micromachines

MDPI

Article

Synapse-Neuron-Aware Training Scheme of Defect-Tolerant Neural Networks with Defective Memristor Crossbars

Jiyong An, Seokjin Oh, Tien Van Nguyen and Kyeong-Sik Min *

School of Electrical Engineering, Kookmin University, Seoul 02707, Korea; sunday1903@kookmin.ac.kr (J.A.); ghj163@kookmin.ac.kr (S.O.); tiennv@kookmin.ac.kr (T.V.N.)
* Correspondence: mks@kookmin.ac.kr

Abstract: To overcome the limitations of CMOS digital systems, emerging computing circuits such as memristor crossbars have been investigated as potential candidates for significantly increasing the speed and energy efficiency of next-generation computing systems, which are required for implementing future AI hardware. Unfortunately, manufacturing yield still remains a serious challenge in adopting memristor-based computing systems due to the limitations of immature fabrication technology. To compensate for malfunction of neural networks caused from the fabrication-related defects, a new crossbar training scheme combining the synapse-aware with the neuron-aware together is proposed in this paper, for optimizing the defect map size and the neural network's performance simultaneously. In the proposed scheme, the memristor crossbar's columns are divided into 3 groups, which are the severely-defective, moderately-defective, and normal columns, respectively. Here, each group is trained according to the trade-off relationship between the neural network's performance and the hardware overhead of defect-tolerant training. As a result of this group-based training method combining the neuron-aware with the synapse-aware, in this paper, the new scheme can be successful in improving the network's performance better than both the synapse-aware and the neuron-aware while minimizing its hardware burden. For example, when testing the defect percentage = 10% with MNIST dataset, the proposed scheme outperforms the synapse-aware and the neuron-aware by 3.8% and 3.4% for the number of crossbar's columns trained for synapse defects = 10 and 138 among 310, respectively, while maintaining the smaller memory size than the synapse-aware. When the trained columns = 138, the normalized memory size of the synapse-neuron-aware scheme can be smaller by 3.1% than the synapse-aware.

Keywords: synapse-neuron-aware training; defect-tolerant neural networks; defective memristor crossbars; memristor defects; neuromorphic

Citation: An, J.; Oh, S.; Nguyen, T.V.; Min, K.-S. Synapse-Neuron-Aware Training Scheme of Defect-Tolerant Neural Networks with Defective Memristor Crossbars. *Micromachines* **2022**, *13*, 273. https://doi.org/10.3390/mi13020273

Academic Editor: Yao-Feng Chang

Received: 21 January 2022
Accepted: 6 February 2022
Published: 8 February 2022

1. Introduction

With the advent of 'Big Data', traditional CMOS-based computing platforms such as CPU, GPU, and FPGA are unable to match the demands of real-time data processing [1–3]. To address the limitations of CMOS digital systems, emerging computing circuits such as memristor crossbars have been investigated as potential candidates for significantly increasing the speed and energy efficiency of next-generation computing systems, which are required for implementing future AI hardware [4–7].

Unfortunately, manufacturing yield remains a serious challenge in adopting next-generation computing systems based on emerging devices due to the limitations of immature fabrication technology. Memristor crossbars, for example, have various fabrication-related issues, such as defects and variations, despite demonstrating superior computing performance and higher energy efficiency than standard CMOS computing systems. Defects in memristor crossbars are commonly referred to as stuck-at-faults, which occur when malfunctioning devices become caught in the High Resistance State (HRS) or the Low Resistance State (LRS) [8–11]. Even when programming pulses are delivered to memristor

cells that become stuck at faults such as HRS or LRS, they are unable to modify their resistance states. Stuck-at-faults are quite common and can be found in the majority of memristor crossbars. For example, it was reported that approximately 10% and 4% of memristor cells were measured as stuck-at-LRS and stuck-at-HRS, respectively [12].

Let us now look at how malfunctioning memristor devices can impair the performance of neural networks based on emerging devices like memristor crossbars. Vector Matrix Multiplication (VMM) is a fundamental computing function used in neural network's training and inference. The VMM process can be achieved in memristor crossbars, where vector and matrix multiplication is performed utilizing the memristor's voltage-current relationship based on Ohm's equation [5,13–16]. The most significant advantage of VMM performed on the memristor crossbar is that the VMM processing capabilities may be enhanced in parallel by adding more columns to the crossbar. Because of its expandable and parallel computing design, the memristor crossbar is well suited for diverse neural network applications processing 'Big Data.' Unfortunately, even a single memristor defect can have a large impact on the VMM calculation, because a fault like stuck-at-LRS can drastically raise the column current of the crossbar to which the damaged cell belongs [12]. The unintentional neuron activation caused by defective cells can drastically decrease the training and inference accuracy of neural network based on emerging devices [12,16–20].

Two types of defect-aware schemes can be considered to compensate for performance degradation caused by memristor defect cells such as stuck-at-faults. The first is the synapse-aware scheme, which trains memristor crossbars while taking into account the types and locations of synapse defects [20–22]. In this case, a defect map containing information on both defect types and locations should be used during the training of memristor crossbars. The neural network's synapses and neurons are mapped on a real memristor crossbar while taking defect types and locations into account. The degradation of neural network's performance can thus be minimized by defect-aware training of real memristor crossbars with defects [23–26].

The second is the neuron-aware scheme. If some neurons are connected to columns with a high number of faulty cells, they are referred to as 'severely-defective columns' in the neuron-aware scheme. During the training of memristor crossbars, the severely defective columns are not permitted to participate in neural network's operations such as training and inference [22]. By doing so, the neural network can prevent the defective columns from affecting the neuron activation during the training and inference.

Figure 1a indicates a conceptual diagram of synapse-aware scheme, where a memristor crossbar with defects and its defect map are shown in left and right, respectively. Here, W means a synaptic weight. For ternary neural networks, the weight can be either -1, $+1$, or 0. W = -1, $+1$, and 0 are represented with solid circle, open circle, and open box, in Figure 1a, respectively [26]. In the crossbar, the defect cells are represented with red flames. The defect map for the synapse-aware scheme is also shown in right in Figure 1a. The red and black boxes represent faulty and normal cells, respectively. The fault types can be stuck-at-(+1), stuck-at-(-1), and stuck-at-(0), as indicated in Figure 1a. The defect location can be represented with the faulty cell's row and column numbers stored in the defect map's memory.

Figure 1b shows a conceptual diagram of the neuron-aware scheme. Like Figure 1a, a defective memristor crossbar and its map of defective columns are shown in left and right, respectively. In Figure 1b, the defect map contains only the information of defective columns. The red and black boxes represent the defective and normal columns, respectively. The defect map of the neuron-aware scheme requires much smaller memory than the synapse-aware, because only the defective column's locations are stored in the defect map's memory in Figure 1b.

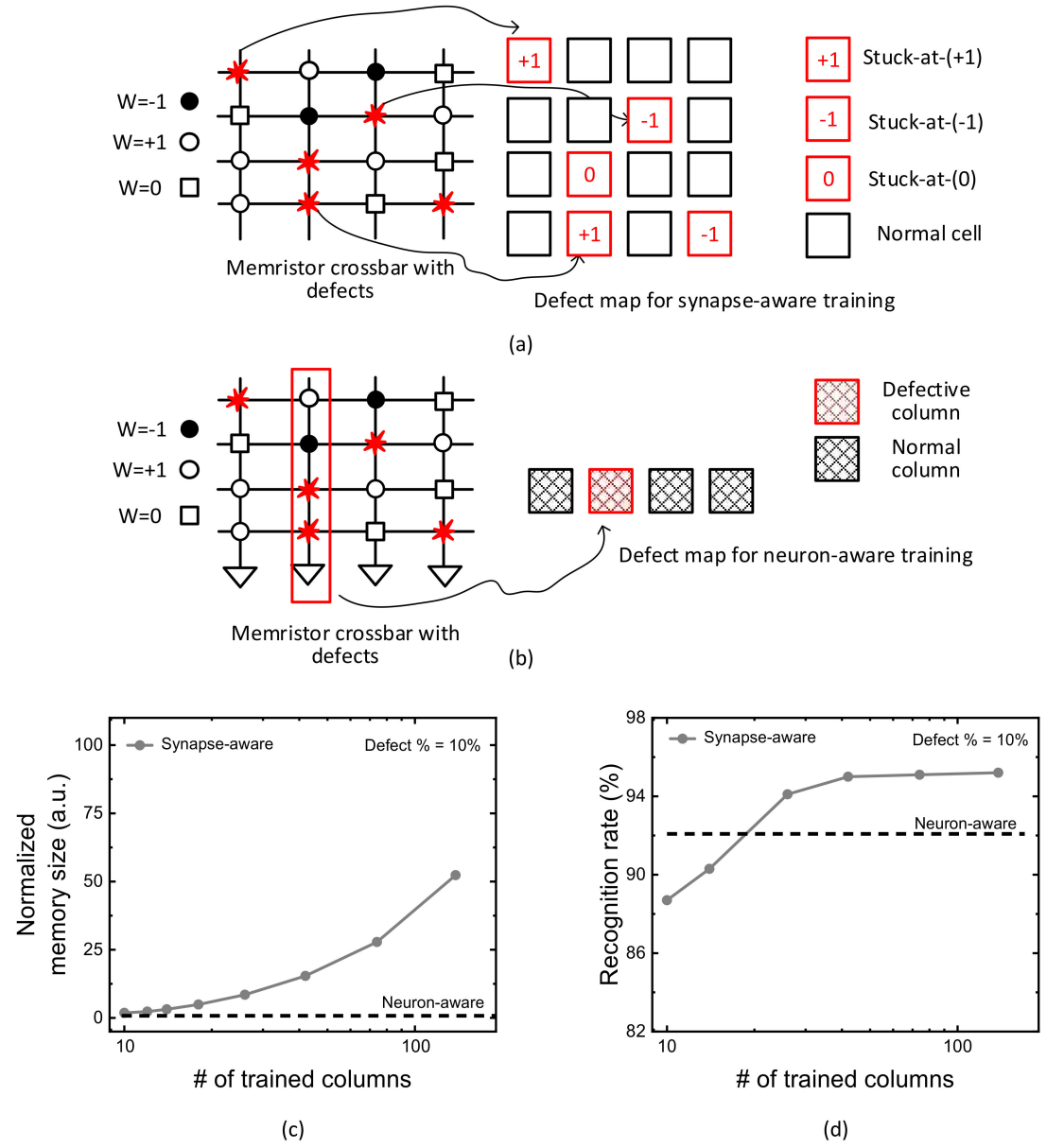

Figure 1. (**a**) The synapse-aware scheme for defective memristor crossbars (**b**) The neuron-aware scheme for defective memristor crossbars (**c**) Comparison of defect map's memory size between the synapse-aware and neuron-aware schemes with increasing the number of crossbar's columns trained for synapse defects, when the crossbar's defect percentage = 10%. The total number of columns in the crossbar is 300 + 10. (**d**) Comparison of MNSIT recognition rate between the synapse-aware and neuron-aware with increasing the number of crossbar's columns trained for synapse defects, when the crossbar's defect percentage = 10%.

Figure 1c compares the defect map's memory size between the synapse-aware and neuron-aware schemes. In Figure 1c, the y-axis is the normalized memory size required for storing the defect maps of the synapse-aware and neuron-aware schemes. The memory size of defect map in the synapse-aware scheme can be calculated by multiplying the number of faulty cells by each defect's memory size. The defect map's memory size for considering faulty cells in the synapse-aware scheme includes both row and column numbers and defect types in the crossbar. On the contrary, for the neuron-aware, the defect map's memory size can be obtained by multiplying the number of defective columns by each defective

column's memory size. Unlike the synapse-aware, the neuron-aware scheme uses only the information of defective column's locations during the training.

In Figure 1c,d, the x-axis is the number of crossbar's columns trained for considering faulty cells in the synapse-aware scheme. For the neuron-aware scheme, however, it should be noted that the number of deactivated columns during the neural network's operation is fixed not changed in Figure 1c,d. For fixing the number of deactivated columns, we obtained an optimal number of crossbar's columns that should be removed from the neural network's operation from the simulation. The optimal number of deactivated columns obtained from the simulation can maximize the neural network's performance trained with the neuron-aware scheme. In Figure 1c,d, the optimal number of crossbar's columns deactivated during the training is 50, which was obtained from the simulation of the neuron-aware scheme. By doing so, the defect map's memory size and the recognition rate of the neuron-aware scheme can be kept constant regardless of changing the number of crossbar's columns trained for considering faulty cells in the synapse-aware scheme, as shown in Figure 1c,d.

Here, an artificial neural network used for calculating the memory size in Figure 1c,d is assumed to have 784 input, 300 hidden, and 10 output neurons for recognizing MNIST dataset [12,27]. For implementing the network, 784×300 and 300×10 crossbars are used for the VMM calculation. Here the total numbers of crossbar columns are 300 and 10 for the hidden and output neurons, respectively.

Figure 1c indicates very clearly that the neuron-aware scheme in Figure 1b needs much smaller defect map's memory size than the synapse-aware in Figure 1a. As explained earlier, the neuron-aware scheme uses only the information of defective columns, while the synapse-aware needs the information of all the faulty cells during crossbar's training. Because the number of faulty cells in the synapse-aware scheme outnumbers the number of the defective columns in the neuron-aware, the neuron-aware can train the defective crossbar using much smaller defect map's memory size than the synapse-aware. Moreover, the defect map's memory size of the neuron-aware scheme can be kept constant regardless of changing the number of crossbar's columns trained for considering faulty cells in the synapse-aware scheme.

Figure 1d compares the recognition rate between the synapse-aware and neuron-aware schemes. In Figure 1d, the y-axis is MNIST recognition rate calculated by counting the number of successful and failed recognitions when the 10,000 MNIST test vectors are applied to the trained crossbar. The comparison in Figure 1d indicates that the neuron-aware scheme exceeds the synapse-aware in terms of recognition rate, when the number of crossbar's columns trained for considering synaptic defects is small. In the neuron-aware scheme, the columns which are defined as 'severely-defective' are deactivated during the training and inference to prevent them from being involved in the neural network's operation. By doing so, the neuron-aware can recognize MNIST images better than the synapse-aware, when the number of crossbar's columns trained for considering synapse defects is small. However, as the number of crossbar's columns trained with defects is increased, the synapse-aware begins to crossover the neuron-aware scheme, as indicated in Figure 1d. The crossover of recognition rate between the synapse-aware and neuron-aware is caused from the columns trained with defects can mitigate the degradation of neural network's performance due to various stuck-at-faults in the crossbar.

In this paper, a synapse-neuron-aware training method is newly proposed, where the synapse-aware scheme is combined with the neuron-aware, for exploiting trade-off relationship between the defect map size and the neural network's performance. In the synapse-neuron-aware training, the memristor crossbar's columns are divided into 3 groups. The Group-1 columns that are defined as 'severely defective' are deactivated during the training and inference, like the neuron-aware scheme. The Group-2 columns that are defined as 'moderately defective' are trained by the synapse-aware scheme, where the defect map containing the synapse defect information such as types and locations is used for training the crossbar. The Group-3 columns containing less defects than Group-1 and Group-2

are defined as 'normal columns'. For the Group-3, they are trained without considering defects. In the synapse-neuron-aware scheme proposed in this paper, each group is trained according to the trade-off relationship between the neural network's performance and the hardware overhead of defect-tolerant training. As a result of this group-based training, the synapse-neuron-aware scheme can achieve the neural network's performance better than the neuron-aware method with less hardware overhead than the synapse-aware one. By combining the synapse-aware with the neuron-aware together, in this paper, we can succeed in reducing the hardware burden of the synapse-aware method and improving the network's performance better than the neuron-aware.

2. Method

A new synapse-neuron-aware scheme for considering faulty memristor cells is proposed in this paper. Figure 2 depicts a conceptual diagram of the proposed synapse-neuron-aware scheme, where the neuron-aware scheme is combined with the synapse-aware one to optimize both the neural network performance and the hardware overhead at the same time.

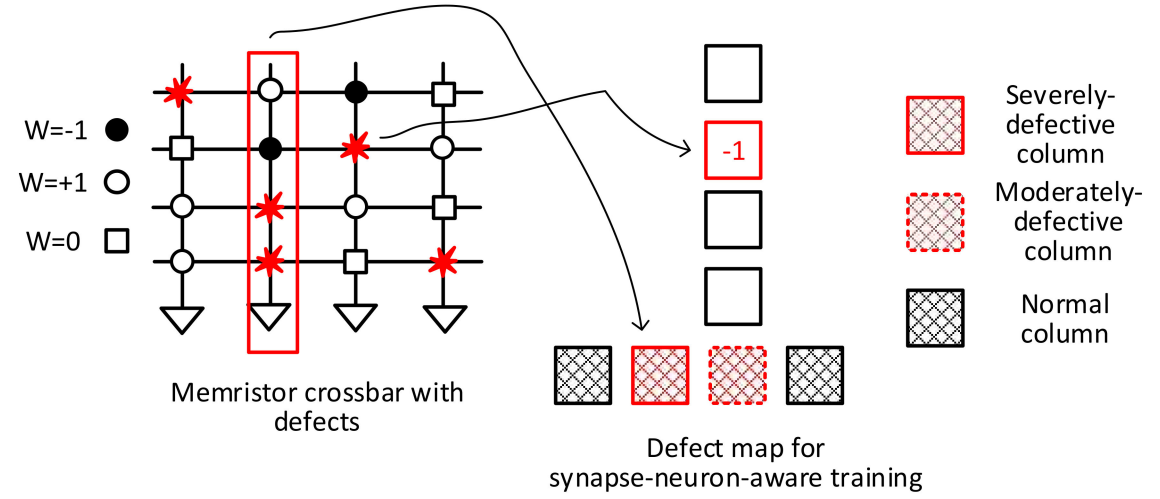

Figure 2. The synapse-neuron-aware scheme proposed for training memristor crossbars with defect cells.

To begin using the synapse-neuron-aware training method, the crossbar's columns are divided into three groups, as mentioned earlier. They are Group-1, Group-2, and Group-3, respectively. As shown in Figure 2, Group-1 represents the severely defective columns, which contain a large number of defect cells that should not be involved in the neural network's operation. To do so, the Group-1 columns are deactivated during the training. They are unable to contribute to the neuron activation in the training and inference. The Group-2 columns defined as 'moderately defective' are trained with the synapse-aware scheme. The synapse defect map containing the synaptic defect information, such as types and locations, is used to train the Group-2 columns. The Group-3 columns with fewer defects than Group-1 and Group-2 are considered normal, and they are trained without taking defects into account during the training. For the Group-3 columns, no defect map is used.

Each group in the synapse-neuron-aware scheme is trained based on the trade-off relationship between the network performance and the hardware overhead of defect-tolerant training. As a result of this group-based training, the synapse-neuron-aware training method outperforms the neuron-aware method in terms of neural network performance while requiring less hardware overhead than the synapse-aware method. By combining the synapse-aware and neuron-aware methods together, the hardware burden of the synapse-

aware method can be reduced and the network performance can be improved better than the neuron-aware method.

Figure 3 shows a flowchart of the proposed synapse-neuron-aware scheme for training memristor crossbars with defects. In Figure 3, first, a defect map is obtained from the crossbar measurement. Using the measured defect map, the crossbar columns can be divided into three groups of Group-1, Group-2, and Group-3. The Group-1 has the severely-defective columns and they should be deactivated to keep them from being involved in the neural network operation. If the Group-1 columns are involved in the neural network operation, they may be activated very frequently regardless of the input vectors. The frequent activation of severely-defective columns can degrade the neural network performance significantly. To avoid the unwanted neuron activation, the severely-defective columns are deactivated by controlling the activation function circuit as will be explained in Figure 4. After deactivating the severely-defective columns (the Group-1 columns), the memristor crossbar is trained considering the synapse defects belonging to the moderately-defective columns (the Group-2 columns), as indicated in Figure 3. After training the memristor crossbar with defects, the inference accuracy is tested. If the accuracy does not meet a target rate, the crossbar is trained again with increasing the number of Group-2 columns until the target accuracy is satisfied.

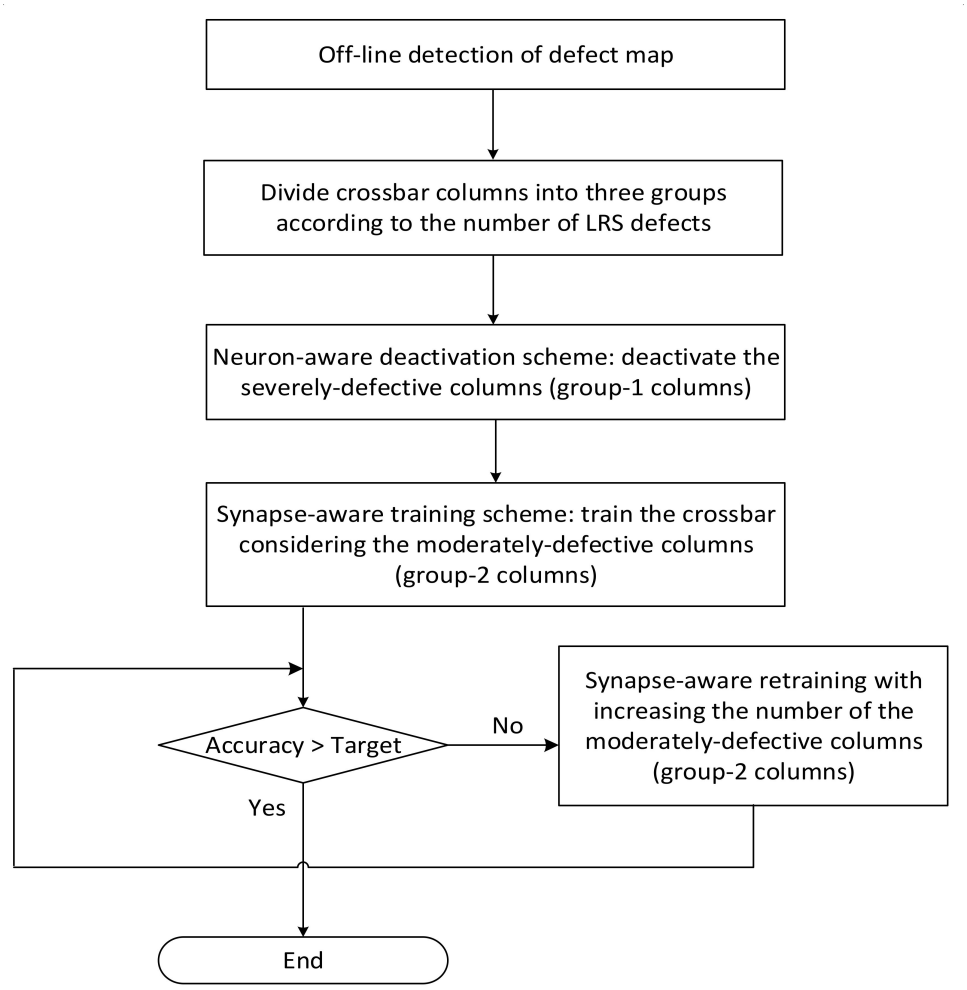

Figure 3. The flowchart of synapse-neuron-aware training scheme.

Figure 4a shows a memristor crossbar composed of transistors and memristors. Here V_1, V_2, etc. represent input voltages. To calculate both positive and negative synaptic weights, two columns are used in the crossbar, which are denoted with (+) and (−) symbols in Figure 4a. I_{1+} and I_{1-} represent positive and negative column currents, respectively. I_1 is

calculated from the difference of I_{1+} and I_{1-}. M_{11+} and M_{11-} are memristors for positive and negative columns, respectively. S_{11} is a selector for controlling the access to M_{11+} and M_{11-}. In Figure 4a, F means the activation function circuit, which is shown in detail in Figure 4b. Y_1, Y_2, etc. represent output voltages from output neurons. Here ternary synaptic weights are used in Figure 4a. For the synaptic weight = 0, both memristors on positive and negative columns are HRS. For the weight = 1, the positive and negative cells are LRS and HRS, respectively. If the positive and negative columns have HRS and LRS, the weight can be −1, respectively.

Figure 4b shows a ReLU activation function circuits with column deactivation. In the activation circuit, OP_2, acts as a current-to-voltage converter. OP_1 is an inverting amplifier. If a column has many memristor defects, this column can be defined as 'severely-defective column (Group-1)'. The column belonging to the Group-1 should be deactivated during the training time. By doing so, the column is not involved in the neural network operation. For deactivating the column, M_1 is programmed with LRS in Figure 4b. By doing so, the OP1's amplifying gain becomes very small. S_1, S_2, and S_3 are controlling switches. V_P means a memristor programming pulse, which is applied to M_1 through S_1 and S_2. S_1 and S_2 are turned on when PRG is high. On the contrary, when PRG is low, S_3 is turned on. In this case, the column current of I_{1+}-I_{1-} is converted to the node voltage of V_1, according to R_1. Figure 4c shows the column current of I_{1+}-I_1 is converted to the output voltage of Y_1, when M_1's conductance is 1/HRS and 1/LRS, respectively. The blue line means the column is deactivated. The red line is for the ReLU transfer curve.

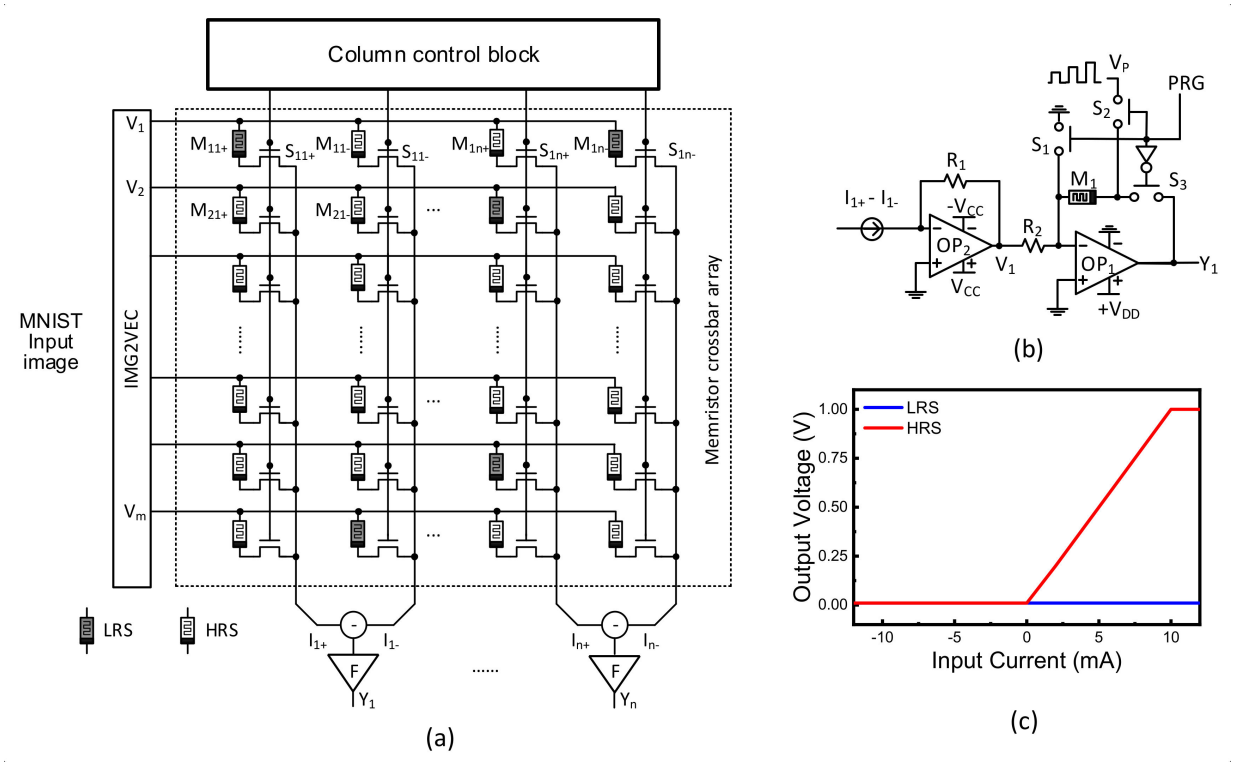

Figure 4. (**a**) The schematic of a 1T-1M crossbar composed of memristors and transistors [28–30] (**b**) The ReLU activation function circuit with column deactivation (**c**) The transfer curve for ReLU activation function with column deactivation. The red line is for the ReLU transfer curve. The blue line is for the column deactivation.

3. Results

Figure 5a shows a cross-sectional view of the memristor that was used in this paper. The measured memristor has a film structure of Pt/$LaAlO_3$/Nb-doped $SrTiO_3$ stacked layer [31,32]. Here, the top electrode layer (TE) is formed with Platinum (Pt). The bottom electrode (BE) is made of SrTiO3. The TE size is 100 μm × 100 μm. The detailed fabrication

process is explained in the previous publication [32]. Figure 5b shows a memristor's butterfly curve, where HRS and LRS are 1 MΩ and 10 kΩ, respectively [31]. Here the symbol and line represent the measurement and the Verilog-A simulation, respectively. The mathematical model equations describing the measured current-voltage relationship can be found in the previous publication [31]. The equations were implemented in Verilog-A in the CADENCE SPECTRE (Cadence Design Systems, Inc., San Jose, CA, USA), which is used for the hybrid circuit simulation of memristors and complementary metal-oxide-semiconductor (CMOS) circuits in this paper. The measurement was performed with Keithley-4200 (Semiconductor Characterization System, Tektronix, Inc., Beaverton, OR, USA) combined with the Keithley-3706 Switching Matrix.

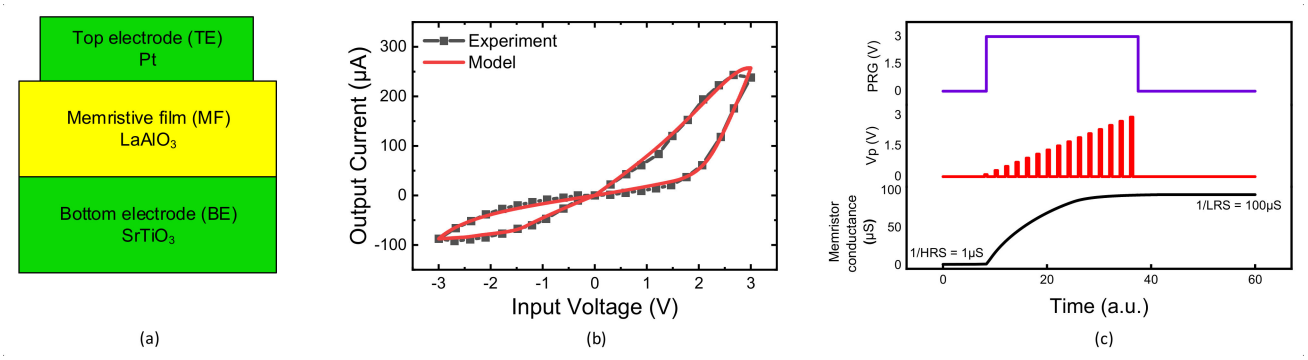

Figure 5. (**a**) The cross-sectional view of memristor used in this paper [31,32]. (**b**) The memristor's butterfly curve (**c**) The memristor's programming transient characteristic used in the ReLU activation function circuit with column deactivation.

Figure 5c shows a memristor's programming transient characteristic with applying the programming voltage pulses with amplitude modulation, as shown in Figure 4b. Initially, the memristor in Figure 4b is programmed with HRS [25]. As the voltage pulses are applied, the memristor's conductance is increased from 1/HRS to 1/LRS. The smallest conductance of memristor can deactivate the corresponding column during the training and inference, because the amplifier gain becomes very small. By doing so, the corresponding crossbar's column can be ignored in the neuron activation.

Figure 6a compares MNIST recognition rate of the neuron-aware, synapse-aware, and synapse-neuron-aware scheme, with changing the number of crossbar's columns trained for synapse defects (Group-2 columns). Here the crossbar's defect percentage is assumed 10% in Figure 6a. The assumption of defect percentage = 10% comes from the measured defect percentage of stuck-at-faults was as much as 10% in the fabricated memristor crossbars [12,33]. For the neuron-aware scheme, first, the severely-defective columns are deactivated in order to keep them from being involved in the neural network's operation. In Figure 6a, the severely-defective 50 columns are deactivated from the neuron activation among 300 and 10 columns of the hidden neurons and output neurons, respectively. Because of the deactivation, when the number of crossbar's columns trained for synapse defects is very small, the neuron-aware scheme can show the rate better than that of the synapse-aware, in Figure 6a.

On the contrary, as the number of crossbar's columns trained for synapse defects is increased, the synapse-aware scheme beings to exceed the neuron-aware. As explained earlier, the neuron-aware scheme does not consider the synapse defects during the training. Hence, the recognition rate of the neuron-aware scheme is constant regardless of increasing the number of crossbar's columns trained for synapse defects, while the synapse-aware scheme can improve the rate.

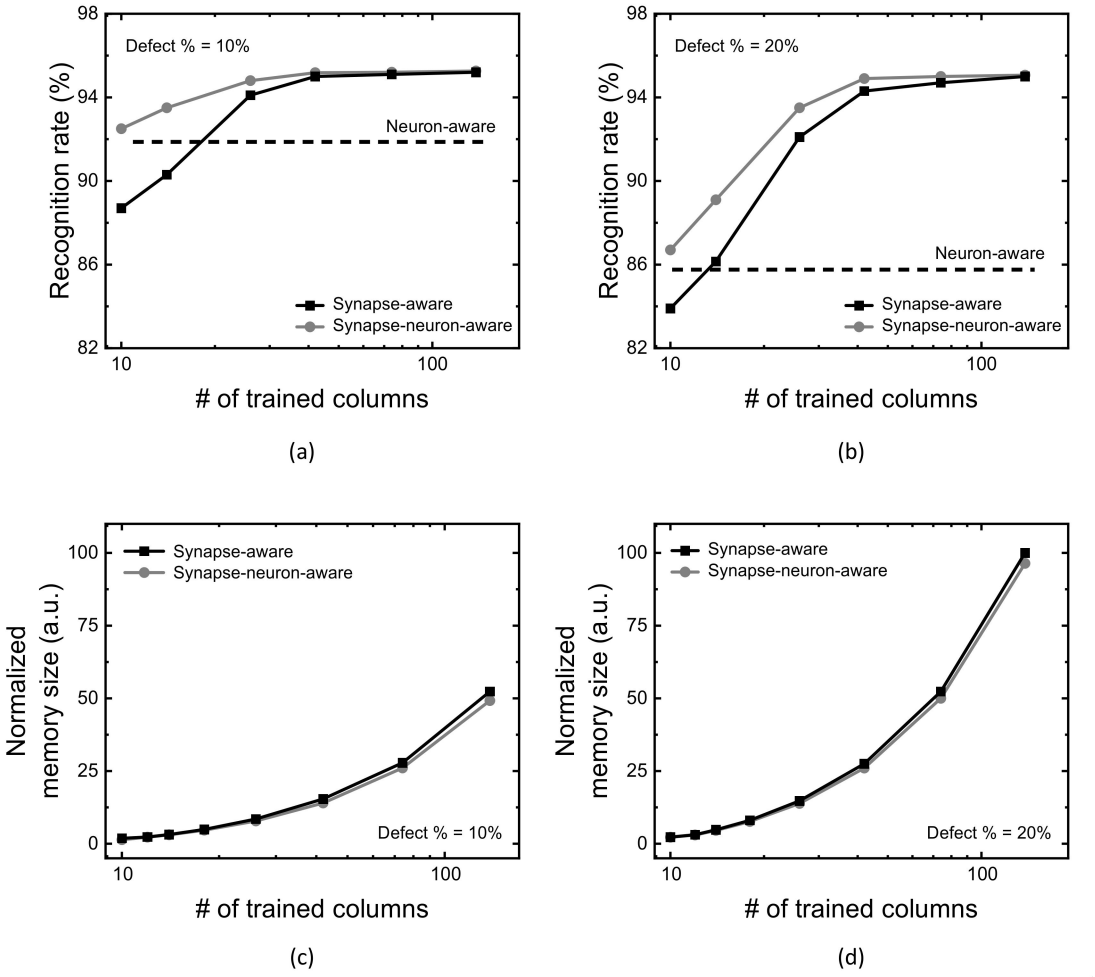

Figure 6. (**a**) MNSIT recognition rate with increasing the number of crossbar's columns trained for synapse defects (Group-2 columns), when the crossbar's defect percentage = 10%. (**b**) MNIST recognition rate with increasing the number of crossbar's columns trained for synapse defects (Group-2 columns), when the crossbar's defect percentage = 20%. (**c**) Defect map's memory size for the crossbar's defect percentage = 10% (**d**) Defect map's memory size for the crossbar's defect percentage = 20%.

The synapse-neuron-aware scheme was proposed as the combination of the synapse-aware and neuron-aware, explained in the previous Section 2. As a result of the combination of two schemes, the synapse-neuron-aware scheme acts similarly with the neuron-aware, when the number of crossbar's columns trained for synapse defects is small. As the number of crossbar's columns trained for synapse defects is increased, the combined scheme seems to follow the synapse-aware, not acting like the neuron-aware. In Figure 6a, with increasing the number of crossbar's columns trained from 10 to 138, the synapse-neuron-aware scheme shows the recognition rate better than the both the synapse-aware and neuron-aware in the entire range of trained columns.

Similarly with Figure 6a,b compares the recognition rate of the neuron-aware, synapse-aware, and synapse-neuron-aware scheme, with changing the number of crossbar's columns trained for synapse defects (Group-2 columns), for the crossbar's defect percentage = 20%.

Figure 6c,d compare defect map's memory size between the synapse-aware and synapse-neuron-aware schemes. The comparison in these figures indicates the synapse-neuron-aware requires a little smaller memory size of defect map than the synapse-aware. This is because the memory size of defect map of the neuron-aware is negligibly small. In addition, the crossbar's column deactivation of the severely-defective columns can

reduce the memory size of synapse-defect map in the synapse-neuron-aware scheme. This is due to the fact that the synapse-neuron aware scheme does not store the information of faulty cells belonging to the severely-defective columns that are removed from the neuronal network's operation. On the contrary, the synapse-aware scheme should store the information of all the faulty cells in the defect map, which requires more memory size than the synapse-neuron aware.

The proposed training method was also tested for CIFAR-10 dataset, which contains RGB colors images. Here, the conventional ResNet architecture is used for simulating the proposed synapse-neuron-aware scheme with CIFAR-10 dataset [34,35]. Like the neural network architecture for testing the MNIST dataset, 300 hidden neurons and 10 output neurons are used for implementing the fully-connected layers in ResNet architecture [36,37].

Figure 7a,b compare CIFAR-10 recognition rate of the neuron-aware, synapse-aware, and synapse-neuron-aware scheme, for the crossbar's defect percentage = 10% and 20%, respectively. In Figure 7a, when the number of crossbar's columns trained for synapse defects is 10, the gap between the synapse-neuron-aware and the synapse-aware is 0.6% for ResNet architecture. To verify the proposed scheme useful for various neural networks, VGG16 architecture is also tested in this paper [38]. From the VGG16 simulation, the synapse-neuron-aware indicates CIFAR-10 recognition rate better by 0.8% than the synapse-aware, which is very similar with the ResNet simulation result, when the number of crossbar's columns trained for synapse defects is 10. Figure 7c,d compare the memory size between the synapse-neuron-aware and synapse-aware, for the crossbar's defect percentage = 10% and 20%, respectively.

Considering both the recognition rate and memory size in Figure 7a,c, respectively, for the defect percentage = 10%, the trade-off relationship between the neural network's performance and hardware overhead is indicated clearly. As the number of trained columns for considering synapse defects is increased, the recognition rate becomes improved. In contrast, the memory overhead due to the defect map is degraded. Reversely, if the number of columns trained in the crossbar is decreased, the recognition rate becomes degraded, while the defect map's size becomes small. Though the synapse-aware and the proposed synapse-neuron-aware show the same trade-off relationship between the neural network's performance and hardware overhead, there is one thing different between the proposed and the synapse-aware schemes, as indicated in Figure 7a. In the entire range of the number of trained columns, the proposed scheme can show better recognition rate than the synapse-aware, as indicated in Figure 7a,b, because the defective column deactivation compensates for the performance degradation caused from severely-faulty columns.

Finally, one thing to note here is that the synapse-neuron-aware scheme proposed in this paper can be more suitable to the on-line training applications compared to the off-line software training. This is owing to the fact that the proposed scheme can demonstrate the neural network's performance better than the synapse-aware, in spite of using a smaller memory size of defect map. Especially, for the edge intelligence applications, where the embedded memory size can be very limited, the smaller defect map's memory size of the proposed scheme can alleviate a hardware burden of the on-device training significantly.

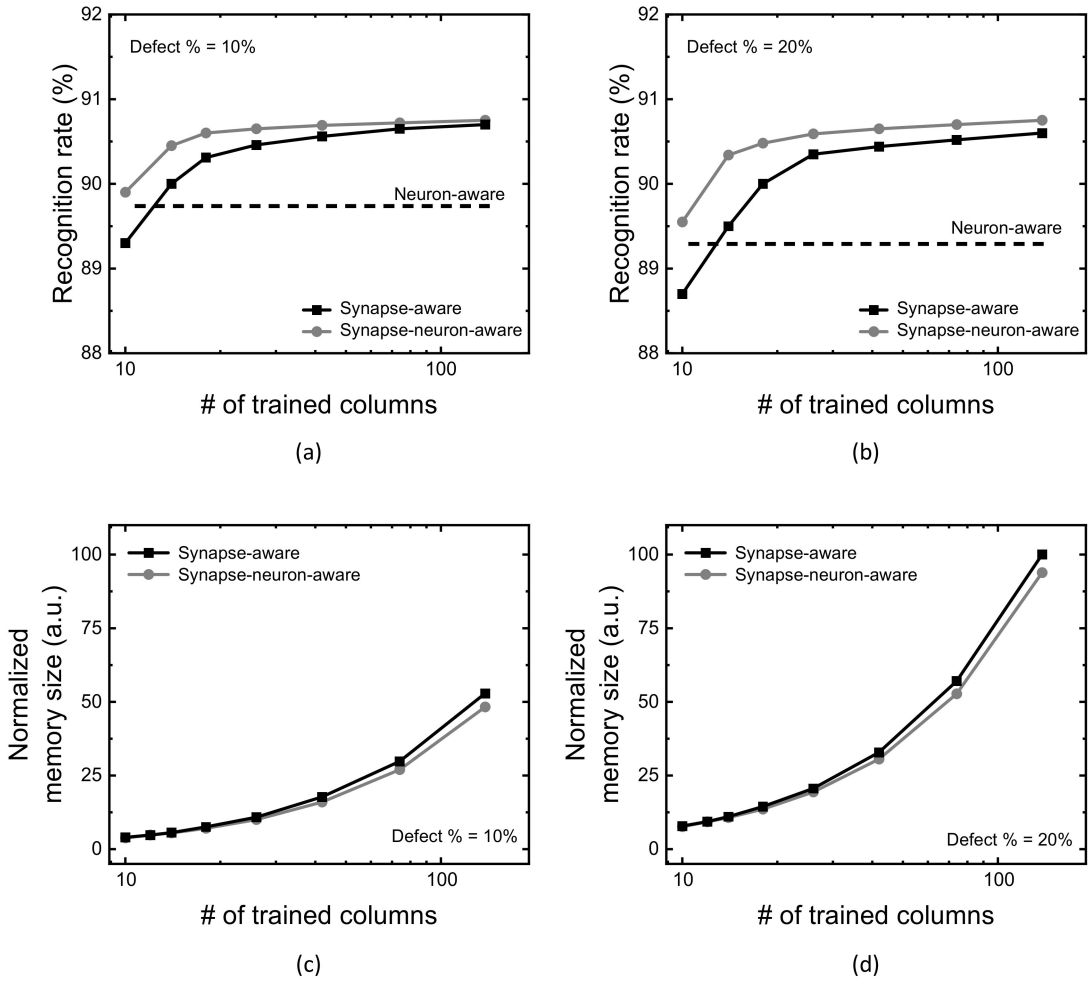

Figure 7. (**a**) CIFAR-10 recognition rate with increasing the number of crossbar's columns trained for synapse defects (Group-2 columns), when the crossbar's defect percentage = 10%. (**b**) CIFAR-10 recognition rate with increasing the number of crossbar's columns trained for synapse defects (Group-2 columns), when the crossbar's defect percentage = 20%. (**c**) Defect map's memory size for the crossbar's defect percentage = 10% (**d**) Defect map's memory size for the crossbar's defect percentage = 20%.

4. Conclusions

To overcome the limitations of CMOS digital systems, emerging computing circuits such as memristor crossbars have been investigated as potential candidates for significantly increasing the speed and energy efficiency of next-generation computing systems, which are required for implementing future AI hardware. Unfortunately, manufacturing yield still remains a serious challenge in adopting memristor-based computing systems due to the limitations of immature fabrication technology.

To compensate for the malfunction of neural networks caused from the fabrication-related defects, the new crossbar training scheme combining the synapse-aware with the neuron-aware together was proposed for exploiting trade-off relationship between the defect map size and the neural network's performance, in this paper. In the proposed scheme, the memristor crossbar's columns are divided into the 3 groups, which are the severely-defective, moderately-defective, and normal columns, respectively, according to the number of LRS defects per columns. Here, each group can be trained according to the trade-off relationship between the neural network's performance and the hardware overhead of defect-tolerant training.

As a result of this group-based training, the synapse-neuron-aware scheme could achieve the neural network's performance better than the neuron-aware method while using less hardware overhead than the synapse-aware one. For example, when testing the defect percentage = 10% with MNIST dataset, the proposed scheme outperformed the synapse-aware and the neuron-aware by 3.8% and 3.4% for the number of crossbar's columns trained for synapse defects = 10 and 138 among 310, respectively, while maintaining the smaller memory size than the synapse-aware. When the trained columns = 138, the normalized memory size of the synapse-neuron-aware scheme could be smaller by 3.1% than the synapse-aware.

Author Contributions: All authors have contributed to the submitted manuscript. K.-S.M. defined the research topic. J.A. and S.O. contributed equally as first authors to the work. J.A., S.O. and T.V.N. performed the simulation and measurement. J.A., S.O., T.V.N. and K.-S.M. wrote the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: The work was financially supported by NRF-2015R1A5A7037615, NRF-2019K1A3A1A 25000279, NRF-2021R1A2C1011631, and NRF-2021M3F3A2A01037972, and SRFC-TA1903-01.

Informed Consent Statement: Not applicable.

Acknowledgments: The CAD tools were supported by IC Design Education Center (IDEC), Daejeon, Korea.

Conflicts of Interest: The authors declare that they have no competing interests.

References

1. Guger, C.; Ramoser, H.; Pfurtscheller, G. Real-time EEG analysis with subject-specific spatial patterns for a brain-computer interface (BCI). *IEEE Trans. Rehabil. Eng.* **2000**, *8*, 447–456. [CrossRef]
2. Carpenter, G.A.; Grossberg, S.; Reynolds, J.H. Artmap: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Netw.* **1991**, *4*, 565–588. [CrossRef]
3. Merolla, P.A.; Arthur, J.V.; Alvarez-Icaza, R.; Cassidy, A.S.; Sawada, J.; Akopyan, F.; Jackson, B.L.; Imam, N.; Guo, C.; Nakamura, Y.; et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **2014**, *345*, 668–673. [CrossRef] [PubMed]
4. Li, B.; Shan, Y.; Hu, M.; Wang, Y.; Chen, Y.; Yang, H. Memristor-based approximated computation. In Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED), Beijing, China, 4–6 September 2013; pp. 242–247.
5. Li, B.; Gu, P.; Shan, Y.; Wang, Y.; Chen, Y.; Yang, H. RRAM-Based Analog Approximate Computing. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2015**, *34*, 1905–1917. [CrossRef]
6. Vanhoucke, V.; Senior, A.; Mao, M.Z. Improving the speed of neural networks on CPUs. In Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Granada, Spain, 12–17 December 2011.
7. Li, C.; Hu, M.; Li, Y.; Jiang, H.; Ge, N.; Montgomery, E.; Zhang, J.; Song, W.; Dávila, N.; Graves, C.; et al. Analogue signal and image processing with large memristor crossbars. *Nat. Electron.* **2018**, *1*, 52–59. [CrossRef]
8. Wu, J.; Choi, M. Memristor lookup table (MLUT)-based asynchronous nanowire crossbar architecture. In Proceedings of the 2010 10th IEEE Conference Nanotechnology, NANO 2010, Ilsan, Gyeonggi-Do, Korea, 17–20 August 2010; pp. 1100–1103. [CrossRef]
9. Haron, N.Z.; Hamdioui, S. On Defect Oriented Testing for Hybrid CMOS/Memristor Memory. In Proceedings of the 2011 Asian Test Symposium, New Delhi, India, 20–23 November 2011; pp. 353–358.
10. Kannan, S.; Karri, R.; Sinanoglu, O. Sneak path testing and fault modeling for multilevel memristor-based memories. In Proceedings of the 2013 IEEE 31st International Conference on Computer Design (ICCD) 2013, Asheville, NC, USA, 6–9 October 2013; pp. 215–220.
11. Kannan, S.; Rajendran, J.; Karri, R.; Sinanoglu, O. Sneak-Path Testing of Crossbar-Based Nonvolatile Random Access Memories. *IEEE Trans. Nanotechnol.* **2013**, *12*, 413–426. [CrossRef]
12. Yeo, I.; Chu, M.; Gi, S.-G.; Hwang, H.; Lee, B.-G. Stuck-at-Fault Tolerant Schemes for Memristor Crossbar Array-Based Neural Networks. *IEEE Trans. Electron Devices* **2019**, *66*, 2937–2945. [CrossRef]
13. Hu, M.; Li, H.; Wu, Q.; Rose, G.S. Hardware realization of BSB recall function using memristor crossbar arrays. In Proceedings of the DAC Design Automation Conference 2012, San Francisco, CA, USA, 3–7 June 2012; pp. 498–503.
14. Tarkov, M.S. Mapping neural network computations onto memristor crossbar. In Proceedings of the 2015 International Siberian Conference on Control and Communications (SIBCON), Omsk, Russia, 21–23 May 2015; pp. 7–10.
15. Xia, L.; Gu, P.; Li, B.; Tang, T.; Yin, X.; Huangfu, W.; Yu, S.; Cao, Y.; Wang, Y.; Yang, H. Technological Exploration of RRAM Crossbar Array for Matrix-Vector Multiplication. *J. Comput. Sci. Technol.* **2016**, *31*, 3–19. [CrossRef]
16. Hu, M.; Li, H.; Chen, Y.; Wu, Q.; Rose, G.; Linderman, R.W. Memristor Crossbar-Based Neuromorphic Computing System: A Case Study. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 1864–1878. [CrossRef]

17. Kataeva, I.; Merrikh-Bayat, F.; Zamanidoost, E.; Strukov, D. Efficient training algorithms for neural networks based on memristive crossbar circuits. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–17 July 2015. [CrossRef]
18. Xia, L.; Huangfu, W.; Tang, T.; Yin, X.; Chakrabarty, K.; Xie, Y.; Wang, Y.; Yang, H. Stuck-at Fault Tolerance in RRAM Computing Systems. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2017**, *8*, 102–115. [CrossRef]
19. Liu, C.; Hu, M.; Strachan, J.P.; Li, H. Rescuing Memristor-based Neuromorphic Design with High Defects. In Proceedings of the 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, USA, 18–22 June 2017. [CrossRef]
20. Choi, W.; Gi, S.-G.; Lee, D.; Lim, S.; Lee, C.; Lee, B.-G.; Hwang, H. WOx-Based Synapse Device With Excellent Conductance Uniformity for Hardware Neural Networks. *IEEE Trans. Nanotechnol.* **2020**, *19*, 594–600. [CrossRef]
21. Jin, S.; Pei, S.; Wang, Y. On Improving Fault Tolerance of Memristor Crossbar Based Neural Network Designs by Target Sparsifying. In Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 9–13 March 2020. [CrossRef]
22. Nguyen, T.V.; Pham, K.V.; Min, K.S. Hybrid Circuit of Memristor and Complementary Metal-Oxide-Semiconductor for Defect-Tolerant Spatial Pooling with Boost-Factor Adjustment. *Materials* **2019**, *12*, 2122. [CrossRef] [PubMed]
23. Gaol, D.; Zhang, G.L.; Yin, X.; Li, B.; Schlichtmann, U.; Zhuo, C. Reliable Memristor-based Neuromorphic Design Using Variation- and Defect-Aware Training. In Proceedings of the 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), Munich, Germany, 1–4 November 2021; pp. 1–9.
24. Kim, Y.; Kim, S.; Yeh, C.C.; Narayanan, V.; Choi, J. Hardware and Software Co-optimization for the Initialization Failure of the ReRAM-based Cross-bar Array. *ACM J. Emerg. Technol. Comput. Syst.* **2020**, *16*, 1–19. [CrossRef]
25. Van Pham, K.; Van Nguyen, T.; Min, K.-S. Partial-Gated Memristor Crossbar for Fast and Power-Efficient Defect-Tolerant Training. *Micromachines* **2019**, *10*, 245. [CrossRef] [PubMed]
26. Van Nguyen, T.; Mo, H.; Kim, D.; Min, K. Training Procedure of Memristor-Crossbar Neural Networks with Ternary Quantization. In Proceedings of the International Conference on Electronics, Information, and Communication (ICEIC), Barcelona, Spain, 19–22 January 2020; pp. 4–6.
27. LeCun, Y.; Cortes, C.; Burges, C. *The MNIST Dataset of Handwritten Digits (Images)*; NYU: New York, NY, USA, 1999.
28. Merced-grafals, E.J.; Dávila, N.; Ge, N.; Williams, R.S.; Strachan, J.P. Repeatable, accurate, and high speed multi-level programming of memristor 1T1R arrays for power ef fi cient analog computing applications. *Nanotechnology* **2016**, *27*, 365202. [CrossRef] [PubMed]
29. Wang, Z.-R.; Su, Y.-T.; Li, Y.; Zhou, Y.-X.; Chu, T.-J.; Chang, K.-C.; Chang, T.-C.; Tsai, T.-M.; Sze, S.M.; Miao, X.-S. Functionally Complete Boolean Logic in 1T1R Resistive Random Access Memory. *IEEE Electron Device Lett.* **2016**, *38*, 179–182. [CrossRef]
30. Chen, P.-Y.; Yu, S. Compact Modeling of RRAM Devices and Its Applications in 1T1R and 1S1R Array Design. *IEEE Trans. Electron. Devices* **2015**, *62*, 4022–4028. [CrossRef]
31. Truong, S.N.; Van Pham, K.; Yang, W.; Shin, S.; Pedrotti, K.; Min, K.-S. New pulse amplitude modulation for fine tuning of memristor synapses. *Microelectron. J.* **2016**, *55*, 162–168. [CrossRef]
32. Jang, J.T.; Ko, D.; Ahn, G.; Yu, H.R.; Jung, H.; Kim, Y.S.; Kim, D.H. Effect of oxygen content of the LaAlO3 layer on the synaptic behavior of Pt/LaAlO3/Nb-doped SrTiO3 memristors for neuromorphic applications. *Solid State. Electron.* **2018**, *140*, 139–143. [CrossRef]
33. Chen, C.-Y.; Shih, H.-C.; Wu, C.-W.; Lin, C.-H.; Chiu, P.-F.; Sheu, S.-S.; Chen, F.T. RRAM Defect Modeling and Failure Analysis Based on March Test and a Novel Squeeze-Search Scheme. *IEEE Trans. Comput.* **2014**, *64*, 180–190. [CrossRef]
34. Krizhevsky, A.; Nair, V.; Hinton, G. CIFAR-10 and CIFAR-100 Datasets. 2018. Available online: https//www.cs.toronto.edu/~{}kriz/cifar.html (accessed on 20 October 2018).
35. Van Pham, K.; Tran, S.B.; Van Nguyen, T.; Min, K.-S. Asymmetrical Training Scheme of Binary-Memristor-Crossbar-Based Neural Networks for Energy-Efficient Edge-Computing Nanoscale Systems. *Micromachines* **2019**, *10*, 141. [CrossRef] [PubMed]
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
37. Van Pham, K.; Min, K.-S. Non-Ideal Effects of Memristor-CMOS Hybrid Circuits for Realizing Multiple-Layer Neural Networks. In Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Hokkaido, Japan, 26–29 May 2019; pp. 1–5.
38. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.