

# Deep Neural Network Optimized to Resistive Memory with Nonlinear Current-Voltage Characteristics

Hyungjun Kim\*, Taesu Kim\*, Jinseok Kim, and Jae-Joon Kim†

Department of Creative IT Engineering, POSTECH, Pohang, South Korea

## Abstract

Artificial Neural Network computation relies on intensive vector-matrix multiplications. Recently, the emerging nonvolatile memory (NVM) crossbar array showed a feasibility of implementing such operations with high energy efficiency, thus there are many works on efficiently utilizing emerging NVM crossbar array as analog vector-matrix multiplier. However, its nonlinear I-V characteristics restrain critical design parameters, such as the read voltage and weight range, resulting in substantial accuracy loss. In this paper, instead of optimizing hardware parameters to a given neural network, we propose a methodology of reconstructing a neural network itself optimized to resistive memory crossbar arrays. To verify the validity of the proposed method, we simulated various neural network with MNIST and CIFAR-10 dataset using two different specific Resistive Random Access Memory (RRAM) model. Simulation results show that our proposed neural network produces significantly higher inference accuracies than conventional neural network when the synapse devices have nonlinear I-V characteristics.

## 1 Introduction

In recent years, Artificial Neural Network (ANN) has been gaining significant interest by claiming several cutting-edge results in solving various nonlinear problems [1]. The breakthrough of ANN heavily depends on the expansion of networks in depth, which requires vast amount of vector-matrix multiplications. With the advent of vector-matrix multiplication acceleration based on graphics processing units (GPUs), large and deep neural networks have been able to handle complex tasks using extensive amounts of data [2]. However, despite the fact that GPUs provide highly parallel computing suitable for ANNs, the high power consumption of GPUs is an obstacle to be improved. To address the issue, many dedicated accelerators for vector-matrix multiplications have been proposed [3–6].

Emerging nonvolatile memory (NVM) technologies including Phase-Change Random Access Memory (PCRAM), Resistive Random Access Memory (RRAM), Conductive-Bridge Random Access Memory (CBRAM), and Spin-Transfer-Torque Magnetic Random Access Memory (STT-MRAM) have been widely studied as next generation memories [7]. While conventional memories such as Static Random Access Memory (SRAM), and FLASH memory are charge-based, emerging NVM is current-based and represents states with different resistance values. This current-based nature opens up the opportunity to use emerging NVM for neural network acceleration. Current-based devices in a crossbar array structure can straightforwardly implement vector-matrix multiplication in neural network computations as shown in Fig. 1. By

---

\*Hyungjun Kim and Taesu Kim equally contributed to this work.

†To whom correspondence should be addressed; E-mail: jaejoon@postech.ac.kr

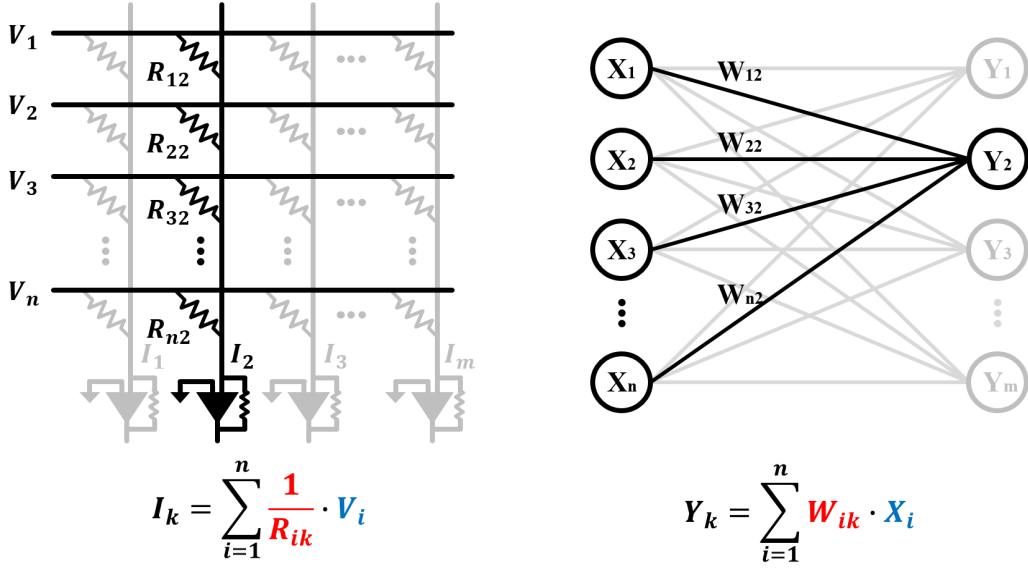


Figure 1: An example of mapping vector-matrix multiplication to a RRAM crossbar array

mapping input vector to input voltages and weight matrix to resistive crossbar array, vector-matrix multiplication can be calculated in a single step by sampling the current flowing in each column [8]. Since this approach can be several orders of magnitude more efficient than CMOS ASIC approaches in terms of both speed and power [3–6], many studies proposed neural network accelerators based on emerging NVM crossbar array [9–12].

However, there are several issues with using emerging NVM crossbar array as an analog multiplier. Sneak path problem is one of the most critical issues [7, 11, 13, 14]. Various works have tried to solve this problem in different ways [7, 14, 15]. The most common idea is to use a device with nonlinear I-V characteristics. For example, it has been proposed to serially connect a selector device such as a transistor or a diode to an emerging NVM cell or to make the I-V characteristic of an emerging NVM cell as nonlinear as possible [7]. However, although this approach can overcome the sneak path problem, it degrades the accuracy of current-based vector-matrix multiplication because nonlinear I-V characteristics hinder precise implementation of linear multiplications required for vector-matrix multiplications. Several works tried to solve this issue by restricting the range of the reading voltage to use the pseudo-linear sub-region of the nonlinear I-V curve [16, 17]. However, limiting reading voltage worsened DAC resolution issue, making it difficult to compute complex neural networks using emerging NVM crossbar arrays. Another previous approach was to address the problem by tuning the weights considering the computational error before mapping [4, 18]. However, it also failed to utilize full input voltage range. In addition, these approaches did not fully address how the increase of the nonlinearity of I-V characteristics affects the inference accuracy of neural networks.

Unlike previous approaches which attempted to precisely map pre-determined weights to the crossbar array to reduce accuracy loss, we propose to rather construct an ANN model itself which accommodates nonlinear I-V characteristics. This allows nonlinear I-V characteristics to be taken into account during both of the learning phase and the inference phase of a neural network, reducing discrepancies between neural network models and emerging NVM-based hardware. In this paper, we have selected two nonlinear RRAM devices as proof-of-concept devices and performed simulations to verify the idea based on the characteristics of the devices. The main contribution of this paper is as follows:

- 1) We analyze the correlation between the degree of nonlinearity of I-V characteristics and the inference accuracy loss in RRAM-based ANN hardware. We show that the degree of accuracy loss depends on the distribution of activation values.

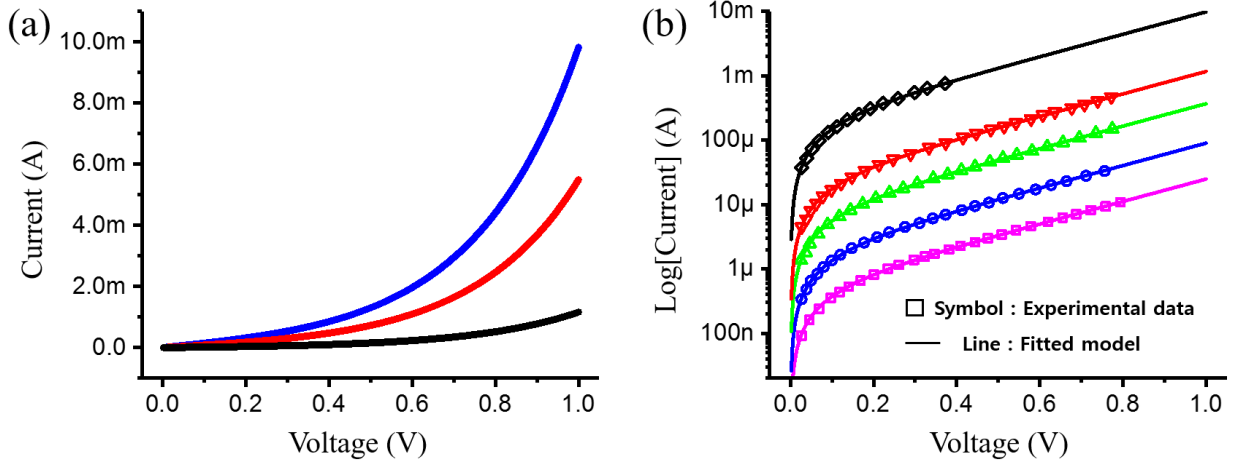


Figure 2: (a) I-V curve of a real RRAM device in different resistance states. (b) Experimental data and fitted curve of the device under various resistance states [19].

2) We propose a modified perceptron model that is compatible with the nonlinear I-V characteristics of resistive memory devices and demonstrate how to train neural networks based on the proposed model. We show that neural networks based on the proposed model can avoid the loss of inference accuracy which happens while mapping the network to RRAM crossbar arrays.

## 2 Preliminaries

### 2.1 Nonlinear I-V Characteristics of RRAM Devices

Fig. 2 illustrates the I-V characteristics of an actual metal-oxide RRAM device extracted from [19]. Each line shows the I-V curve of the RRAM device given a specific sequence of set-voltage pulses. As shown in Fig. 2(a), the I-V relationship of the RRAM device has an exponential form. [19] suggests that the I-V characteristics can be modeled as a empirical model with a sinh function

$$I(V) = e^{d/d_0} \sinh(BV) \quad (1)$$

where  $d$  is average tunneling gap distance,  $d_0$  is a fitting parameter, and  $B$  is a constant. Measurement results and the values from the empirical model match well as shown in Fig. 2(b). Each I-V curve corresponding to a different state can be obtained by appropriately determining a state variable  $d$ .

The degree of nonlinearity of an I-V curve can be represented by half-bias nonlinearity  $k$ , which is defined as

$$k = \frac{I(V_{r,\max})}{I(V_{r,\max}/2)} \quad (2)$$

where  $V_{r,\max}$  is the maximum value of read voltage that can be used without disturbing the state of the device. A  $k$  value does not guarantee unique I-V characteristics of a particular device, as two different devices with the same  $k$  value can have different I-V curves. However, it can be still said that  $k$  value represents the nonlinearity in some degree. In this paper, we assume  $V_{r,\max} = 1$  V for simplicity without loss of generality. Then,  $k$  can be expressed as a function of  $B$  as follows,

$$k = \frac{e^B - e^{-B}}{e^{B/2} - e^{-B/2}} \quad (3)$$

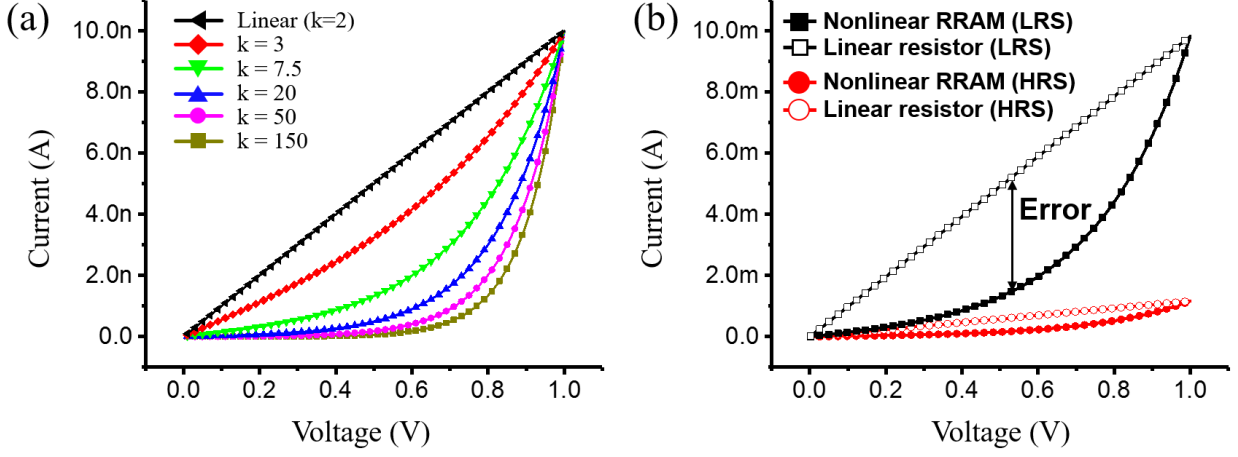


Figure 3: (a) I-V curves in sinh form with various degrees of nonlinearity. In this case parameter  $B$  determines nonlinearity  $k$ . The model device has  $k = 7.5$ , leading to  $B = 4$  in Eq. (1). (b) Error between nonlinear I-V curves of RRAM devices and corresponding linear I-V curves of resistors. Hollow symbols stand for the linear resistor model and filled symbols stand for the real device model.

When  $k = 2$ , the device has a linear I-V characteristic. As  $k$  increases, the I-V characteristic of the corresponding device becomes more nonlinear and exhibits a larger current difference from the linear I-V characteristic with the same resistance state (Fig. 3(a)). To investigate the  $k$  values of existing RRAM devices, we surveyed several papers and manually extracted I-V characteristic data of the proposed devices. We could observe that  $k$  has a wide distribution ranging from 2.5 [20] to 70 [21,22]. Among the devices, we chose a device with  $k = 7.5$  as the model device for the rest of the paper.

## 2.2 Weight Mapping

ANN utilizes a vector-matrix multiplication of the corresponding input vector and weight matrix to obtain a weighted sum for a layer as

$$\mathbf{s} = \mathbf{x} \cdot \mathbf{W} \quad (4)$$

with  $\mathbf{s}$  as the weighted sum vector,  $\mathbf{x}$  as the input vector, and  $\mathbf{W}$  as the weight matrix. To implement Eq. (4) using an emerging NVM crossbar array, each weight in particular row and column of the weight matrix must be mapped to a characteristic parameter of a corresponding device in the crossbar array. In previous approaches [4, 16, 23] which used RRAM crossbar arrays, weights were mapped to the conductance of the devices assuming linear I-V characteristics as follows,

$$G(w) = \frac{(G_{\max} - G_{\min})(w - w_{\max})}{w_{\max} - w_{\min}} + \frac{G_{\max}(w_{\max} - w_{\min})}{G_{\max} - G_{\min}} \quad (5)$$

Eq. (5) takes a **naive linear-mapping approach** that maps the minimum weight to the minimum conductance and maps the maximum weight to the maximum conductance. In this approach, input vectors can be represented by a set of voltages applied to rows of the crossbar array and the result of the vector-matrix multiplication can be obtained by sampling the current in each column. Previous works mostly relied on the naive mapping and attempted to mimic linear I-V characteristics by limiting the reading voltage into a small linear range [16, 24].

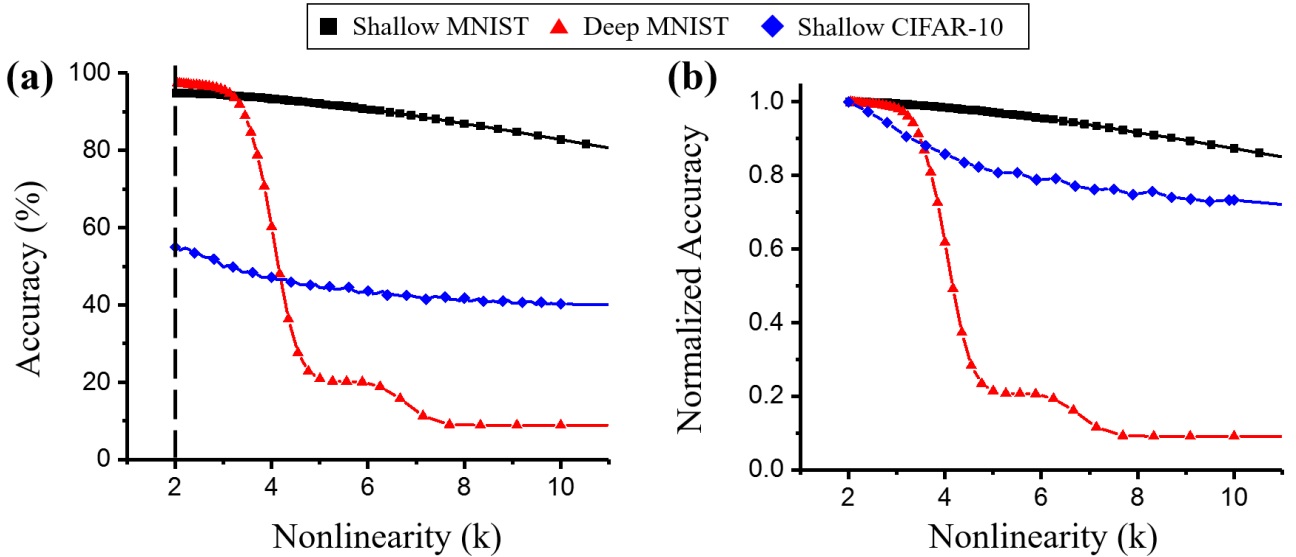


Figure 4: (a) Accuracy vs. nonlinearity ( $k$ ) and (b) normalized accuracy loss vs. nonlinearity ( $k$ ) for shallow MNIST case (black square), deep MNIST case (red uptriangle), and shallow CIFAR-10 case (blue diamond). (b) describes the accuracy loss relative to the baseline (linear case) accuracy.

### 3 Error analysis

In section 3, we analyze the inference error of the neural networks naively mapped to a RRAM crossbar array consisting of the devices with nonlinear I-V characteristics. To analyze the errors that occur while using full range of  $V_{\text{read}}$ , we simulated the consequences of applying the naive mapping method to the devices with the I-V characteristics given in [19] (Fig. 2). Based on the range of device current at 1 V, weights were linearly converted to conductance using Eq. (5). The empirical I-V model given in Eq. (1) was used for inference simulations. Fig. 3(b) shows the mapping strategy.

To acquire model weights for analysis, two different Multilayer Perceptron (MLP) networks were trained with MNIST dataset. A shallow network (784-500-250-10, **shallow MNIST case**) with two hidden layers and a deep network (784-2500-2000-1500-1000-500-10, **deep MNIST case**) with five hidden layers were each trained with Stochastic Gradient Descent (SGD) using MATLAB. Also, a shallow MLP (2352-4000-1000-4000-10, **shallow CIFAR-10 case**) was trained with the same method to classify CIFAR-10 dataset for additional analysis.

For each neural network model, the inference accuracy was evaluated using the naive mapping discussed above. Several simulations were performed by sweeping the  $k$  values to investigate the relationship between the degree of the inference accuracy degradation and the nonlinearity of the I-V characteristics. For each  $k$ , corresponding parameter  $B$  in the numerical I-V model Eq. (1) could be retrieved using Eq. (3). Evaluation results are illustrated in Fig. 4. We could observe that overall inference accuracies decrease as the I-V characteristics of the devices become more nonlinear. In addition, the inference accuracy of the deep MNIST case began to drop at lower  $k$  compared to the shallow MNIST case. Another observation was that the inference accuracy of the shallow CIFAR-10 case also started to fall at low  $k$ .

To analyze the cause of accuracy degradation, we investigated the relationship between input value distribution and computation error. Because the current difference between linear and nonlinear I-V curves is the largest at an input voltage of about 0.5 V (Fig. 3(b)), we speculated that input values around 0.5 V are prone to errors.

To verify this claim, two distinct distributions were given as input vectors to the second layer of the deep MNIST case. Input data A (Fig. 5(a)) was generated by truncating a normal distribution of numbers so that all data were around 0 and 1. Input data B (Fig. 5(d)) was

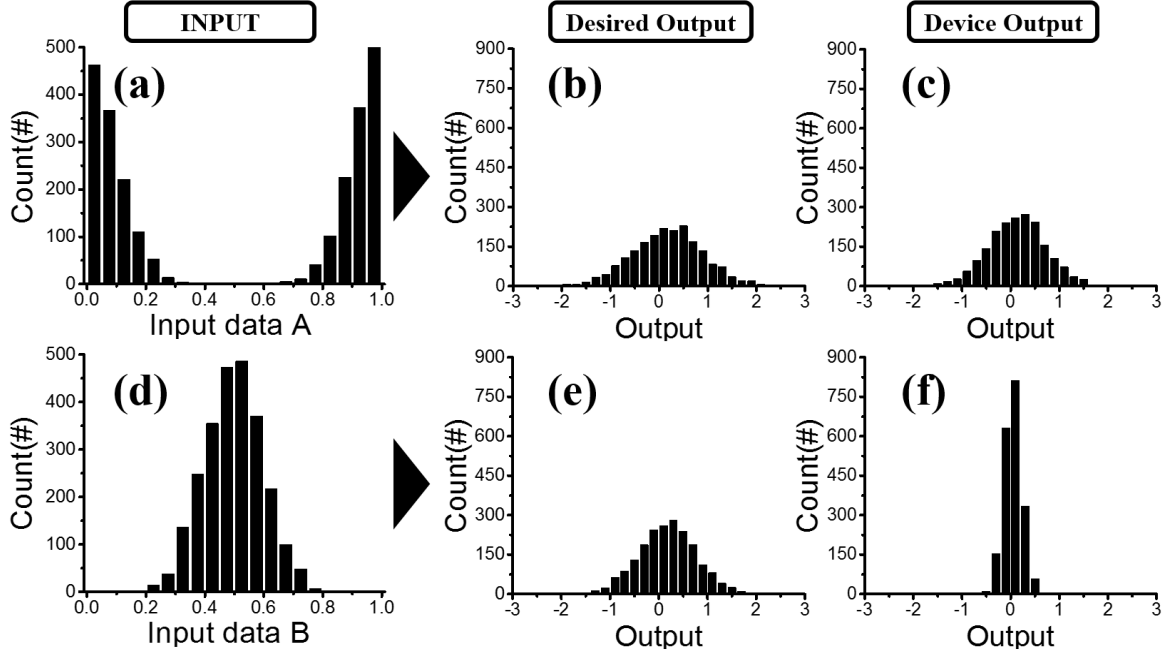


Figure 5: Distribution of input data A (a) and B (d). Desired output and actual device output for input data A (b,c) and B (e,f).

generated as a normal distribution of numbers with a mean of 0.5 to let all data be around 0.5. The result showed that when input data A was fed to the layer, an output distribution similar to that of the ideal vector-matrix multiplication appeared (Fig. 5(b), (c)). While, feeding input data B to the layer resulted in an output distribution with large error (Fig. 5(e), (f)). Based on this observation, we investigated two representative cases that can induce mid-range input and cause inference accuracy degradation.

### 3.1 Impact of Network Depth

In the shallow MNIST case, the activation values of neurons tend to yield extreme values such as 0 and  $V_{\text{read}}$  (Fig. 6(a), (c), (d)), since the network can easily fall into saturation due to the lack of training parameters. In contrast, a deeper network with an increased number of training parameters can reduce the probability of saturation and induce mid-range activation values. Increased depth of a layer can make the accuracy even worse because computation error due to mid-range activation values can accumulate on several layers. Fig. 6(e), (f) demonstrates that the activation values of the third and fourth hidden layer of the deep MNIST case have relatively more mid-range values compared to shallow networks. This explains why the shallow MNIST case could maintain relatively high accuracy while the deep MNIST case was vulnerable to the I-V nonlinearity, resulting in greater accuracy loss.

### 3.2 Impact of Input Data Distribution

Fig. 6(a), (b) present the distribution of randomly selected training data from the MNIST dataset and the CIFAR-10 dataset. As the CIFAR-10 dataset consists of natural images with various RGB data, it has more mid-range values compared to the MNIST dataset. Such an input data distribution causes degradation in inference accuracy due to imprecise activation values at the first layers of the network. Besides, hidden layers of the ANNs for the CIFAR-10 dataset tend to generate mid-range values as the networks must extract complex features to classify complex images. As a result, the neuron activation values for CIFAR-10 neural

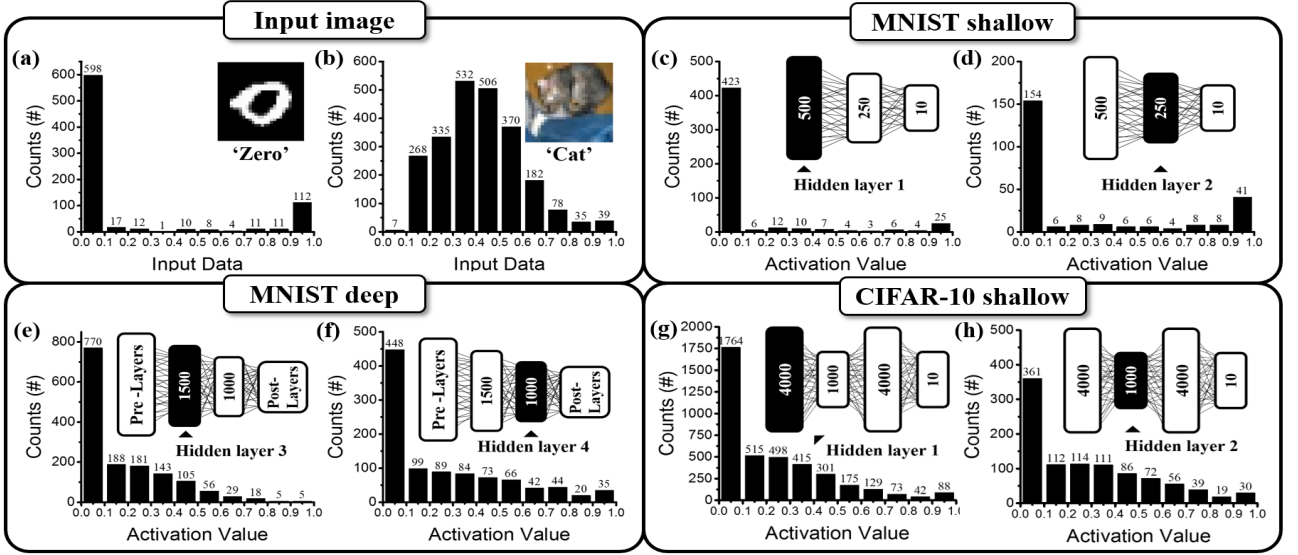


Figure 6: Distributions of input data in (a) MNIST dataset and (b) CIFAR-10. Activation values of (c) first hidden layer and (d) second hidden layer of shallow MNIST network, (e) third hidden layer and (f) fourth hidden layer of deep MNIST network, and (g) first hidden layer and (h) second hidden layer of shallow CIFAR-10 network.

networks have large portion of mid-range values even in the shallow network case (Fig. 6(g), (h)).

## 4 Proposed Methodology

Previous approaches attempted to exploit the linear sub-range in the nonlinear I-V curve for more accurate vector-matrix multiplications. However, its effectiveness was only explored for particular empirical I-V models. DAC resolution also arose as a critical problem for the cases with increased nonlinearity. Thus, there is a pressing need to address the I-V nonlinearity without such limitations. Here, we take a totally reverse approach to solve the problem; we suggest to reconstruct the neural network model itself to reflect the I-V nonlinearity by replacing linear vector-matrix multiplications.

Section 4 proposes a method to build an optimized neural network based on a device model. For the optimized neural network, the basic computation block for vector-matrix multiplications of a neural network is replaced by the nonlinear I-V model of a given device. Any device with nonlinear I-V characteristics can be adopted as far as the numerical model of the I-V curve is differentiable. By reconstructing the neural network considering the nonlinear I-V model of a given device, we aim at overcoming the causes of accuracy loss discussed in Section 3 without limiting the functionality of crossbar arrays.

### 4.1 Network Construction

A perceptron produces its activation value using transfer function and activation function as

$$y = g(f(\mathbf{w}, \mathbf{x})) \quad (6)$$

with  $\mathbf{x}$  as a vector of input values,  $\mathbf{w}$  as a vector of weights, function  $f$  as the transfer function and function  $g$  as the activation function. Conventional ANN uses weighted sum as the transfer function,

$$f(\mathbf{w}, \mathbf{x}) = \sum_i w_i x_i \quad (7)$$

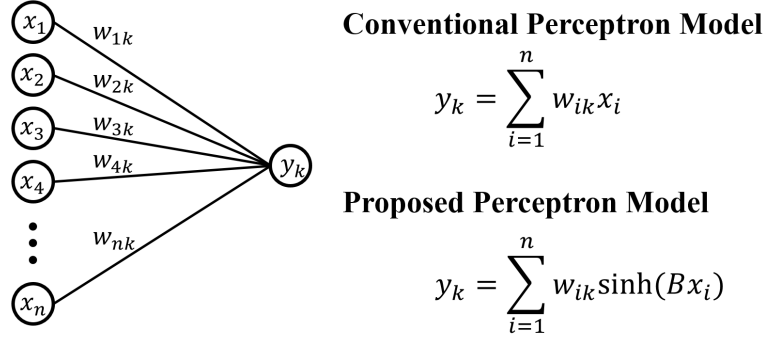


Figure 7: Conventional and proposed perceptron model

For the activation function, there are several choices such as sigmoid, tanh, and ReLU. In case of a fully connected layer with  $k$  perceptrons for an output vector  $\mathbf{y}$  and  $i$  inputs, input vector  $\mathbf{x}$  is fed to each  $k$  perceptrons with corresponding weight vectors  $\mathbf{w}_k$ . Since the weight vector and the input vector are independent, this computation can be simplified to a vector-matrix multiplication by concatenating the weight vectors into a vector matrix  $\mathbf{W}$ ,

$$\mathbf{y} = g(\mathbf{x} \cdot \mathbf{W}) \quad (8)$$

Different from conventional ANN which uses the weighted sum as the transfer function, we propose to use the numerical model of the nonlinear I-V characteristics of a given device as the transfer function of a perceptron. By introducing nonlinearity to the perceptron model itself, we can reduce the gap between the neural network model and the nonlinear I-V characteristics. As an example case, let us construct a neural network using the device from [19]. The empirical model of the device is given as Eq. (1). The equation can be simplified to

$$I(G, V) = G \sinh(BV) \quad (9)$$

As  $B$  is a characteristic constant of each RRAM device, there are two variables  $G$  and  $V$  which determine the output current. Based on the observation, we can define a transfer function as

$$f(\mathbf{w}, \mathbf{x}) = \sum_i w_i \sinh(Bx_i) \quad (10)$$

using conductance  $G_i$  as weight  $w_i$  (Fig. 7). Because the weight vector and the input vector are independent similar to the transfer function of the conventional ANN, we can also concatenate the weight vectors to simplify the computation of an output layer into a vector-matrix multiplication as

$$\mathbf{y} = g(\sinh(B\mathbf{x}) \cdot \mathbf{W}) \quad (11)$$

Because conductance value is always positive, two adjacent columns of the crossbar array are used to express a single column of weights. We decompose a single weight to a pair of positive and negative sub-weights as

$$w_i = w_i^+ - w_i^- \quad (12)$$

with  $w_i^+$  as the positive sub-weight and  $w_i^-$  as the negative sub-weight. With this expression, expected computation result can be obtained by subtracting the computation results from two adjacent columns. The transfer function is given as

$$\mathbf{y} = g(\sinh(B\mathbf{x}) \cdot \mathbf{W}^+ - \sinh(B\mathbf{x}) \cdot \mathbf{W}^-) \quad (13)$$



Besides the modifications, the proposed network use same activation functions, cost functions, optimizers and other components as conventional ANN. For the rest of the paper, we used modified Rectified Linear Unit (ReLU) as the activation function and logistic regression with softmax as the cost function. ReLU function was modified to have an upper bound as the value of maximum read voltage. We used 1 as the upper bound since  $V_{\text{read}} = 1$  V. Under the condition, logical output value of a layer could be directly fed into the next layer as the input voltage.

## 4.2 Training

The proposed network can be trained using gradient descent similar to conventional ANN. Gradient for  $k_{\text{th}}$  weight matrix in a  $n$ -layer network can be derived using chain rule as follows,

$$\frac{dE}{d\mathbf{W}_k} = \frac{dE}{ds_n} \cdot \frac{ds_n}{d\mathbf{y}_{n-1}} \cdot \frac{d\mathbf{y}_{n-1}}{ds_{n-1}} \cdot \frac{ds_{n-1}}{d\mathbf{y}_{n-2}} \cdot \dots \cdot \frac{d\mathbf{y}_{k+1}}{ds_{k+1}} \cdot \frac{ds_{k+1}}{d\mathbf{W}_k} \quad (14)$$

where  $\mathbf{s}$  stands for the result of the transfer function,  $\mathbf{y}$  stands for the result of the activation function, and  $E$  means the error according to the cost function used. After evaluating the gradient of each weight matrix, gradient descent can be applied to the proposed network to update the weights with  $\mu$  as the learning rate:

$$\mathbf{W}_k^* = \mathbf{W}_k - \mu \frac{dE}{d\mathbf{W}_k} \quad (15)$$

Each term in Eq. (14) can vary depending on the device I-V model used in the neural network. Let us derive the terms for the example case discussed above.

$$\frac{dE}{ds_n} = (O_{\text{true}} - O_{\text{predict}}) \quad (16)$$

The derivative term of error is given as the difference between desired output and predicted output since the example case uses the conventional cross-entropy loss with softmax function.

$$\frac{d\mathbf{y}_n}{ds_n} = \begin{cases} 1, & 0 \leq s_n \leq 1 \\ 0, & \text{else} \end{cases} \quad (17)$$

Eq. (17) shows the derivative term of the activation function used in the example case. At  $s_n = 0$  and  $s_n = 1$ , we assign 1 to the term for computation although the derivative cannot be explicitly defined.

$$\frac{ds_n}{d\mathbf{y}_{n-1}} = B\mathbf{W}_{n-1} \cdot \cosh(B\mathbf{y}_{n-1}) \quad (18)$$

Eq. (18) is derived by taking the derivative of the transfer function with respect to the input vector.

$$\frac{ds_n}{d\mathbf{W}_{n-1}} = \sinh(B\mathbf{y}_{n-1}) \quad (19)$$

Eq. (19) can be obtained by taking the derivative of the transfer function with respect to the weight matrix. For both Eq. (18) and Eq. (19), Eq. (12) is not considered to simplify the equation. However, simplified sub-weight decomposition can still be used by defining two sub-weights in adjacent columns as

$$w^+ = \begin{cases} w, & w \geq 0 \\ 0, & w < 0 \end{cases} \quad (20)$$

$$w^- = \begin{cases} 0, & w \geq 0 \\ |w|, & w < 0 \end{cases} \quad (21)$$

given a weight  $w$ . With all the derivative terms above, the proposed network can be trained with gradient descent algorithm.

## 5 Evaluation

Several evaluation networks based on the example case in Section 4 were trained and simulated using MATLAB. A shallow (784-500-250-10) network and a deep (784-2500-2000-1500-1000-500-10) network were trained with MNIST dataset. A shallow (2352-4000-1000-4000-10) network for CIFAR-10 dataset was trained for additional analysis.

Because  $e^{d/d_0}$  in Eq. (1) is simplified to  $G$  and used as the weight in the proposed network, trained weights must be mapped back into the range of the term  $e^{d/d_0}$  that actual device exhibits. Manual fitting described in Fig. 2 showed that maximum value for  $e^{d/d_0}$  is  $e^{-8}$  and the minimum value is  $e^{-14}$ .  $k$  value was also measured as 7.5 through fitting.

Sub-weights were mapped to the available range via linear transformation. The linear transformation was as follows,

$$e_{\text{mapped}}^{d/d_0} = W^\pm * \frac{e^{-8} - e^{-14}}{\max(W^+, W^-)} + e^{-14} \quad (22)$$

Then, simulated current was sampled from each sub-weight column. After subtracting the sampled value of each negative sub-weight column from the sampled value of the corresponding positive sub-weight column, output of the transfer function was retrieved with inverse function of Eq. (22).

### 5.1 MNIST Inference Accuracy

Example MLP models were trained with SGD. For the shallow network, we used 30 epochs for training. Learning rate was set to  $5 \times 10^{-6}$  at first and to  $1 \times 10^{-6}$  after 16 epochs. The deep network was trained for 65 epochs. Learning rate for the deep network was set to  $2 \times 10^{-6}$  at first and  $7 \times 10^{-7}$  after 15 epochs. To demonstrate the robustness against nonlinear I-V characteristics of various devices, model networks with various  $k$  values were also tested. Same SGD was used but learning rate varied for each case. Fig. 8 demonstrates the inference evaluation results.

The proposed network did not show noticeable degradation in accuracy for various  $k$  values while networks based on naive mapping exhibited drastic accuracy loss as  $k$  increased. Accuracies for the case with the example device ( $k = 7.5$ ) were as shown in Table 1. This result shows that the proposed network can minimize the error demonstrated in Section 3.1.

Table 1: Evaluation results for device with  $k = 7.5$  (%)

|                  | Shallow MNIST | Deep MNIST | Shallow CIFAR-10 |
|------------------|---------------|------------|------------------|
| Ideal case       | 94.8          | 97.43      | 54.97            |
| Naive mapping    | 87.90         | 9.05       | 41.94            |
| Proposed network | 96.74         | 96.91      | 52.09            |

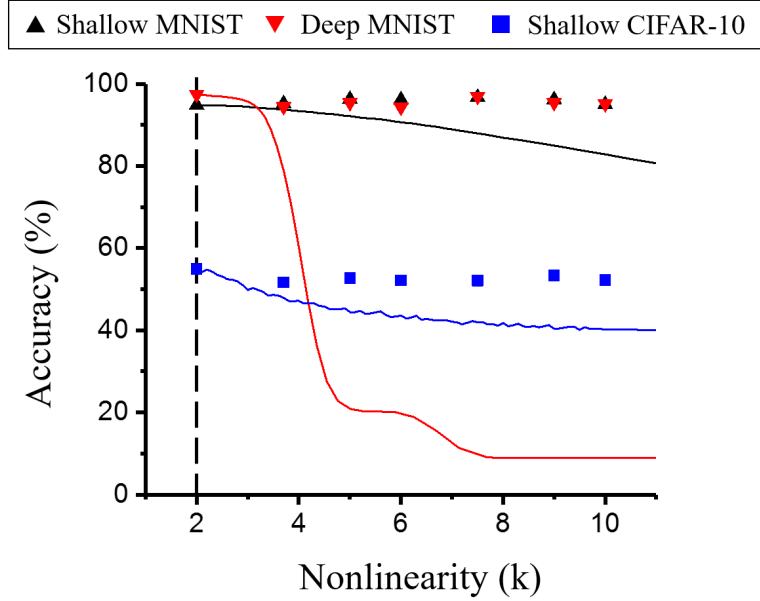


Figure 8: Inference accuracies of networks trained using the proposed method for various  $k$  values. Simulation results from naive mapping are in solid lines and that from proposed mapping are in symbols. Data for Shallow MNIST, Deep MNIST, and Shallow CIFAR-10 cases are in black, red, blue color respectively. Proposed networks (symbol) do not show significant dependency on  $k$  value while naively mapped networks (line) do.

## 5.2 CIFAR-10 Accuracy

Another MLP with the proposed model was trained for CIFAR-10 dataset as a proof-of-concept model. The training set was randomly cropped into a set of 28x28 images and exposed to random image distortions. The random distortions included horizontal flips and contrast and brightness adjustments. Then, the images were divided by 255 to ensure input range between 0 and 1 because the raw data are unsigned 8-bit integers. The MLP was trained for this image dataset using SGD for 100 epochs. Learning rate was fixed to  $1 \times 10^{-6}$ . MLP networks with varying  $k$  values for CIFAR-10 dataset were also trained.

CIFAR-10 classification results also showed that the accuracy of the proposed network does not vary much over wide range of the  $k$  value. As illustrated in Table 1, the proposed scheme achieved better inference accuracy than the naive mapping case. Since MLP is not capable of achieving high inference accuracy for such complex task, inference accuracies of both ideal and proposed network were limited. Still, the result demonstrates that the proposed network can also circumvent the error discussed in Section 3.2.

## 6 Application to more complex I-V model

In addition to the device model derived in [19], there are several other resistive devices and corresponding empirical I-V models [15, 25]. While each device has a different I-V model, the proposed method is generally applicable as far as the I-V model is differentiable.

To demonstrate the validity of this claim, we built and tested three MLPs using the device model obtained by manually fitting the I-V characteristics of the device in [26] as the transfer function. We intentionally made a complex empirical model to verify our assertion. The empirical model is as follows,

$$I(w, V) = e^{Aw+B}(e^{CV^{w+D}} - 1) \quad (23)$$

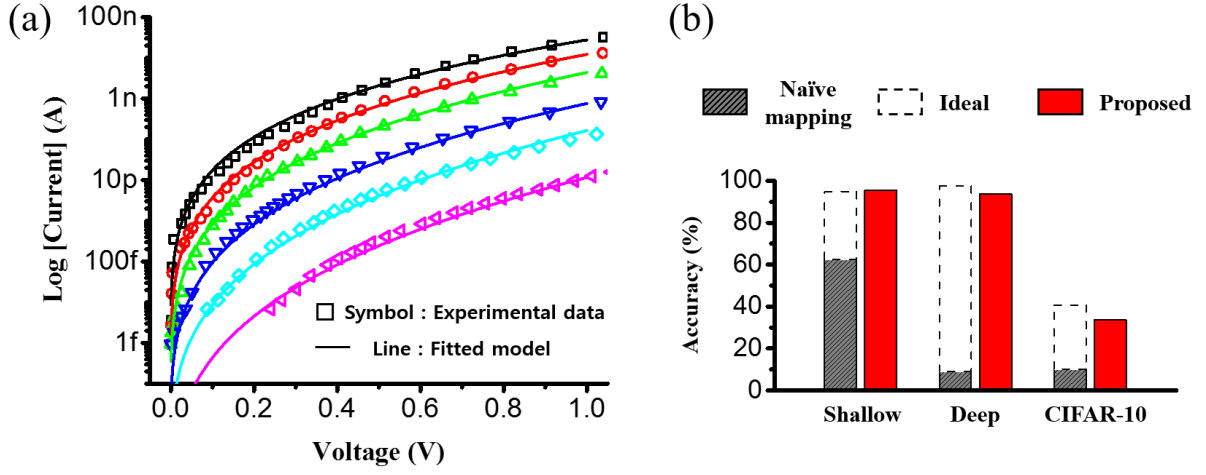


Figure 9: (a) Experimental data and fitted curve of another device under various resistance states [26]. (b) Accuracy comparison between previous and proposed approaches. This result shows that our proposed method can also be applied to complex device models.

with  $A = -53.59$ ,  $B = -37.058$ ,  $C = 20$ ,  $D = 0.2$ , and  $0 < W < 0.15$ . Note that the model is more complex than Eq. (1). Fitting result of the model was as shown in Fig. 9(a). MLPs for MNIST and CIFAR-10 with same structures as the ones described earlier were built and evaluated. To train the network using gradient descent, same chain rule as Eq. (14) was derived. Since the cost function and activation function were the same, derivative terms Eq. (16) and Eq. (17) remained the same. However, for other terms, the derivation process was very different because the weight term and the input voltage term were correlated. Because of the correlation, we had to consider the simplified sub-weight decomposition while choosing the transfer function as

$$s = I(w^+, V) - I(w^-, V) \quad (24)$$

where Eq. (20) and Eq. (21) lead to

$$s_{n,j} = \begin{cases} \sum_i (e^{Aw_{n-1,i,j}+B}(e^{Cx_{n-1,i}^{w_{n-1,i,j}+D}} - 1) - e^B(e^{Cx_{n-1,i}^D} - 1)), & w_{n-1,i,j} \geq 0 \\ \sum_i (-e^{-Aw_{n-1,i,j}+B}(e^{Cx_{n-1,i}^{-w_{n-1,i,j}+D}} - 1) + e^B(e^{Cx_{n-1,i}^D} - 1)), & w_{n-1,i,j} < 0 \end{cases} \quad (25)$$

for the  $j_{\text{th}}$  neuron in  $n_{\text{th}}$  layer. Since the transfer function must be computed element-wise, derivative terms of the transfer function with respect to the input vector and the weight vector had to be also expressed element-wise as

$$\frac{ds_{n,j}}{dw_{n-1,i,j}} = \begin{cases} \sum_i e^{Cx_{n-1,i}^{w_{n-1,i,j}+D}+Aw_{n-1,i,j}+B}(Cx_{n-1,i}^{w_{n-1,i,j}+D} \ln V - A), & w_{n-1,i,j} \geq 0 \\ \sum_i e^{Cx_{n-1,i}^{-w_{n-1,i,j}+D}-Aw_{n-1,i,j}+B}(Cx_{n-1,i}^{-w_{n-1,i,j}+D} \ln V - A), & w_{n-1,i,j} < 0 \end{cases} \quad (26)$$

$$\frac{ds_{n,j}}{dx_{n-1,i}} = \begin{cases} \sum_i x_{n-1,i}^{w_{n-1,i,j}+D-1} e^{Cx_{n-1,i}^{w_{n-1,i,j}+D}+Aw_{n-1,i,j}+B-1} (Cw_{n-1,i,j} + CD) - CDx_{n-1,i}^{D-1} e^{Cx_{n-1,i}^D+B}, & w_{n-1,i,j} \geq 0 \\ \sum_i -x_{n-1,i}^{-w_{n-1,i,j}+D-1} e^{Cx_{n-1,i}^{-w_{n-1,i,j}+D}-Aw_{n-1,i,j}+B-1} (-Cw_{n-1,i,j} + CD) + CDx_{n-1,i}^{D-1} e^{Cx_{n-1,i}^D+B}, & w_{n-1,i,j} < 0 \end{cases} \quad (27)$$

Since Eq. (27) was not defined for  $x = 0$ , we substituted 0 as the term for such cases during the training phase. Using the terms above, we could obtain the accuracy results as shown in Fig. 9(b). The results showed that the proposed method is applicable to very complex nonlinear device I-V model.

## 7 Conclusion & Future Research

In this paper, we aimed for accurate computation of neural networks using emerging NVM crossbar arrays. We first analyzed the cause of inaccuracy in using naive mapping method. Simulation results showed that mid-range activation values induce computation error for complex tasks and networks. To overcome the accuracy degradation due to the nonlinear I-V characteristics of emerging NVM devices, we proposed a method to construct neural networks optimized to the characteristics. Neural networks based on empirical models of two RRAM devices were trained and tested to classify MNIST and CIFAR-10 dataset using the proposed approach. Results showed that proposed networks could achieve inference accuracies comparable to the baseline. Also, proposed networks did not show accuracy degradation for a variety of nonlinearity values while naive mapping showed significant accuracy loss.

Classification accuracy for the shallow CIFAR-10 case was limited due to the inherent lack of computing power of the MLP structure. Thus, next step of this research will be applying the proposed methodology to more complex neural networks such as CNN and RNN. Also, we did not take into account other non-ideal characteristics of emerging NVM arrays such as I-R drop for simulations. Another next step of this work will be to simulate and analyze the effects of such characteristics.

## 8 Acknowledgement

This work was in part supported by the Ministry of Science, ICT and Future Planning of South Korea under the "IT Consilience Creative Program" (IITP-R0346-16-1007) and "Nano-Material Technology Development Program" (No. 2016910249) through the National Research Foundation of Korea (NRF). It is also in part supported by the Industrial Technology Innovation Program (10067764) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

## References

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] K. Chellapilla, S. Puri, and P. Simard, "High Performance Convolutional Neural Networks for Document Processing," *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.
- [3] T. Gokmen and Y. Vlasov, "Acceleration of deep neural network training with resistive cross-point devices: Design considerations," *Frontiers in Neuroscience*, vol. 10, no. JUL, pp. 1–13, 2016.
- [4] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, R. S. Williams, and J. Yang, "Dot-Product Engine for Neuromorphic Computing: Programming 1T1M Crossbar to Accelerate Matrix-Vector Multiplication," *IEEE Design Automation Conference*, p. p. 19, 2016.
- [5] A. Shafiee, A. Nag, N. Muralimanohar, and R. Balasubramonian, "ISAAC : A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars," *ISCA 2016*, 2016.
- [6] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "Prime: A novel processing-in-memory architecture for neural network computation in rram-based main memory," in *Proceedings of the 43rd International Symposium on Computer Architecture*, pp. 27–39, IEEE Press, 2016.
- [7] S. Yu and P.-Y. Chen, "Emerging memory technologies: recent trends and prospects," *IEEE Solid-State Circuits Magazine*, vol. 8, no. 2, pp. p43–p56, 2016.
- [8] M. Hu, H. Li, Y. Chen, Q. Wu, G. S. Rose, and R. W. Linderman, "Memristor crossbar-based neuromorphic computing system: A case study," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 10, pp. 1864–1878, 2014.

- [9] M. Prezioso, F. Merrih-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors.," *Nature*, vol. 521, no. 7550, pp. 61–4, 2015.
- [10] 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 *Proceedings of the 53rd Annual Design Automation Conference*, p. 125, ACM, 2016.
- [11] G. W. Burr, R. M. Shelby, A. Sebastian, S. Kim, S. Kim, S. Sidler, K. Virwani, M. Ishii, P. Narayanan, A. Fumarola, *et al.*, "Neuromorphic computing using non-volatile memory," *Advances in Physics: X*, vol. 2, no. 1, pp. 89–124, 2017.
- [12] G. W. Burr, R. M. Shelby, S. Sidler, C. Di Nolfo, J. Jang, I. Boybat, R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti, B. N. Kurdi, and H. H., "Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element," *IEEE Transactions on Electron Devices*, vol. 62, no. 11, pp. 3498–3507, 2015.
- [13] M. Prezioso, F. Merrih-Bayat, B. Chakrabarti, and D. Strukov, "Rram-based hardware implementations of artificial neural networks: progress update and challenges ahead," in *SPIE OPTO*, pp. 974918–974918, International Society for Optics and Photonics, 2016.
- [14] M. A. Zidan, H. A. H. Fahmy, M. M. Hussain, and K. N. Salama, "Memristor-based memory: The sneak paths problem and solutions," *Microelectronics Journal*, vol. 44, no. 2, pp. 176–183, 2013.
- [15] Y. Deng, P. Huang, B. Chen, X. Yang, B. Gao, J. Wang, L. Zeng, G. Du, J. Kang, and X. Liu, "Rram crossbar array with cell selection device: A device and circuit interaction study," *IEEE Transactions on Electron Devices*, vol. 60, no. 2, pp. 719–726, 2013.
- [16] P. Gu, B. Li, T. Tang, S. Yu, Y. Cao, Y. Wang, and H. Yang, "Technological exploration of rram crossbar array for matrix-vector multiplication," in *The 20th Asia and South Pacific Design Automation Conference*, pp. 106–111, 2015.
- [17] B. Li, P. Gu, Y. Shan, Y. Wang, Y. Chen, and H. Yang, "Rram-based analog approximate computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 12, pp. 1905–1917, 2015.
- [18] M. Hu, J. P. Strachan, Z. Li, R. Stanley, *et al.*, "Dot-product engine as computing memory to accelerate machine learning algorithms," in *Quality Electronic Design (ISQED), 2016 17th International Symposium on*, pp. 374–379, IEEE, 2016.
- [19] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, and H. S. P. Wong, "A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation," *Advanced Materials*, vol. 25, no. 12, pp. 1774–1779, 2013.
- [20] S. H. Misha, N. Tamanna, J. Woo, S. Lee, J. Song, J. Park, S. Lim, J. Park, and H. Hwang, "Effect of Nitrogen Doping on Variability of TaOx -RRAM for Low-Power 3-Bit MLC Applications," *ECS Solid State Letters*, vol. 4, no. 3, pp. P25–P28, 2015.
- [21] J. Zhou, F. Cai, Q. Wang, B. Chen, S. Gaba, and W. D. Lu, "Very low-programming-current RRAM with self-rectifying characteristics," *IEEE Electron Device Letters*, vol. 37, no. 4, pp. 404–407, 2016.
- [22] S. Gao, F. Zeng, F. Li, M. Wang, H. Mao, G. Wang, C. Song, and F. Pan, "Forming-free and self-rectifying resistive switching of the simple Pt/TaOx/n-Si structure for access device-free high-density memory application.," *Nanoscale*, vol. 7, no. 14, pp. 6031–8, 2015.
- [23] M. Hu, H. Li, Q. Wu, G. S. Rose, and Y. Chen, "Memristor crossbar based hardware realization of BSB recall function," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 498–503, 2012.
- [24] P. O. Vontobel, W. Robinett, P. J. Kuekes, D. R. Stewart, J. Straznicky, and R. S. Williams, "Writing to and reading from a nano-scale crossbar memory based on memristors," *Nanotechnology*, vol. 20, no. 42, p. 425204, 2009.
- [25] K. Sonoda, A. Sakai, M. Moniwa, K. Ishikawa, O. Tsuchiya, and Y. Inoue, "A compact model of phase-change memory based on rate equations of crystallization and amorphization," *IEEE Transactions on Electron Devices*, vol. 55, no. 7, pp. 1672–1681, 2008.
- [26] S. Park, J. Noh, M.-L. Choo, A. M. Sheri, M. Chang, Y.-B. Kim, C. J. Kim, M. Jeon, B.-G. Lee, B. H. Lee, and H. Hwang, "Nanoscale RRAM-based synaptic electronics: toward a neuromorphic computing device.," *Nanotechnology*, vol. 24, no. 38, p. 384009, 2013.