence (AI) has spawned a large number of new applications. These have led to innovations at the software level, as well as the continuous development of hardware in the field of intelligence. The dominant computing of deep learning is to perform a lot of convolution operations on massive data. Research shows that the multiplication-and-accumulation (MAC) operations of CNN algorithms account for more than 90% of the entire operations [1]. As the network continues to deepen, it brings opportunities and challenges to the development of hardware [2]. In non-volatility devices, RRAM has the characteristics of non-volatile, high density, extremely low leakage power etc., which can well meet the demand of next-generation memory [3]. RRAM crossbar has been widely studied to accelerate neural network because the structure is suitable for the MAC operations [4] [5]. The advantage is that the current passing through a resistor is given by the product of the voltage across it and the conductance. And we generally configure all the kernels on RRAM crossbar networks.

However, RRAM cells are facing several inevitable technological challenges, such as the parametric variability, fabrication defects, stochastic programming properties of memristors [6] [7] and IR-drop [8]. In the RRAM crossbar, the IR-drop caused by wire resistance will induce unexpected current branch, that is, sneak-path. With the increase of crossbar size and the accumulation in the calculation, the sneak-path variation will become severe and impose a impact on the algorithm accuracy. In reliability sensitive fields, such as health care and auto self-driving, any loss of accuracy can have a fatal effect in high-reliability scenario. Therefore, it is critical to reduce the variation as much as possible in order to meet the high accuracy requirement.

In this paper, it is observed that the distribution of sneak-path is highly related RRAM resistance, cell position and input voltage in a RRAM crossbar. The observations can be described in the following three aspects: 1) As the resistance of a RRAM cell goes higher, the sneak-path variation gets smaller; 2) As the distance from the input voltage gets larger, the sneak-path variation turns greater; and 3) Large input voltage can bring the positive growth of the sneak-path variation. On the contrary, small input voltage can bring in the negative growth of the sneak-path variation.

Based on the above insights on the RRAM variation issue, a current amplification compensation approach and an input voltage resetting approach are proposed from the perspectives of hardware and software levels respectively. By this means, the IR-drop induced variation can be mitigated and thus the algorithm accuracy and the system reliability can be improved.

In summary, this paper makes the following contributions.

- The insights of the sneak-path distribution in RRAM crossbar are presented through experiments.
- Two optimization strategies are proposed at software and hardware levels to mitigate the effect of RRAM variation.
- The experiments validate that the accuracy can be improved by the proposed techniques.

The remainder of this paper is organized as follows. Section II introduces the background of relevant research work on sneak-path. Section III presents our observations on the distribution of sneak-path. Based on the observed distribution rules, we propose the optimization strategies to mitigate sneak-path variation in Section IV and discuss the experimental results in Section V. Finally, Section VI concludes this paper.

II. Sneak-Path Background

A. MAC operation on RRAM crossbar

RRAM combines a cross-point structure with a horizontal bit line and a vertical word line to form a RRAM crossbar. Each intersection in the cross array is a memory cell and the network weights are configured in the cell conductance. Fig. 1 shows the dot-product operation in a RRAM crossbar. When performing a convolution operation, an electrical signal V is input at the horizontal bit lines, and the output current

978-1-7281-4123-7/20/$31.00 ©2020 IEEE

magnitude through each cross node is calculated as I=V×G, where G represents the operational node conductance, and the output current converges at the end of the vertical word line. In this way the MAC operation in neural networks can be accomplished in parallel with a RRAM crossbar naturally.

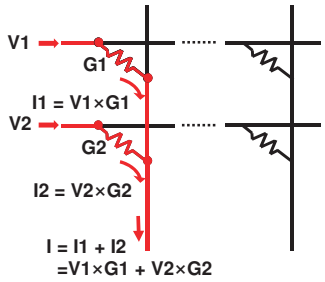


Fig. 1: Dot-product operation schematic at RRAM crossbar [4].

B. Related work on the sneak-path problem

The RRAM crossbar structure has been widely studied as an acceleration platform for neural networks due to its structural property. However, in the RRAM crossbar, the voltage is divided due to the wire itself carrying the resistance, so that the voltage distribution in the circuit cannot be consistent as expected and results in an IR-drop problem. This will breed unexpected current branches, introducing the so called sneak-path problem. The shunt current of the sneak-path will further lead to the reduction of the accuracy of the calculation results, affecting the reliability of the system.

Recently, there are many researches addressing the sneak-path issue introduced by IR-drop which can be divided into two categories.

At the circuit level, putting selection device in the memory cell to form a new array is studied, such as transistor-one-memristive device (1T1M) and one-diode-one-memristive device (1D1M) structures [9]. E, Linn et al. proposed a complementary resistive switch that consists of two antiserial bipolar memory cells [10]. Although these techniques can somewhat avoid sneak-path, the integration of circuits is reduced and the manufacturing difficulty is greatly increased.

At the software level, researchers have proposed algorithms to optimize the sneak-path analysis. Woorham Bae et al. proposed a sneak-current compensation port, which collects sneak current information from multiple unselected cells and eliminate the influence in readout port, but this method is only for 1T1R and 1S1R (one-seletor and one-resistor) cell structure [11]. Chul-Moon Jung et al. proposed a two-step write scheme to reduce the sneak path based on the complementary memristor array [12]. However, those studies only address at specific circuit structures and are not applicable to commonly used structures such as 1R RRAM crossbar.

Addressing the basic 1R RRAM crossbar circuit structure, this paper first analyzes the distribution of the sneak-path variation from the system perspective and obtains the factors affecting the sneak-path. According to the obtained laws, this paper optimizes the sneak-path variation from both the circuit level and the software level perspectives.

III. Observations

We built different RRAM crossbars to simulate the sneak-path by HPspice for the sizes of 3×3, 32×32, 64×64. Fig. 2 is the schematic diagram of the peripheral circuit built before and after the variation. In the experiments, parameter settings are shown in Tab. I, the RRAM resistances fall into two states which are high resistance and low resistance assigned by 1MΩ and 10KΩ respectively. The input voltage is set in the range of 0V to 1.2V, and the different voltage are arbitrarily distributed. The IR-drop is simulated by setting the wire resistance value of 25Ω between two adjacent RRAM cells. The variation due to the effect of IR-drop in this paper refers to the current difference between the real value and the ideal one. We used Python scripts to analyze the generated parameter files, and calculated the sneak-path current variation distribution in the circuits.

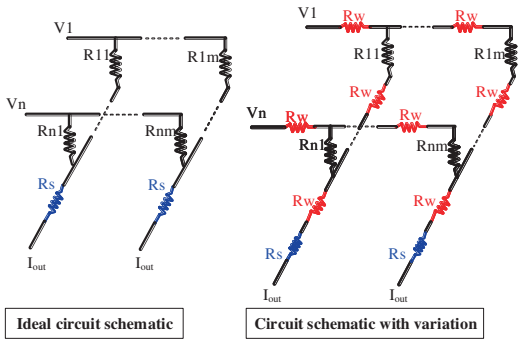


Fig. 2: Schematic diagram of peripheral circuit

TABLE I: parameter assumptions [13]

Wire Resistance: 25Ω	Input voltage: 0∼1.2V
RL/RH:10KΩ/1MΩ	AMP: 4.8mW
ADC: 8-bit, 35mW	DAC: 8-bit, 40mW

Fig. 3 shows the RRAM resistance assignment of a 64×64 RRAM crossbar. The horizontal and vertical coordinates represent 64 row inputs and 64 column outputs respectively. The white dots represent a high RRAM resistance of 10KΩ and the black dots represent a low RRAM resistance of 1MΩ. They are arbitrarily sparse-distributed in the circuit. According to the sparsity of the neural network, the zero value accounts for a large proportion in the weight distribution of network. To reduce the loss of system power, this method, which is mapping the high-resistance by the low-weight value, is utilized in RRAM crossbar. This method is also adapted to this paper. The measurement unit of the right bar is KΩ.

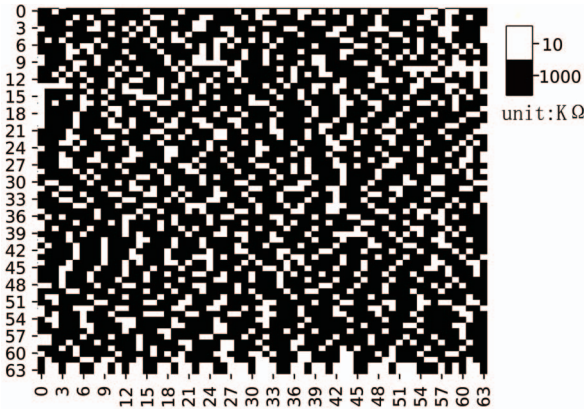


Fig. 3: RRAM resistance distribution in 64×64 crossbar.

Fig. 4 is a heat map by simulating the sneak-path variation with the metric of current. The variation analysis is depicted in Fig. 5(a), which is extracted in the horizontal direction of the 63rd and 64th line in Fig. 4 where the inputs are 0V and 1.2V respectively and their RRAM resistance values with the same distribution. We can quickly obtain that when the RRAM cell is set with high-resistance state, the variation is minimized regardless of whether the input is high-voltage state or low-voltage state. What is more, it is observed that the current variation is positive in the circuit when the input voltage is 1.2V, while the current variation is negative when the input voltage is 0V. And the variation gradually increases in the positive direction of the X-axis. Fig. 5(b) depicts the variation distribution of the 61st column of Fig. 4. It can be observed that the peak of the curve of the variation is present when the input voltage is at the maximum and minimum values. When the RRAM cell is set to the high resistance state, indicating by a red column in the Fig. 5(b), the variation approaches zero no matter the voltage input.

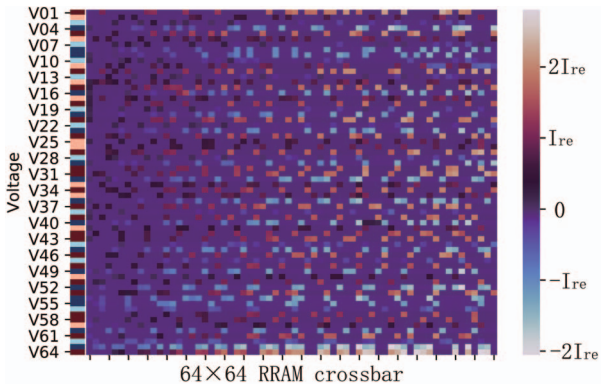
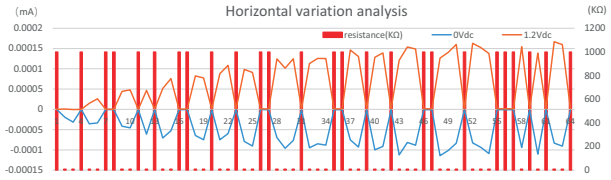
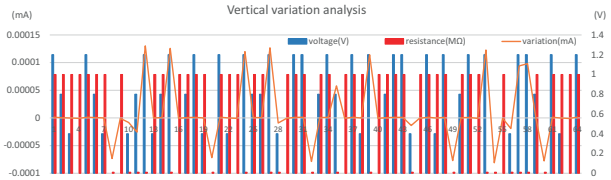


Fig. 4: The sneak-path variation distribute in 64×64 crossbar, I_{re} is defined as reference: I_{re}=0.1mA.

On top of the circuit simulation and data analysis, we can obtain the following insights:



(a) Horizontal variation analysis of line 63 and 64 in 64×64 crossbar.



(b) Vertical variation analysis of column 61 in 64×64 crossbar.

Fig. 5: The horizontal and vertical variation of observation experiments were analyzed

Law I: Considering the RRAM resistance, the current variation is significantly lower in the RRAM cell with higher resistance.

Fig. 3 shows the RRAM resistance distribution in the circuit, and the black point indicates the high RRAM resistance. Because the sneak-path is restrained by the high resistance RRAM cell, we can clearly observe that the sneak-path variation is relatively small in the high resistance state as shown in Fig. 4.

Law II: Considering the location of RRAM, the current variation presents a gradual increasing as the cell distance in a row from the input increases.

In Fig. 4, The color shows a trend to fade in the horizontal direction. The lighter the color is, the greater the variation exists. The two tendencies of color change respectively represent positive and negative variations.

Law III: Considering the influence of input voltage, the current variation is positive type when inputting high voltage while negative type when inputting low voltage.

The variation types hugely depend on input voltages. In Fig. 4, the bar on the left side represents 64 input voltages. The darker red the color is, the larger the voltage value is. The darker blue the color is, the smaller the voltage value is. We observe that the red bar fades in row-wise direction as high voltages input. This means that the actual current passing the RRAM cell is positively larger than the ideal current value, and its value gradually increases in the horizontal direction. In contrast, the blue bar horizontally fades in a row as low voltages input. That is, the current value actually passing the RRAM cell is negatively smaller than the ideal current value, and the variation gradually increases.

The above laws reveal the distribution of the sneak-path variation observed by the circuit simulation. In order to observe the impact of variation on accuracy, we establish the LeNet-5 neural network to train and predict MNIST data. The values following the normal distribution were picked up

as the random variation value in the circuit simulation. It is known that the variation value was accumulated as the calculation process is going on. In Fig. 6, we can see that the network accuracy loss is significant after introducing variations. That is, as the variation accumulates, the accuracy presents a gradually decreasing trend.

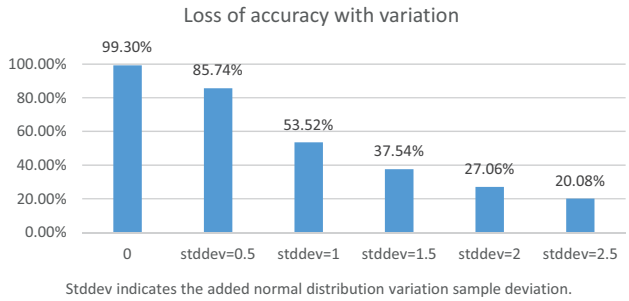


Fig. 6: The impact from variations on classification accuracy for LeNet-5 based on MNIST data set.

IV. Proposing Optimization Strategies

The observations reveal how the RRAM cell resistance and position, and input voltage impact the current variation. Motivated by the variation distribution patterns, we propose two compensation methods from software and hardware levels respectively to effectively mitigate the variations.

A. Hardware-level optimization: adjusting the sense amplifier (SA) magnification

As Law II indicates, the current variation increases horizontally at row wise as the array size increases. Therefore, the output variation of each column increases at left-to-right direction in the crossbar. It is known that the current accumulates of each column in the crossbar circuit structure. The analog output signals are then converted into digital data through SA and ADC. Understanding the MAC computation process and the fact of variation vs. position, we are motivated to use different magnifications for different columns to make the current variation compensated at the output.

In Fig. 7, the ideal current without variation is denoted by the green solid line. Actually the current variation increases along the row-wise from left to right where the green line turns red and deepens gradually. We propose a strategy by adjusting the SA signal with a coefficient of X_i for column i as shown in Equ. 1.

$$x_i = \frac{Curr^{out-ideal}}{Curr^{out-var}} \tag{1}$$

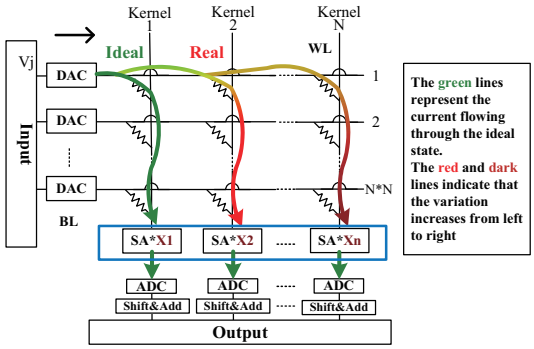


Fig. 7: The adjustment of sense amplifier magnification.

The coefficient X_i represents the ratio of the output currents with or without the variation. In this way, different magnifications are assigned for different outputs. The output current variation is approximately restored to the ideal state after adjusting, thereby reducing the loss of the output precision during the convolution procedure.

B. Software-level optimization: input voltage resetting

As Law III indicates, the variation increase caused by high voltage input and the variation decrease caused by low voltage input can balance the final variation. It is defined $function_+$ as the positive variation function, $function_-$ as the reduced negative variation function brought by the input low voltage as shown in Equ. 2. The function inputs are the input voltage and resistance of the current position, and the output denotes the node variation. Our goal is to make the positive and the negative variations counteract each other as much as possible. Let $S_{var,min}$ represent the global minimum variation. The values of VH_i and VL_i are bounded between minimum and maximum voltages, as shown in *Constraint 1 and 2*. As Law I shows that the high resistance has a restraining effect on the sneak-path, that is, the variation is small, we assume that the variation value is 0 at the high resistance value. Note that $function_+$ and $function_-$ derived by a regression method.

$$S_{var,min} = \sum_{j=1}^{n}\Big(\sum_{i=1}^{n_{VH}} function_+(VH_i, R_{i,j}) + \sum_{i=1}^{n_{VL}} function_-(VL_i, R_{i,j})\Big) \tag{2}$$

$$\begin{cases} Constraint\ 1 : VH_i \in (VH_{min}, VH_{max}), \\ Constraint\ 2 : VL_i \in (VL_{min}, VL_{max}), \\ Constraint\ 3 : \\ \quad if\ RH : function_+(VH_i) = function_-(VL_i) = 0 \end{cases}$$

Algorithm 1 describes the flow of how to well counteract the positive and negative variation. Step 1 is defined as a cyclic function to calculate the global variation. The global variation value in the circuit is obtained through nested-loop algorithm in line 2-10. Step 2 can find the optimal voltage inputs, when the global variation value is greater than the tolerable threshold. Here we set the tolerable error threshold to 10^{-8}.

Algorithm 1 Input Voltage Resetting.

Input: Input voltage matrix $V[V1...Vn]$;
The value range of high voltage input (VH_{min}, VH_{max});
The value range of low voltage input (VL_{min}, VL_{max});
Crossbar size: n rows and m columns;
Weight distribution matrix $R[n][m]$;
High resistance value RH;
Output:
step 1: Calculate the global variation S_{var};
for $i = 1$ to n **do**
  for $j = 1$ to n **do**
    if RH: $function_+(V_i, R_{i,j})$=$function_-(V_i, R_{i,j})$=0
    if $VH_{min} < V_i < VH_{max}$:
      S_{var}+=$function_+(V_i, R_{i,j})$
    if $VL_{min} < V_i < VL_{max}$:
      S_{var}+=$function_-(V_i, R_{i,j})$
  end for
end for
return S_{var};
step 2: Find the optimal voltage inputs;
while $|S_{var}| < 10^{-8}$ **do**
  // the variation is less than the threshold
  for $i = 1$ to n do **do**
    if $S_{var} > 0$ **then**
      // high voltage lower, low voltage higher
      if $VH_{min} < V_i < VH_{max}$: V_i=(VH_{min}+V_i)/2
      if $VL_{min} < V_i < VL_{max}$: V_i=(V_i+VL_{max})/2
    end if
    if $S_{va} < 0$ **then**
      // high voltage higher, low voltage lower
      if $VH_{min} < V_i < VH_{max}$: V_i=(V_i+VH_{max})/2
      if $VL_{min} < V_i < VL_{max}$: V_i=(VL_{min}+V_i)/2
    end if
    step 1:Calculate the resetting variation
    if $|S_{var}| > 10^{-8}$
      break
  end for
end while
return The resetting input matrix V;

If $S_{var} > 0$, it means that the variation type is positive. There are two methods to eliminate the positive variation. First, the positive value can be reduced by decreasing the high input voltage. Second, the negative value can be increased by increasing the low input voltage. On the contrary, if $S_{var} < 0$, it means that the variation type is negative. The solution is to compensate the negative variation. In all, the positive value can be enhanced by adjusting the high input voltage value meanwhile the negative variation can be reduced by lowing the input voltage. When the input voltage value is adjusted, we call Step 1 to calculate the global variation iteratively. By Step 2, inputs are updated until the variation value of S_{var} is minimized.

Researchers previously proposed different optimization methods addressing variations and defects in the crossbar circuit, such as key weight compensation and network weight remapping (some important weights are compensated, and the adjustment of weights with little influence is ignored [14], according to the variation distribution law, the network structure is adjusted to map the important weight to the place with small variation and the unimportant weight to the place with large variation, so as to reduce the fault output [15]). Those methods are coarse-grained and work at the software level without considering fine adjustment. The optimizations proposed in this work are fine-grained solutions for compensation, specifically compensating the variation in each unit or convolutional calculation process. They can make the network calculation results more accurate by designing the compensation in software and hardware collaboratively. And moreover, the proposed hardware and software-level approaches are orthogonal, so they can be combined to conduct optimizations.

V. Experiments

RRAM crossbar is used as a convolution calculation of neural network due to its natural characteristics. However, it has brought about the inevitable influence of sneak-path since industrial manufacturing. Through experiments, we observed the distribution law of sneak-path related to the factors of position, input voltage and the memristor resistance. We accordingly propose optimization measures from software and hardware perspectives.

To evaluate our proposed optimizing strategies, we build a crossbar circuit simulation with HPspice. The circuit parameters are set as shown in Tab. I. The wire resistance value is set to 25Ω, the input voltage is 0V to 1.2V to indicate the voltage input from low to high, and the RL and RH are set to 10Ω and 1MΩ respectively to represent the high and low resistance states.

The experiment results show that as the size of crossbar increases, the average variation increases. In Tab. II, it can be seen the variation was significantly reduced after SA magnification adjustment and the variation of input voltage resetting was also reduced. For re-adjusting the SA amplification strategy, it is more prominent for the effect of the sneak-path variation optimization as the size of the crossbar increases. Because the larger the variation is, the more precise the magnification multiple is. The variation after optimization of the input voltage resetting strategy is reduced by nearly half compared to the original variation value. Since the impact of adjusting the input on each column in the crossbar is different, it is necessary to find the optimal group of input voltage to ensure that the overall variation is as small as possible.

TABLE II: comparison of variation and optimization results of different scales crossbars (the current variation unit: μA)

Crossbar size	Var(avg)	ad-SA[a](avg)	re-V[b](avg)
9 ×8	66.225	0.602	37.362
32×32	489.781	0.380	290.562
64×64	892.304	0.287	574.437

[a] ad-SA: the adaptive amplifier magnification.
[b] re-V: input voltage resetting.

We take MNIST handwritten digital input as a classic neural network test data set, consisting of 28×28 grayscale images composed of 60,000 training images and 10,000 test images. In order to evaluate the impact of optimization strategy on accuracy, we built a classic LeNet-5 and AlexNet network to train the MNIST dataset, and obtained the network

weight model as the crossbar model under our ideal conditions. The proposed method is mainly applied to the inference of applications. We modify the network by introducing weight loss to simulate the inaccuracy caused by the sneak-path variation on the crossbar. The variation is adjusted according to the input of the convolution.

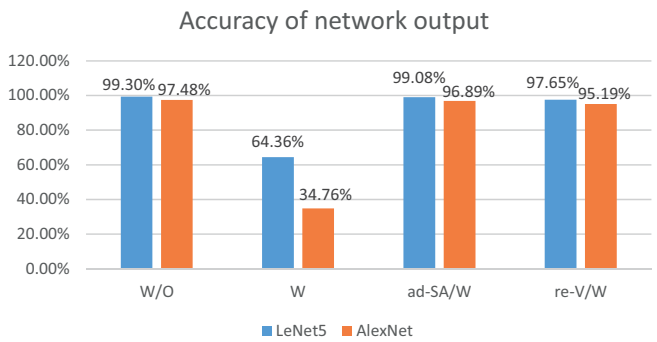


Fig. 8: The influence of sneak-path on accuracy and the improvement of accuracy after optimization

By simulating the loss of crossbar array, the neural network algorithm of MNIST is analyzed. In Fig. 8, the recognition accuracy of MNIST is 99.30% in the ideal case with LeNet5 training and inference. When the weight variation is added, the accuracy of LetNet-5 dropped by 34.94%, and the accuracy of AlexNet dropped by 62.72%. There are two layers of convolutional layers in the LeNet5 network, and five layers of convolutional layers in AlexNet. As the number of network layers increases, accumulated variation introduces more loss on precision. Applying the proposed two optimization methods for sneak-path variation, the accuracy of the applications have been significantly improved. The results of the SA adjustment strategy are close to the ideal situation. The software-level input re-adjustment approach can also achieve good results. In the future work, we will further optimize the algorithms to achieve better results.

VI. Conclusion

RRAM crossbars have been proposed to perform MAC operations in convolution calculations due to its structural characteristics. However, RRAM suffers from sneak-path problem caused by IR-drop variation. In order to reduce the sneak-path variation, this work first analyze the distribution of sneak-path. It is found to be highly related to RRAM cell resistance, input voltage, and cell location in a crossbar. The SA magnification adjustment and the input voltage resetting method are proposed at the hardware and software levels respectively. Experiments show that the proposed sneak-path optimization methods can reduce the calculation variation of the crossbar and thus improve the algorithm accuracy and system reliability.

Acknowledgements

This work is supported by the grants from National Natural Science Foundation of China [Project No.61872251], Beijing Advanced Innovation Center for Imaging Technology and Beijing Innovation Center for Future Chip. Thanks to Dr. Lixue Xia (Alibaba Group, lixue.xlx@alibaba-inc.com) for his valuable advice on this work. The corresponding author is Keni Qiu (qiukn@cnu.edu.cn).

References

[1] L. Xia, T. Tang, W. Huangfu, M. Cheng, X. Yin, B. Li, Y. Wang, and H. Yang, "Switched by input: Power efficient structure for RRAM-based convolutional neural network," in *DAC Design Automation Conference 2016 (DAC)*, 2016, pp. 125:1–125:6.

[2] C. Xu, D. Niu, N. Muralimanohar, R. Balasubramonian, T. Zhang, S. Yu, and Y. Xie, "Overcoming the challenges of crossbar resistive memory architectures," *IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, pp. 476–488, 2015.

[3] K. Liu, M. Zhao, L. Ju, Z. Jia, C. J. Xue, and J. Hu, "Design exploration for multiple level cell based non-volatile FPGAs," in *2017 IEEE International Conference on Computer Design (ICCD)*, 2017, pp. 257–264.

[4] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. W. Strachan, M. Hu, S. Williams, Srikumar, and Vivek, "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *ACM SIGARCH Computer Architecture News*, vol. 44, pp. 14–26, 2016.

[5] M. Cheng, L. Xia, Z. Zhu, Y. Cai, Y. Xie, Y. Wang, and H. Yang, "Time: A training-in-memory architecture for RRAM-based deep neural networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 38, no. 5, pp. 834–847, 2019.

[6] M. Hu, H. Li, Q. Wu, and G. S. Rose, "Hardware realization of bsb recall function using memristor crossbar arrays," in *Design Automation Conference 2012 (DAC)*, 2012, pp. 498–503.

[7] L. Xia, M. Liu, X. Ning, K. Chakrabarty, and Y. Wang, "Fault-tolerant training enabled by on-line fault detection for RRAM-based neural computing systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 38, no. 9, pp. 1611–1624, 2019.

[8] J. Liang, H., and P. Wong, "Cross-point memory array without cell selectors—device characteristics and data storage pattern dependencies," *IEEE Transactions on Electron Devices*, vol. 57, no. 10, pp. 2531–2538, 2010.

[9] V. S. Srinivasan, S. Chopra, P. Karkare, P. Bafna, S. Lashkare, P. Kumbhare, Y. Kim, S. Srinivasan, S. Kuppurao, S. Lodha, and U. Ganguly, "Punchthrough-diode-based bipolar RRAM selector by si epitaxy," *IEEE Electron Device Letters*, vol. 33, no. 10, pp. 1396–1398, 2012.

[10] E. Linn, R. Rosezin, C. Kügeler, and R. Waser, "Complementary resistive switches for passive nanocrossbar memories," *Nature materials*, vol. 9, pp. 403–6, 05 2010.

[11] T. S. Woorham Bae, Kyung Jean Yoon and B. Nikolić, "A variation-tolerant, sneak-current-compensated readout scheme for cross-point memory based on two-port sensing technique," *IEEE Transactions on Circuits and Systems II: Express Briefs (TCAS-II)*, pp. 1–1, 2018.

[12] C. Jung, J. Choi, and K. Min, "Two-step write scheme for reducing sneak-path leakage in complementary memristor array," *IEEE Transactions on Nanotechnology*, vol. 11, no. 3, pp. 611–618, 2012.

[13] K. Qiu, W. Chen, Y. Xu, L. Xia, Y. Wang, and Z. Shao, "A peripheral circuit reuse structure integrated with a retimed data flow for low power RRAM crossbar-based CNN," in *2018 Design, Automation & Test in Europe Conference&Exhibition (DATE)*, 2018, pp. 1057–1062.

[14] B. Liu, H. Li, Y. Chen, X. Li, T. Huang, Q. Wu, and M. Barnell, "Reduction and IR-drop compensations techniques for reliable neuromorphic computing systems," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2014, pp. 63–70.

[15] B. Liu, Hai Li, Yiran Chen, Xin Li, Qing Wu, and Tingwen Huang, "Vortex:variation-aware training for memristor x-bar," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2015, pp. 1–6.