# Reduction and IR-drop Compensations Techniques for Reliable Neuromorphic Computing Systems

Beiye Liu[1], Hai Li[6] Yiran Chen[7]
Department of Electrical and Computer Engineering
University of Pittsburgh
Pittsburgh, PA, 15261
{bel34[1], hal66[6], yic52[7]}@pitt.edu

Xin Li[2]
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA, 15213
xinli@cmu.edu

Tingwen Huang[3]
Texas A&M University
Doha, Qatar
PO Box 23874
tingwen.huang@qatar.tamu.edu

Qing Wu[4], Mark Barnell[5]
Information Directorate
Air Force Research Laboratory
Rome, NY 13441-4505
{qing.wu.[4], mark.barnell.[5]}@us.af.mil

## ABSTRACT

Neuromorphic computing system (NCS) is a promising architecture to combat the well-known memory bottleneck in Von Neumann architecture. The recent breakthrough on memristor devices made an important step toward realizing a low-power, small-footprint NCS on-a-chip. However, the currently low manufacturing reliability of nano-devices and the voltage IR-drop along metal wires and memristors arrays severely limits the scale of memristor crossbar based NCS and hinders the design scalability. In this work, we propose a novel system reduction scheme that significantly lowers the required dimension of the memristor crossbars in NCS while maintaining high computing accuracy. An IR-drop compensation technique is also proposed to overcome the adverse impacts of the wire resistance and the sneak-path problem in large memristor crossbar designs. Our simulation results show that the proposed techniques can improve computing accuracy by 27.0% and 38.7% less circuit area compared to the original NCS design.

## 1. INTRODUCTION

Computer technology has been experiencing great revolutions in its two foundation stones: semiconductor manufacturing and computing architecture: On the one hand, the scaling of conventional CMOS devices is approaching the limit [1]. Emerging nano-devices, such as spintronic [2] and resistive devices (memristor) [8], have been under extensive investigation and studies. On the other hand, the well-known "memory wall" challenge in Von Neumann architecture [3], i.e., the ever-increasing gap between CPU performance and memory bandwidth, has motivated many research efforts on new computer architectures.

Neuro-biological architecture is a promising alterative to Von Neumann architecture. After twenty-year through, neuromorphic computing, which denotes the VLSI realization of neuro-biological architecture, is recently revitalized by the discovery of nanoscale resistive devices, e.g., the memristors [8]. The similarity between the programmable resistance state of memristors and the variable synaptic strengths of biological synapses can lead to a dramatically simplified structure of neural network circuits [4]. Moreover, the crossbar structure, which is the densest interconnect topology that can be achieved by modern semiconductor manufacturing, further boosts the integration density and power efficiency of memristor-based neuromorphic computing systems (NCS) [5] to the levels of $10^{10}$ synapses per square inch and over one trillion operations per second (TOPS) per Watt, respectively.

However, the implementation of an NCS with memristor-based crossbar (MBC) is facing several major technical challenges, including: 1) the parametric variability, fabrication defects and stochastic programming properties of memristors [5]; and 2) the IR drop along the resistance network composed of metal wire and memristors. The analysis of the impact of IR-drop on MBC-based digital memory shows a 64×64 MBC-based memory already has severe voltage degradation [6]. Following the increase of the memristor crossbar size, the impact of the IR drops becomes more critical, resulting in the performance variations or even functional failures of the NCS.

The impacts of memristor variation on NCS have been extensively studied [5]. However, the IR-drop caused physical limitation and reliability issue in MBCs still lacks of investigations. In this work, we first formulate the effect of IR-drop in NCS designs and evaluate its impact. In order to enhance the computing capacity and reliability of NCS, we propose a system reduction scheme that can effectively reduce the required MBC size for a specific problem while still maintaining high computation accuracy and robustness, enabling simpler and more scalable NCS implementations. To further improve the robustness of NCS, we propose a novel design method that can actively compensate the IR-drop induced signal degradations in training and computing. Note that system reduction and IR-drop compensation methods are implemented at different design levels and thus, complementary to each other. Experiment results demonstrate much smaller implementation area (i.e., 61.3% of original design circuit area) and better computing robustness (i.e., 27.0% computing accuracy improvement) of NCS after combining these two approaches.

## 2. Preliminary

### 2.1 Memristor Basics

As predicted by Prof. Leon Chua [10], memristor is the fourth fundamental circuit element uniquely defining the relationship between magnetic flux and electrical charge. The resistance state of a memristor can be programmed by applying current or voltage. In 2008, HP Labs reported that the memristive effect was realized by moving the doping front along a $TiO_2$ thin-film device [11].

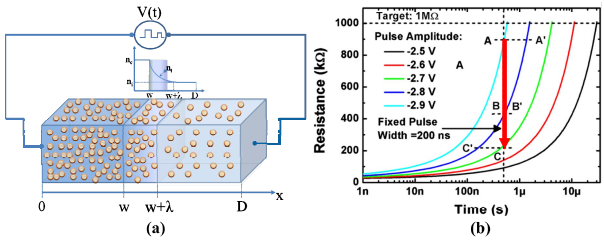


Fig. 1: (a) Metal-oxide memristor [7]. (b) Device programming [8].

Fig. 1(a) depicts an ion migration filament model of metal-oxide memristors [7]. A metal-oxide layer is sandwiched between two metal electrodes. During *reset* process, the memristor switches from low resistance state (LRS) to high resistance state (HRS). The oxygen ions migrate from the electrode/oxide interface and recombine with the oxygen vacancies. A partially ruptured conductive filament region with a high resistance per unit length ($R_{off}$) is formed on the left of the conductive filament region with a low resistance per unit length ($R_{on}$). During *set* process, the memristor switches from HRS to LRS. The ruptured conductive filament region shrinks. The resistance of a memristor can be programmed to any arbitrary value between LRS and HRS by applying a programming current or voltage with different pulse widths or magnitudes. Note that the relationship between the programming voltage amplitude/pulse-width and the memristor resistance change is usually a highly nonlinear function, as shown in Fig. 1(b) [8].

## 2.2 MBC-based NCS

Fig. 2(a) depicts a conceptual overview of a neural network that can be implemented with a MBC-based NCS in Fig. 2(b). Two groups of neurons are connected by a set of synapses. Input neurons send voltage signals to MBC. The output neurons collect the information (current) from the input neurons through the synapses (MBC) and process them with an activation function. The synapses apply different weights (synaptic strengths) on the information during the transmission. In general, the relationship between the activity patterns of the input neurons $\boldsymbol{x}$ and the output neurons $\boldsymbol{y}$ can be described as [5]:

$$\boldsymbol{y}_n = \boldsymbol{W}_{n\times m} \cdot \boldsymbol{x}_m. \quad (1)$$

Here the weight matrix $\boldsymbol{W}_{n\times m}$ denotes the synaptic strengths between the two neuron groups.

***Recall***: The computation process defined by Eq. (1) is called "recall". As shown in Fig. 2(b), during the recall process of a MBC-based NCS, $\boldsymbol{x}$ is mimicked by the input voltage vector applied to the word-lines (WLs) of the MBC while the bit-lines (BLs) are grounded. Each memristor is programmed to a resistance state representing the weight of the correspondent synapse. The current along each BL of the MBC is collected and converted to the output voltage vector $\boldsymbol{y}$ by "neurons", e.g., CMOS analog circuit or emerging domain wall devices. "Neurons" integrate and quantize the output. The matrix $\boldsymbol{W}_{n\times m}$ is often implemented by two MBCs, which represent the positive and negative elements of $\boldsymbol{W}_{n\times m}$, respectively.

***Training***: Another important operation of the MBC-based NCS is "training". For a neural network model, there are two types of training schemes: *open-loop training* and *close-loop training*. In *Open-loop* training, weight connection matrix $\boldsymbol{W}$ is directly calculated based on the stored patterns $\boldsymbol{a}_q$ $(q = 1,2\cdots p)$. For example, by Hebbian learning rule, the $\boldsymbol{W}$ can be calculated as [15]:

$$\boldsymbol{W} = \sum_{q=1}^{p} \boldsymbol{a}_q \cdot \boldsymbol{a}'_q. \quad (2)$$

*Close-loop training*, however, denotes a recursive algorithm, i.e., gradient descent, that updates the $\boldsymbol{W}$ iteratively with the feedback from output [22].

Both *open-loop training* and *close-loop training* can be implemented with "pre-design & mapping" method, i.e., directly programming the MBC to a resistance $\boldsymbol{R}$ that represents a pre-calculated $\boldsymbol{W}$, say, $\boldsymbol{R}=1/\boldsymbol{W}$. Also, *close-loop training* can be realized in an iterative method which adaptively tunes the resistance state of the MBC to the target state based on the distance between the current output and the target output, as proposed in [22]. Although such method may conceptually achieve higher resolution and better robustness than "pre-design & mapping", its hardware implementation is generally expensive in terms of training time, energy consumption, and design cost. In this work, we focus on only "pre-design & mapping" method.

During the programming of MBC, different amplitude and duration of programming pulses are directly applied to the target memristor based on the pre-designed $\boldsymbol{R}$: the voltages of the WL and BL connecting the target memristor are set to $+V_{bias}$ and $GND$, respectively, while all other WLs and BLs are connected to $+V_{bias}/2$. Hence, only the target memristor is applied with the full $V_{bias}$ above the threshold that can change the device's resistance state while the rest of memristors in the crossbar remain unchanged because they are only half selected with a voltage of $V_{bias}/2$ [7].

# 3. IMPACT OF IR-DROP

## 3.1 Impact of IR-Drop on MBC

In an MBC, the voltage applied to the two terminals of a memristor is affected by the device location in the crossbar and the resistance states of all other memristors. In [7], the author explained that in the worst case, both reading (recall) and writing (training) of the MBC will encounter severe reliability issues when the array size is beyond 64×64. Although an NCS intrinsically can tolerate certain random errors in recall process, IR-drop remains an issue in NCS training.

Fig. 3(a) depicts the distribution of the actual programming voltage $\boldsymbol{V'}$ on each memristor in a 128×128 MBC during the training process. Here $V_{bias}$ =2.9V. $\boldsymbol{V'}_{ij}$ is the voltage actually applied to the memristor between $WL_i$ and $BL_j$. The largest IR drop normally occurs at the far-end of the WL and BL (i.e., $\boldsymbol{V'}_{(128,128)}$). The smallest/largest voltage degradation (IR-drop) occurs when all memristors are at their HRS/LRS. Fig. 3(b) shows that in the worst case, the largest IR-drop quickly increases to an unacceptable level as the crossbar size increases. It greatly decreases the programmability of the MBC and degrades computation accuracy of the NCS. Degradation also occurs in recall process as shown in Fig. 3(c) and (d).

## 3.2 Problem Formulation

### *3.2.1 Training*

Normally the training of an MBC starts with an initial state where all memristors are at their HRS. To program the initialized MBC ($\boldsymbol{R}_{HRS}$) to the target memristor resistance state $\boldsymbol{R}$ that representing weight matrix $\boldsymbol{W}$, a training time matrix $\boldsymbol{T}$ is generated based on the characterized relationship between the memristor resistance change and the programming time and voltage [8]:

$$\boldsymbol{R} = f(\boldsymbol{T}, \boldsymbol{V}, \boldsymbol{R}_{HRS}), \quad (3)$$

where $\boldsymbol{V}$ is the ideal programming voltage ($V_{(i,j)}=V_{bias}$). After including the impact of IR-drop, the actual trained MBC resistance state is $\boldsymbol{R'}=f(\boldsymbol{T},\boldsymbol{V'},\boldsymbol{R}_{HRS})$. Thus, if the $\boldsymbol{V'}$ deviates from the ideal $\boldsymbol{V}$ due to IR-drop, the actual trained MBC $\boldsymbol{R'}$ will be

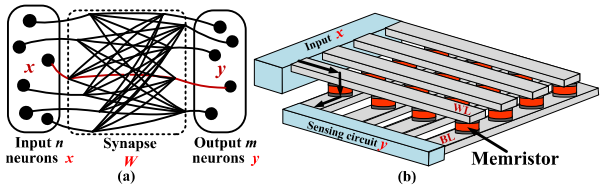


Fig. 2: (a) Conceptual overview of a neural [5]. (b) Circuit architecture of a MBC-based NCS [4].

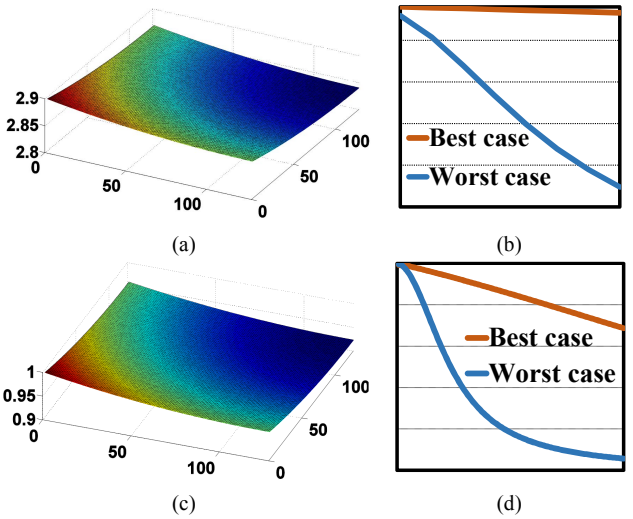


Fig. 3: (a) Write voltage distribution on a 128×128 all-HRS-memristor crossbar (the best case). (b) Voltage degradation vs. crossbar size. (c) Read voltage distribution on a 128×128 all-HRS-memristor crossbar (the best case). (d) Read current discrepancy.

distinctive from $\boldsymbol{R}$. The difference between $\boldsymbol{R}$ and $\boldsymbol{R'}$ depends on the size of MBC. As shown in Fig. 1(b), when the programming voltage arriving at the memristor degrades from the ideal 2.9V to 2.7V (6.8% off). The programmed memristor resistance drifts from 900KΩ (point "A") to 200KΩ (point "C") at a programming duration of 0.4μs. More detailed experiments will be presented in Section 6.1.

### *3.2.2 Recall*

Normally during recall process, the MBC is read column by column, e.g., the WLs are connected to a certain input pattern and BLs are all grounded. As a result, the IR-drop induced voltage degradation demonstrates different patterns from that in training process. For example, when all WLs and BLs are respectively connected to 1 (1V) and 0 (GND), the ideal voltage distribution $\boldsymbol{V}$ of a 128×128 MBC should be an all-ones matrix and the ideal output will be:

$$\boldsymbol{y} = \boldsymbol{W} \circ \boldsymbol{V} \cdot \boldsymbol{x} = \boldsymbol{W} \cdot \boldsymbol{x}, \tag{4}$$

where "∘" denotes the Hadamard product of two matrices and we assume the MBC is ideally trained. However, as shown in Fig. 3(c), the actual voltage distribution $\boldsymbol{V^*}$ deviates from $\boldsymbol{V}$ and generates the actual output as:

$$\boldsymbol{y}^* = (\boldsymbol{W} \circ \boldsymbol{V}^*) \cdot \boldsymbol{x} = \boldsymbol{W}^* \cdot \boldsymbol{x}. \tag{5}$$

Here, we define $\boldsymbol{W^*}$ as the distorted weight matrix producing the actual current output of the MBC when IR-drop is taken into account. As MBC is a pure resistance network, $\boldsymbol{W^*}$ is a function of memristor resistance state and wire resistance $R_{\_wire}$:

$$\boldsymbol{W}^* = g(\boldsymbol{R}, R_{\_wire}). \tag{6}$$

Here $\boldsymbol{R}$ is the target memristor resistance state. Fig. 3(d) shows that $\boldsymbol{y^*}$ is directly determined by $\boldsymbol{V^*}$.

## 4. MBC SYSTEM REDUCTION

The impact of IR-drop is heavily determined by the size of the MBC. Hence, if we can reduce the scale of the involved computation on the MBC, the required size of the MBC will decrease and the computation reliability of the NCS will improve.

### 4.1 Weight Matrix Approximation

The first step of our proposed MBC system reduction scheme is to approximate the weight matrix $\boldsymbol{W}$ ($n \times m$) in Eq. (1). In general, for any given weight matrix $\boldsymbol{W}$, we can leverage singular value decomposition (SVD) method to approximate $\boldsymbol{W}$ as [14]:

$$\boldsymbol{W} = \boldsymbol{U} \sum \boldsymbol{V} \approx \boldsymbol{W}_{appx} = \sum_{i=1}^{r} \delta_i \cdot \boldsymbol{u}_i \cdot \boldsymbol{v}_i, \tag{7}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary matrices, $\boldsymbol{\Sigma}$ is an rectangular diagonal matrix with singular values of $\boldsymbol{W}$. $\delta_i$ ($i$=1,…r) are the first $r$ (i.e., the rank of $\boldsymbol{W}_{appx}$) singular values of $\boldsymbol{W}$. $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the approximate left and right singular vectors of $\boldsymbol{W}$ [14], respectively. The sequence of $\delta_i$ indicates the weights of each item of $\boldsymbol{u}_i \cdot \boldsymbol{v}_i$. By collecting a few multiplication product of $\boldsymbol{u}_i$ (an $n \times 1$ vector) and $\boldsymbol{v}_i$ (a $1 \times m$ vector), we can obtain a very good approximation of $\boldsymbol{W}$. The difference between $\boldsymbol{W}$ and $\boldsymbol{W}_{appx}$, that is $\Delta\boldsymbol{W} = \|\boldsymbol{W} - \boldsymbol{W}_{appx}\|$, is decided by the coverage of $\sum_{i=1}^{r} \delta_i$ on the overall summed $\sum_{i=1}^{m} \delta_i$. The difference, hence, $\Delta\boldsymbol{W}$ can be controlled by the value of $r$.

In general, a larger rank $r$ leads to a better approximation of $\boldsymbol{W}$ but increases MBC size and training time cost. However, increasing $r$ does not necessarily result in a more robust MBC hardware implementation due to the following reasons: First, $r$ is limited by an upper bound, say, the rank of $\boldsymbol{W}$. Increasing $r$ beyond this upper bound is meaningless; Second, $r$ solely defines the size of one dimension of the reduced MBC (say, $n \times r$, which will be shown in the next section). Increasing $r$ will directly aggravate the impact of IR-drop.

For the above reasons, a threshold "ε" of singular value coverage is heuristically predefined for $r$ selection during the approximation of $\boldsymbol{W}$. Here ε∈ [0,1]. $r$ is then selected as:

$$min: r, s.t.\ 1 - \frac{\sum_{i=1}^{r} \delta_i}{\sum_{i=1}^{m} \delta_i} < \varepsilon. \tag{8}$$

The efficacy of this select strategy will be shown in Section VI.B.

### 4.2 One-dimensional (1-D) Reduction

Based on the approximation result, we are able to transform the weight connection function in Eq. (1) as:

$$\begin{aligned} \boldsymbol{W} \cdot \boldsymbol{x} &\approx (\textstyle\sum_{i=1}^{r} \delta_i \cdot \boldsymbol{u}_i \cdot \boldsymbol{v}_i) \cdot \boldsymbol{x} \\ &= (\delta_1 \cdot \boldsymbol{u}_1) \cdot (\boldsymbol{v}_1 \cdot \boldsymbol{x}) + (\delta_2 \cdot \boldsymbol{u}_2) \cdot (\boldsymbol{v}_2 \cdot \boldsymbol{x}) \cdots (\delta_r \cdot \boldsymbol{u}_r) \cdot (\boldsymbol{v}_r \cdot \boldsymbol{x}) \\ &= [\delta_1 \cdot \boldsymbol{u}_1 \ldots \delta_r \cdot \boldsymbol{u}_r] \cdot \begin{bmatrix} \boldsymbol{v}_1 \cdot \boldsymbol{x} \\ \vdots \\ \boldsymbol{v}_r \cdot \boldsymbol{x} \end{bmatrix} = [\delta_1 \cdot \boldsymbol{u}_1 \ldots \delta_r \cdot \boldsymbol{u}_r] \cdot \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_r \end{bmatrix} \cdot \boldsymbol{x} \\ &= \boldsymbol{W}_{\boldsymbol{left}} \cdot \boldsymbol{W}_{\boldsymbol{right}} \cdot \boldsymbol{x}, \end{aligned} \tag{9}$$

where

$$\boldsymbol{W}_{left} = [\delta_1 \cdot \boldsymbol{u}_1 \ldots \delta_r \cdot \boldsymbol{u}_r], \boldsymbol{W}_{right} = \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_r \end{bmatrix}. \tag{10}$$

Here $\boldsymbol{W}$ was originally represented on an $n \times m$ MBC and $m \times 1$ vector $\boldsymbol{x}$ is represented by the input voltage vector of the MBC.

Eq. (9) and (10) show that the connection function can be transformed to a new two-stage system that consists of a $n \times r$ weight matrix $\boldsymbol{W}_{left}$ and a $r \times m$ weight matrix $\boldsymbol{W}_{right}$. Note that $r << n$ or $m$.

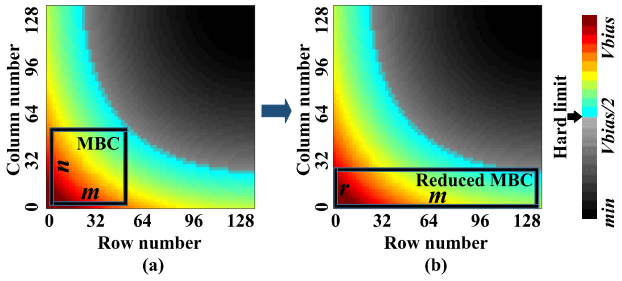


Fig. 4: System reduction improves reliability.

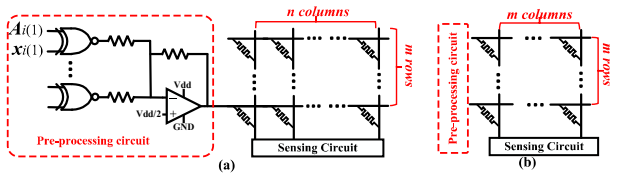


Fig. 5: Conceptual schematics of (a) One-dimensional reduction (b) Two-dimensional reduction.

We named this method as one-dimensional (1-D) reduction, which has several significant advantages: First, after the original $n \times m$ memristor array is divided into two smaller arrays of $n \times r$ and $r \times m$, respectively, the programming time of the NCS is reduced from $O(n \times m)$ to $O(n+m)$; Second, 1-D reduction significantly improves the programming robustness of an single MBC, which can be described below:

Fig. 4 depicts the programming voltage drop distribution on a 128×128 MBC, similar to Fig. 3(a). The only difference is that in Fig. 4, all memristors are at LRS, demonstrating the worst-case impact of IR-drop on MBC programming. As aforementioned in Section 3.1, during the programming of the MBC, the voltage reaching the target memristor may degrade from the original programming voltage when IR-drop is considered. And the degradation level relies on the location of the target memristor. In Fig. 4, we highlight (colored) the memristor locations with a voltage drop higher than $V_{bias}$ /2 when the voltages of the WL and BL connecting the memristor are set to $+V_{bias}$ and $GND$, respectively. We name the boundary of the highlighted area as the "hard-limit" of a single MBC scale. Any memristors outside the "hard-limit" will not be effectively programmed because they are practically "half-selected" (see Section 2.2). Increasing the programming voltage to raise the voltage applied on the memristors outside the "hard-limit", however, will affect the memristors that should be "half-selected". Hence, the scale of the "hard-limit" serves as a good measurement of MBC programming robustness. As shown in Fig. 4(a), the largest MBC size within the "hard-limit" is only 48×48 by assuming the sizes of the two dimensions of the MBC are the same, i.e., $n = m$. The maximum size of the data can be processed is only 48. If we can reduce the size of one dimension down to a smaller value, say, $r$ = 22, then the size of the another dimension can be extended to 128, as shown in Fig. 4(b). Such a MBC is sufficient to process the data with a size of 128 by leveraging our proposed 1-D reduction method, as long as the rank of $\boldsymbol{W}_{appx}$ is not higher than 22.

## 4.3 Two-dimensional (2-D) Reduction

1-D reduction can downscale the size of the needed MBC from $n \times m$ to $n \times r$ and $r \times m$, resulting in significant saving on the hardware design cost and better robustness. In some pattern classification tasks, we may further reduce the MBC size in both dimensions. For example, when classifying a noisy input pattern $\boldsymbol{x}_i$ (e.g., an $n \times 1$ vector), weighted network connections (a $n \times n$ matrix) are needed to associate a noisy input pattern to as one of the standard pattern $\boldsymbol{a}_q$ ($q$=1, 2, ⋯, $r$). Our proposed two-dimensional (2-D) reduction can further reduce the scale of the computing system by transforming the concerned neuromorphic algorithm to a distance comparison based classification as follows:

Without loss of generality, the similarity between the output vector $\boldsymbol{W} \cdot \boldsymbol{x}_i$ and the standard pattern vector $\boldsymbol{a}_q$ ($q$=1,2,⋯$r$) can be quantitatively measured by:

$$P_{iq} = (\boldsymbol{W} \cdot \boldsymbol{x}_i)' \cdot \boldsymbol{a}_q = (\boldsymbol{a}_1' \cdot x_i) \cdot (\boldsymbol{a}_1' \cdot \boldsymbol{a}_q) + (\boldsymbol{a}_2' \cdot \boldsymbol{x}_i) \cdot (\boldsymbol{a}_2' \cdot \boldsymbol{a}_q) + \cdots (\boldsymbol{a}_r' \cdot \boldsymbol{x}_i) \cdot (\boldsymbol{a}_r' \cdot \boldsymbol{a}_q). \quad (11)$$

Similar to Eq. (9), we can use $\boldsymbol{a}_q' \cdot \boldsymbol{x}_i$ ($q = 1, 2 \cdots r$) to form a new input vector $\widetilde{\boldsymbol{x}}_i$and calculate the similarity between $\widetilde{\boldsymbol{x}}_i$and other patterns as:

$$P_i = \begin{bmatrix} P_{i1} \\ \vdots \\ P_{ir} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_1' \cdot \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_r' \cdot \boldsymbol{a}_1 \\ \vdots & \ddots & \vdots \\ \boldsymbol{a}_1' \cdot \boldsymbol{a}_r & \cdots & \boldsymbol{a}_r' \cdot \boldsymbol{a}_r \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{a}_1' \cdot \boldsymbol{x}_i \\ \vdots \\ \boldsymbol{a}_r' \cdot \boldsymbol{x}_i \end{bmatrix} = \widehat{\boldsymbol{W}} \cdot \begin{bmatrix} \boldsymbol{a}_1' \cdot \boldsymbol{x}_i \\ \vdots \\ \boldsymbol{a}_r' \cdot \boldsymbol{x}_i \end{bmatrix}. \quad (12)$$

where $\widehat{\boldsymbol{W}}$ is the new weight matrix with a dimension of $r \times r$.

Eq. (12) implies that after the proposed 2-D reduction, the size of the needed MBC is no longer determined by the large dimension size of data pattern ($n$) but the number of the patterns needs to be trained ($r$). *This new property is of particularly importance to applications that process the data with large dimensions but only limited number of patterns to be concerned, e.g., identifying objects on high resolution image or video.*

## 4.4 Implementation Example

The proposed 1-D reduction scheme is applicable to any network models that contain the operation described in Eq. (1) while the 2-D reduction scheme can fit in some applications like Auto-Associate Memory (AAM) well. Here we use Hopfield-network as an example to illustrate the basic concept of hardware implementation of the two proposed system reduction schemes.

Conventional Hopfield-network uses recurrent data process architecture to implement associative memory by training the connecting synapse weights based on stored standard patterns [9]. Each of the neurons has an activation "sign" function, which determines whether this neuron fires an excitation or not. The input of each neuron is the summation of the activations from all the neurons of the synapse network during last iteration. Weight matrix $\boldsymbol{W}$ is trained with Hibbian rule as shown in Eq. (2).

Fig. 5 shows the conceptual schematic of our proposed system reduction schemes, including both 1-D and 2-D designs. For the purpose of demonstration, here we assume that the inputs of the NCS, i.e.,$\boldsymbol{x}_i$, are all binary information (0 or 1). Both 1-D and 2-D reduction schemes require the inputs to be preprocessed by multiplying with the concerned patterns. In normal implementation of Hopfield network, the outputs of the MBC are directly sent to comparators which conduct "*sign*" function. In our reduction schemes, a slightly more complex post-processing is performed at the outputs, which can be implemented with a traditional analog selecting circuit [13]. The analysis of system robustness and implementation area tradeoff will be presented in Section 6.6.3.

Compared to the 1-D reduced weight matrix$\widetilde{\boldsymbol{W}}$, the 2-D reduced weight matrix $\widehat{\boldsymbol{W}}$is more sensitive to memristor device variations as the variability of one memristor has relatively higher impact on the computation accuracy due to significantly reduced number of the memristors participating in the computation. More details onthe design tradeoffs between the two reduction schemes will be discussed in Section 6.

# 5. IR-DROP COMPEMSATION

In addition to reducing the dimension sizes of the MBC, we can also actively compensate the impact of IR-drop to further improve the computation reliability of the NCS. In this section, we propose an adaptive compensation method that can compensate the impact of IR-drop in both training and recall processes.

## 5.1 Recall Compensation

Based on Eq. (6), the weight matrix represented by a MBC with resistance state $\boldsymbol{R}$ is not $\boldsymbol{W} = 1/\boldsymbol{R}$ but $\boldsymbol{W}^* = g(\boldsymbol{R}, R_{\_wire})$ when IR-drop is considered. As summarized in Fig. 6, the IR-drop

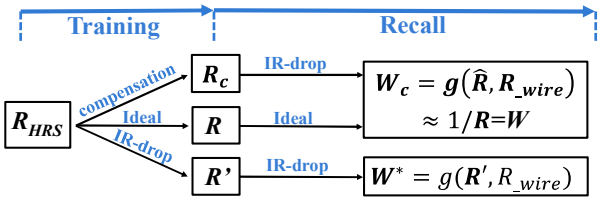


Fig. 6: Compensation for both training and recall process.

compensation can be performed by searching the new MBC resistance state $\boldsymbol{R}_c$ that generates a weight matrix $\boldsymbol{W}_c$ closest to the ideal target $\boldsymbol{W}$ as:

$$\min_{\boldsymbol{R}_c} \overbrace{\|\boldsymbol{W}-\boldsymbol{W}_c\|_F^2}^{F(\boldsymbol{R}_c)} = \sum_{i=1}^{n}\sum_{j=1}^{m}(\boldsymbol{W}_{(i,j)} - \boldsymbol{W}_{c(i,j)})^2. \quad (13)$$

Here, we define a cost function $F(\boldsymbol{R}_c)$ as the square of F-norm distance. $\boldsymbol{W}$ is a $n\times m$ matrix. This optimization problem can be solved by the gradient search method with $\boldsymbol{R}_c$ starting from $\boldsymbol{R}_c = \boldsymbol{R}$ as [18]:

$$\boldsymbol{R}_{c\_k+1} = \boldsymbol{R}_{c\_k} - \gamma\nabla F(\boldsymbol{R}_{c\_k})$$
$$= \boldsymbol{R}_{c\_k} + \gamma \cdot \sum_{i=1}^{n}\sum_{j=1}^{m}(2\cdot(\boldsymbol{W}_{(i,j)} - \boldsymbol{W}_{c(i,j)})\cdot\frac{\partial \boldsymbol{W}_{c(i,j)}}{\partial \boldsymbol{R}_{c\_k}}), \quad (14)$$

where $\gamma$ is the step length. The gradient direction relies on the relation between $\boldsymbol{W}_c$ and $\boldsymbol{R}_c$, i.e., $\boldsymbol{W}_c = g(\boldsymbol{R}_c, R_{\_wire})$. Here $g$ is a function that can be explicitly measured as follows: when we apply 1V on *i-th* WL of a MBC with resistance state of $\boldsymbol{R}_c$ and wire resistance of $R_{\_wire}$ and ground all other WLs and BLs, the magnitudes of the output current from BLs are equal to the elements in the *i-th* row of $\boldsymbol{W}_c$.

In general, the currents from every BL can be calculated by Modified Nodal Analysis as [21]:

$$\boldsymbol{Y}(\boldsymbol{R}_c, R_{\_wire})\cdot\begin{bmatrix}\boldsymbol{v}\\ \boldsymbol{k}\end{bmatrix} = \begin{bmatrix}\boldsymbol{i}\\ \boldsymbol{e}\end{bmatrix}. \quad (15)$$

$$\boldsymbol{Y}_{(i,j)} = \sum_{i=1}^{n}\sum_{j=1}^{m}(a_{(i,j)}/\boldsymbol{R}_{c(i,j)}) + b/R_{\_wire}. \quad (16)$$

Here $\boldsymbol{Y}$ denotes a conductance matrix that is a polynomial function of $\boldsymbol{R}_c$ and $R_{\_wire}$. $\boldsymbol{v}$ is the vector of total $2\times n\times m$ node voltages. $\boldsymbol{k}$ is the vector of $n+m$ WL/BL currents. $\boldsymbol{i}$ is the vector of current sources at each node, most of which are zeros except for the elements corresponding to WL/BL ports. $\boldsymbol{e}$ is the vector of $n+m$ voltage sources ($\boldsymbol{e}_{(i,1)}$=1V, other=0V). Then we have,

$$\boldsymbol{W}_{c(i,j)} = \boldsymbol{k}_{(n+j,1)}\ (j = 1\cdots m). \quad (17)$$

For the last term in Eq. (14), we have:

$$\frac{\partial \boldsymbol{W}_{c(i,j)}}{\partial \boldsymbol{R}_c} = \frac{\partial \boldsymbol{W}_{c(i,j)}}{\partial \boldsymbol{Y}(\boldsymbol{R}_c, R_{\_wire})}\cdot\frac{\partial \boldsymbol{Y}(\boldsymbol{R}_c, R_{\_wire})}{\partial \boldsymbol{R}_c} = \frac{\partial \boldsymbol{k}_{(n+j,1)}}{\partial \boldsymbol{Y}(\boldsymbol{R}_c, R_{\_wire})}\cdot\frac{\partial \boldsymbol{Y}(\boldsymbol{R}_c, R_{\_wire})}{\partial \boldsymbol{R}_c} \quad (18)$$

where $\partial \boldsymbol{Y}(\boldsymbol{R}_c, R_{\_wire})/\partial \boldsymbol{R}_c$ can be directly calculated based on Eq. (16). $\partial \boldsymbol{k}_{(n+j,1)}/\partial \boldsymbol{Y}(\boldsymbol{R}_c, R_{\_wire})$ is the sensitivity of current $\boldsymbol{k}$ to the conductance parameters $\boldsymbol{Y}$ in Eq. (15). This sensitivity can be solved by Adjoint Sensitivity Analysis (ASA) [21]. Fig. 7 demonstrated an example about how the imapct of IR-drop in a64×64 MBC is compensated. Simulation results show that the difference between $\boldsymbol{W}_c$ and $\boldsymbol{W}$ ($\|\boldsymbol{W}-\boldsymbol{W}_c\|_F^2$, as Y-axis) can be reduced down to below 1% only within 6 update steps described in Eq. (14).

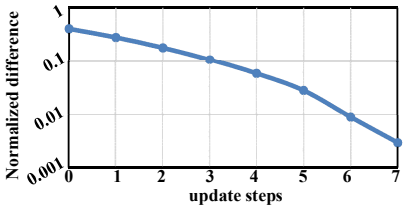


Fig. 7: Sensitivity analysis based compensation.

TABLE I. Experiment parameters setup

| Parameters | | Values |
|---|---|---|
| $R_{wire}$ for 2F×2F | | 2.5Ω |
| Amplifier area | | 5340 $F^2$ [19] |
| XOR gate area | | 280 $F^2$ |
| Memristor | High resistance state | 1 MΩ |
| | Low resistance state | 10 KΩ |
| | Cell area | 4 $F^2$ |
| Programming pulse amplitude($V_{bias}$) | | 2.9V |
| Reading voltage | | 1V |

## 5.2 Training Compensation

The objective of IR-drop compensation during MBC training is to minimize the difference between the trained resistance state $\boldsymbol{R'}$ of the MBC and the ideal resistance state $\boldsymbol{R}$ that represents target weight matrix$\boldsymbol{W}$. According to[7], IR-drop leads to minimum voltage degradation when all memristors are set to HRS. Thus, before training starts, all the memristors in the MBC should be initialized to HRS ($\boldsymbol{W}_{\boldsymbol{HRS}}$) to minimize the impact of IR-drops.

We define the ideal training time matrix $\boldsymbol{T}$ as the required programming pulse widths on the memristors and $\boldsymbol{V}$ as the ideal training voltage distribution applied on the memristors without considering IR-drop. $\boldsymbol{R}$ is the function of $\boldsymbol{T}, \boldsymbol{V},$ and $\boldsymbol{R}_{HRS}$ or $\boldsymbol{R} = f(\boldsymbol{T}, \boldsymbol{V}, \boldsymbol{R}_{HRS})$. Here $\boldsymbol{T}_{\boldsymbol{ij}}$ is the programming pulse width applied on the memristor connected by $WL_i$ and $BL_j$. $f$ is the memristor switching function that can be derived from Fig. 1(b). However, when IR-drop is considered, the training voltage distribution matrix is distorted to $\boldsymbol{V'}$. To compensate the voltage degradation in $\boldsymbol{V'}$, we can first calculate $\boldsymbol{V'}$ before programming each memristor and then derive a new training time matrix $\boldsymbol{T'}$ in order to obtain a trained MBC $\boldsymbol{R'}$ close to $\boldsymbol{R}$. For example, when programming voltage reduces from 2.9V to 2.7V (see Fig. 1 (b)), the required programming time needs to be extended from 500ns to 3μs to program the memristor to the same resistant state.

# 6. EXPERIMENTAL RESULTS

In this section, we will evaluate the effectiveness of the proposed schemes through a set of experiments: Section 6.1 shows the training quality improvement via IR-drop compensation; Section VI.B defines reading accuracy and discusses the selection of rank $r$ in system reduction; Section 6.3, 4 and 5 evaluate implementation area, performance and robustness of both system reduction methods, respectively. The trade-off between two methods will be particularly discussed in Section 6.5.3; Section 6.6 gives a case study of the applications of the proposed methods. TABLE I summarizes the parameters of the memristors and MBC designs used in our simulations.

## 6.1 Training Quality

In an MBC, the voltage applied to the two terminals of a memristor is affected by the device's location in the crossbar Fig. 8 shows the simulation results on the memristor resistance discrepancy between the target MBC and the actual trained MBC under the impacts of IR-drop and process variations. Similar to the training voltage degradation pattern shown in Fig. 3, the largest memristor resistance discrepancy of occurs at the far end of the MBC. Here we assume that the programmed memristor resistance follows the log-normal distribution as $r = r_0 \cdot \exp(\theta)$ [16]. $r_0$ stands for the mean of the programmed memristor resistance and $\theta \sim N(0, \sigma)$ is a random variable that follows Gaussian distribution.

We use the example from Section 4.4 to illustrate the design of NCS. The MBC scale can be reduced from 128×128 (original

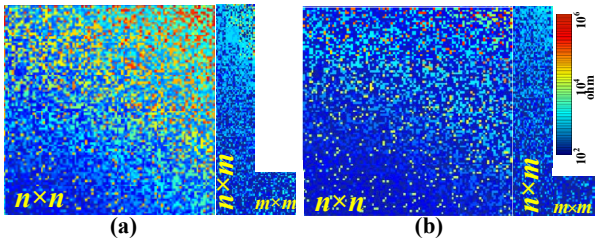


Fig. 8: Trained memristor resistance discrepancy (a) without IR-drop compensation. (b) with IR-drop compensation.

$n \times n$) down to 128×19 ($n \times r$) and 19×19 ($r \times r$) by applying 1-D and 2-D reduction schemes, respectively. Here $r$ is selected as 15%·$n$, which is the maximum pattern numbers that can be stored in a 128×128 Hopfield network in theory [17].As shown in Fig. 8(a), the memristor resistance discrepancy significantly reduces when MBC size decreases, implying a better training quality.

To further enhance NCS training quality, we introduce the IR-drop compensation technique given in Section 5 into training process. Fig. 8(b) shows that the compensation technique effectively minimizes the memristor resistance discrepancy. As weshall show in Section 6.5, the training quality enhancement can substantially improve recall successful rate of the NCS.

## 6.2 Reading Accuracy and Selection of r

In this experiment, reading of a MBC is defined as the case that all WLs of the MBC are connected to 1V while all BLs are grounded. In such a case, the ideal output current from the BLs should equal $\boldsymbol{W} \cdot \boldsymbol{x}$ ($\boldsymbol{x}$ is all one vector). However, due to IR-drop, the actual output current $\boldsymbol{I}$ will show deviation from $\boldsymbol{W} \cdot \boldsymbol{x}$, which can be described as reading accuracy issue. In this experiment, we evaluate the reading accuracy of the proposed reduction schemes under different conditions. We will also discuss the rank selection ($r$) of the 1-D reduced weight matrix based on the read accuracy. To achieve the maximum representation, the benchmarks adopted in the experiment include Hopfield network, BP training based weight connection and random weight matrix, all of which have a size of 100×100. We program the MBC to target weight matrix $\boldsymbol{W}$ for different benchmarks under impact of same memristor variation as section 6.1.

We first scan $r$ from 1 to 100 and see how the value of $r/n$ ($n$=100) affects reading discrepancy, i.e., $|\boldsymbol{I} - \boldsymbol{W} \cdot \boldsymbol{x}|/|\boldsymbol{W} \cdot \boldsymbol{x}|$.

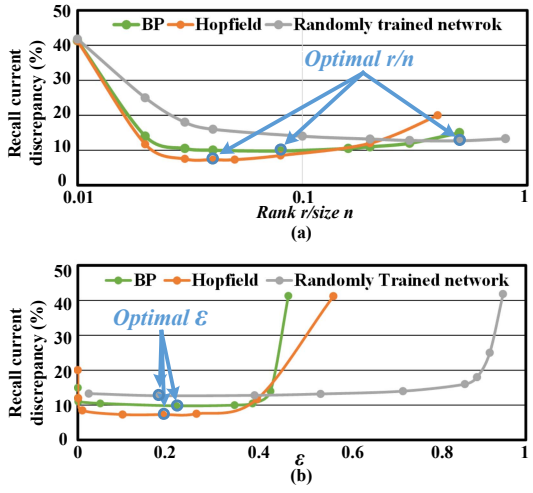


Fig. 9: Recall discrepancy comparison (a) discrepancy with respect to r/n, (b) discrepancy with respect to ε.

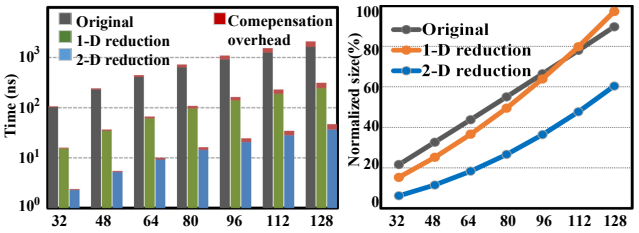


Fig. 10: Training time comparison. Fig. 11: Area cost comparison.

Based on the experiment results shown in Fig. 9(a), the optimal $r$ varies significantly in different benchmarks. For example, Hopfield network reaches the lowest read discrepancywhen $r/n$=*0.08~0.19*, while random weight matrix reaches it when $r/n$=*0.3~0.8*. So finding a generic proper range of $r/n$ for all benchmarks becomes impossible. The main reason is because each benchmark has different distributions of SVD singular value. The singular value sequence of the Hopfiled network used in our experiment, for example, is 53.7, 4.6, 3.8, 3.2…. while that of the random weight matrix is 5.03, 2.89, 2.77, 2.62….. Due to the highly skewed distribution of the singular values, a small $r$ (low rank) is sufficient for the Hopfield network. The random weight matrix, however, needs a large $r$ because of the relatively similar singular values.

We note that the threshold $\varepsilon$ introduced in Eq. (8) serves a generally good guidance for the rank selection. Fig. 9(b) shows the read accuracy degradation followed by the increase of $\varepsilon$. The optimal values of $\varepsilon$ for the three benchmarks all locate within the range of [0.1, 0.3]. Note that $\varepsilon$ = 0 means $r$ equals the rank of the original weight matrix $\boldsymbol{W}$. Continue increasing $r$ beyond the rank of $\boldsymbol{W}$ will not improve the read accuracy of the MBC but introducing extra IR-drop and noise. In the following experiments, we heuristically choose $r$ by setting $\varepsilon$ to 0.2.

## 6.3 Training Performance

Neither system reduction nor IR-drop compensation will affect the recall time of the NCS. However, training time can be affected by both techniques. Here we still use the example from Section 4.4. Fig. 10 compares the training times of the corresponding NCS designs with and without system reduction techniques and the training compensation time overheads. As memristors are programmed one by one in MBC, system reduction naturally shortens the training time by reducing the total memristor number. The overall training time of 2-D reduced design is only 3.3% of that of the original design. When MBC size rises, longer time is consumed on IR-drop compensation because of the severer voltage degradation. For instance, compensation overhead contributes to 4.12% of total training time when $n = m = 32$, and 28.3% when $n = m = 128$.

## 6.4 Area

As discussed in Section 4.3, system reduction scales down the size of MBC while introducing additional peripheral circuit. We evaluate the overall circuit area cost of original, 1-D and 2-D reduced NCS designs, as shown in Fig. 11. The circuit design details are also illustrated in Table I. 1-D reduced design always has a smaller area than original design until $n$ = 96, beyond which the overhead of extra circuit starts to dominate. 2-D reduced design, however, always has the smallest area: when $n$ = 128, the area of 2-D reduced design is only 61.3% of that of original design. Note that the areas cost shown in Fig. 11 is for only a single MBC and its peripheral circuit. When the NCS is scaled up to a level capable of

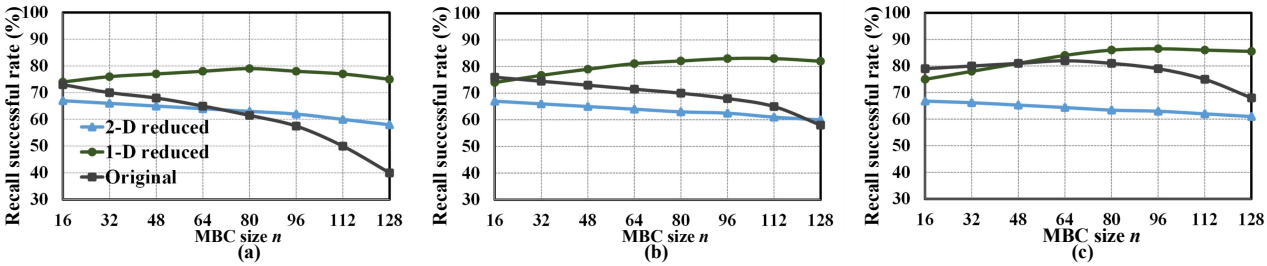


Fig. 12: Recall successful rates of three NCS designs considering IR-drop: (a) Training and Recall with IR-drop, (b) with training compensation, (c) with both training and recall compensation

processing large data size, e.g., high resolution image, multiple MBC may be needed. Routing and analog data transmission will occupy significant portion of the circuit area. A relevant case study will be presented in Section 6.6.

## 6.5 Robustness

Similar to Section 6.1, we set the number of standard patterns stored in a MBC to $0.15 \cdot n$ in our recall robustness analysis on the circuit implementation from Section 4.3. Each standard pattern is a randomly-generated binary vector. Test input patterns are the defected standard patterns where each digit has a 15% probability to be inversed. We determine whether a recall is successful by comparing the mismatch between the outputs of the test inputs and the corresponding standard patterns. Recall successful rate is obtained by running 1000 times Monte-Carlo simulations. Besides the process variations of memristor devices ($\sigma_m = 0.2$), we also assume each memristor has 0.1% chance to be stuck at HRS or LRS.

### *6.5.1 Training and recall with IR-drop*

We first evaluate the NCS performance under the impact of IR-drop. The same experiment setup and training method in Section 6.2 are adopted in these simulations. Fig. 12(a) shows the recall successful rate of three NCS designs when IR-drop is considered during both training and recall process. Conventional NCS design suffers from the largest degradation among all the designs when the scale of the MBC increases from 16×16 to 128×128. And 1-D reduction outperforms the other two designs.

Fig. 12(b) shows the results when IR-drop compensation is introduced during only training process. When the size of the MBC increases, the training quality of the original MBC is significantly improved by IR-drop compensation.

Fig. 12(c) shows the recall successful rate of all NCS designs when the IR-drop compensation is applied during both training and recall. IR-drop compensation substantially enhanced the robustness of original and 1-D reduced designs. As the MBC size of all 2-D reduced designs is smaller than20×20, neither the IR-drop effect nor the compensation improvement is significant. When $n = m = 128$, the 1-D reduced design shows a recall successful rate of 85.3%, which is 27.0% higher than original design (68.3%).

**TABLE II. Recall successful rate of NCS with different sizes**

| $\sigma_{wire}$ | $\sigma_m$ | Reduction scheme | Stored patterns/Sizes | | | |
|---|---|---|---|---|---|---|
| | | | 5/32 | 10/64 | 15/96 | 20/128 |
| 0 | 0.1 | Original | 0.934 | 0.902 | 0.835 | 0.716 |
| | | 1-D reduction | 0.917 | 0.933 | 0.958 | 0.944 |
| | | 2-D reduction | 0.884 | 0.875 | 0.844 | 0.833 |
| | 0.2 | Original | 0.803 | 0.822 | 0.797 | 0.683 |
| | | 1-D reduction | 0.788 | 0.844 | 0.868 | 0.853 |
| | | 2-D reduction | 0.663 | 0.644 | 0.628 | 0.607 |
| 0.05 | 0.1 | Original | 0.924 | 0.912 | 0.828 | 0.725 |
| | | 1-D reduction | 0.919 | 0.925 | 0.972 | 0.934 |
| | | 2-D reduction | 0.894 | 0.873 | 0.841 | 0.828 |
| | 0.2 | Original | 0.805 | 0.829 | 0.784 | 0.674 |
| | | 1-D reduction | 0.799 | 0.844 | 0.858 | 0.847 |
| | | 2-D reduction | 0.658 | 0.653 | 0.621 | 0.614 |

### *6.5.2 Impact of memristor/wire resistance variation*

In neural network model development, hardware device variations, e.g., memristor variation and metal wire resistance variation are generally not considered. In our IR-drop compensation design, wire resistance is also considered as a fixed value. In reality, these variations may harm the robustness of both training and recall process. Based on the experiment setup in section 6.5.1, we simulated the impacts of memristor and metal wire resistance variations and IR-drop. Table 2 shows the simulated recall successful rates of the NCS with different MBC sizes and memristor variation assumptions. When the memristors have lower variation (i.e., $\sigma_m = 0.1$), all three designs have better recall successful rates. However, the recall successful rate of the 2-D reduced MBC degrades more phenomenally than the other two designs when $\sigma$ increases, implying less tolerance to process variations as discussed in Section 4.4.

Although wire resistance greatly affects the impact of IR-drop, its variation $\sigma_{wire}$ does not show visible impact on overall system robustness due to its relatively small magnitude (2.5 ohm) and variance ($\sigma_{wire} = 0.05$).

### *6.5.3 Tradeoff between 1-D/2-D system reduction*

It is clear that among all the designs, 2-D reduced design has the best area efficiency even though it may not offer the same computation reliability as the 1-D reduced and conventional designs. However, computation reliability of 2-D reduced design shows higher sensitivity to the variation of memristor resistance ($\sigma_m$) than that of other two designs, as shown in Table 2. Hence, 2-D reduced design is a good solution for a large-scale data processing with well-controlled variability of memristor device as well as the relatively low computation accuracy per iteration.

1-D reduced design offers a good balance among area efficiency, computation accuracy, and tolerance to IR-drop and memristor variations. In fact, 1-D reduced design even shows better computation accuracy then the original design when the problem size is large because of the significantly improved tolerance to IR-drop and memristor variations, as shown in Fig. 12.

## 6.6 Case study

The advantages of system reduction become prominent when the size of the NCS is large. In our case study, a two-layer neural network for fingerprint recognition is demonstrated. Here, the network is trained with BP training method. A set of 256 fingerprint patterns $\boldsymbol{a}_i$ ($i$=1…256) with 64×64 pixels from SFinGe is used as training patterns [20].

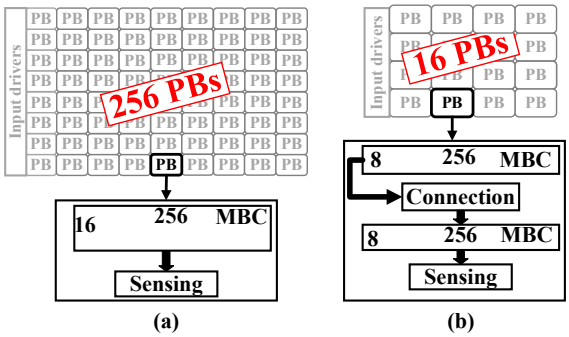


Fig. 13: Case study: (a) Conventional hardware design. (b) 1-D reduced hardware design.

After training, defects and noises are added on one randomly picked training pattern to generate a test pattern. Memristor device parameters and other experiment variables can be found in Table I. In the neural network, the input layer has 64×64=4096 neurons, and the output layer has 256 neurons, each of which indicates one of the training patterns. During the test, if the input pattern originates from pattern $\boldsymbol{a}_i$, the output of the $i^{th}$ output neuron will be "1" while that of any other neurons will be "0".

Fig. 13 (a) illustrates the hardware implementation of the network with conventional MBC designs. Since we have 256 output neurons, a process block (PB) contains a 16×256 MBC the size of which is constrained by the "hard-limit" described in Section 4.B. Each PB also has 16 input drivers and 1 output sensing amplifier that can be shared by all columns of the MBC. Total 256 PBs are needed to process the input data with a size of 4096 while the signals from all PBs will be summed in an extra PB.

Fig. 13(b) shows the designs with 1-D reduction scheme. The input and output sizes of each PB are all 256. Hence, each PB can be implemented with two reduced MBCs (256×8 and 8×256). Compared with the conventional design, eight amplifies are needed between the two MBCs in every PB. Since a PB with reduced MBCs can have an input data size of 256 without violating the "hard-limit", only 16 PBs are needed in this implementation. The significant decrease of total number of PBs is associated with the reduction in the number of sensing amplifiers, which consumes a large proportion of circuit area. Fig. 14 (b) compares areas of the two designs in Fig. 13(a) and (b). 1-D reduction scheme saves 81.4% of the circuit area. Moreover, the recognition successful rate of the 1-D reduced design is 85.3%, which is substantially improved from the one of the conventional design (71.7%). The adopted MBC parameter details are listed in Table I and $\sigma_m = 0.2$. Due to space limit, we did not include the case study on 2-D reduction scheme but the details will be similar to the discussions in Section 6.5.3.

# 7. CONCLUSION

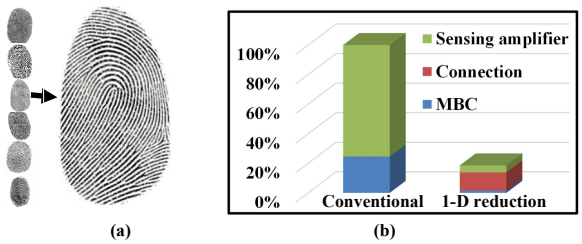


Fig. 14: Case study: (a) Finger prints pattern. (b) Area comparison.

MBC-based NCS is a promising solution to combat the memory bottleneck in Von Neumann architecture. Our analysis reveals that the IR-drop along the metal wires and memristor arrays in MBCs significantly affects the reliability and scalability of the NCS. In this work, we proposed system reduction schemes that can substantially reduce the size of the MBC required in NCS implementation for system robustness enhancement. We also proposed IR-drop compensation technique that can improve the training and recall reliability of the NCS. Simulations show that these techniques substantially improve the operation robustness of the NCS by 27.0% and reduce 38.7% of circuit area.

# 8. Acknowledgement

This work was supported in part by NSF CNS-1253424 (CAREER).

# 9. REFERENCE

[1] Asenov, A, *et al.*, "Intrinsic Parameter Fluctuations in Decananometer MOSFETs Introduced by Gate Line Edge Roughness", IEEE Transaction on Electron Devices, 2003.

[2] X. Dong, *et al.*, "Circuit and microarchitecture evaluation of 3D stacking magnetic RAM (MRAM) as a universal memory replacement", DAC, 2008.

[3] A. Sally, "Reflections on the Memory Wall," CCF, 2004.

[4] M. Sharad, *et al.*, "Ultra Low Power Associative Computing with Spin Neurons and Resistive Crossbar Memory", DAC, 2013.

[5] M. Hu, *et al.*, "Hardware realization of BSB recall function using memristor crossbar arrays", DAC, 2012.

[6] J. Liang, *et al.*, "Cross-Point Memory Array Without Cell Selectors—Device Characteristics and Data Storage Pattern Dependencies", IEEE Trans. on Electron Devices, 2010.

[7] L. Zhang, *et al.*, APL,2013.

[8] S. Yu, "Investigating the switching dynamics and multilevel capability of bipolar metal oxide resistive switching memory", APL, 2011.

[9] B. Liu, "Digital-assisted noise-eliminating training for memristor crossbar-based analog neuromorphic computing engine",DAC, 2013.

[10] L. Chua, "Memristor-the missing circuit element", IEEE Trans. On Circuit Theory, 1971.

[11] D. B. Strukov, *et al.*, "The Missing Memristor Found," Nature, 2008.

[12] K. Kim, *et al.*, "A Functional Hybrid Memristor Crossbar-Array/CMOS System for Data Storage and Neuromorphic Applications", Nano Letters, 2010.

[13] S. Vlassis, *et al.*, "Precision Multi-Input Current Comparator and Its Application to Analog Median Filter Implementation" AICSP,2003.

[14] N. Halko, *et al.*,"Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions",. Sep. 2009.

[15] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", Proceedings of the National Academy of Sciences of the USA, 1982.

[16] S. R. Lee, *et al.*,"Multi-level Switching of Triple-layered TaOx RRAM with Excellent Reliability for Storage Class Memory," Symposium on VLSI Technology, 2012.

[17] X. Y. Wang, *et al.*,"A Compact High-Accuracy Rail-to-Rail CMOS Operational Amplifier", ICBBE, 2010.

[18] Mordecai Avriel. "Nonlinear Programming: Analysis and Methods. Dover Publishing". ISBN 0-486-43227-0, 2003.

[19] X. Y. Wang, *et al.*,"A Compact High-Accuracy Rail-to-Rail CMOS Operational Amplifier", ICBBE, 2010.

[20] D. Maltoni, D. Maio, A.K. Jain and S. Prabhakar, Handbook of Fingerprint Recognition (Second Edition), Springer (London), 2009.

[21] Lawrence T. Pillage, *et al.* "Electronic circuit and system simulation methods", McGraw-Hill, 1995.

[22] MiaoHu, *et al*,"BSB training scheme implementation on memristor-based circuit," CISDA, 2013.