Received 29 February 2020; revised 1 April 2020, 9 April 2020, and 30 April 2020; accepted 30 April 2020.
Date of publication 4 May 2020; date of current version 6 July 2020.

Digital Object Identifier 10.1109/JXCDC.2020.2992306

A Relaxed Quantization Training Method for Hardware Limitations of Resistive Random Access Memory (ReRAM)-Based Computing-in-Memory

WEI-CHEN WEI[1], CHUAN-JIA JHANG[1], YI-REN CHEN[1], CHENG-XIN XUE[1], SYUAN-HAO SIE[1], JYE-LUEN LEE[1], HAO-WEN KUO[1], CHIH-CHENG LU[2], MENG-FAN CHANG[1] (Fellow, IEEE), and KEA-TIONG TANG[1] (Senior Member, IEEE)

[1]Department of Electrical Engineering, National Tsing Hua University (NTHU), Hsinchu 30013, Taiwan
[2]Information and Communication Labs, Industrial Technology Research Institute, Chutung 31030, Taiwan
CORRESPONDING AUTHOR: K.-T. TANG (kttang@ee.nthu.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Contract MOST 108-2218-E-007-021 and Contract MOST 108-2262-8-007-017.

ABSTRACT Nonvolatile computing-in-memory (nvCIM) exhibits high potential for neuromorphic computing involving massive parallel computations and for achieving high energy efficiency. nvCIM is especially suitable for deep neural networks, which are required to perform large amounts of matrix–vector multiplications. However, a comprehensive quantization algorithm has yet to be developed, which overcomes the hardware limitations of resistive random access memory (ReRAM)-based nvCIM, such as the number of I/O, word lines (WLs), and ADC outputs. In this article, we propose a quantization training method for compressing deep models. The method comprises three steps: input and weight quantization, ReRAM convolution (ReConv), and ADC quantization. ADC quantization optimizes the error sampling problem by using the Gumbel-softmax trick. Under a 4-bit ADC of nvCIM, the accuracy only decreases by 0.05% and 1.31% for the MNIST and CIFAR-10, respectively, compared with the corresponding accuracies obtained under an ideal ADC. The experimental results indicate that the proposed method is effective for compensating the hardware limitations of nvCIM macros.

INDEX TERMS Compression, computing-in-memory (CIM), deep learning, quantization, resistive random access memory (ReRAM).

I. INTRODUCTION

DEEP neural networks (DNNs) have highly flexible parametric properties, and these properties are being exploited to develop artificial intelligence (AI) applications in various domains ranging from cloud computing to edge computing. Transferring a DNN to an edge device remains challenging because of the high requirements for storage, computing, and power. To overcome this challenge, numerous high-throughput, low-power devices have been proposed in recent years to reduce the time complexity of matrix–vector multiplications [1], [2]. Moreover, an increasing number of studies have attempted to break the memory wall and reduce the transmission overhead by providing memory with computational ability [3]–[5]. Meanwhile, customized model compression algorithms are essential for the DNNs to be effectively deployed on AI edge devices.

Devices based on the Von Neumann architecture are required to transfer massive amounts of data across hierarchical memory layers and the system bus when inferencing models. As Moore's law ebbs, the energy efficiency of memory approaches saturation, and the power consumption of memory access is difficult to reduce [6]. To break the Von Neumann bottleneck, many studies have investigated nonvolatile computing-in-memory (nvCIM), such as resistive random access memory (ReRAM). nvCIM is expected to improve multiply-and-accumulate (MAC) operations through highly parallel computing, low standby power consumption, and reducing the data transfer time from memory to the

AI processor. However, nvCIM is associated with critical challenges. Due to the limitations of I/O, word lines (WLs), and ADC outputs, a tradeoff exists between throughput (TOPS/W) and data precision when performing MAC operations. Some works focus on how to split the weight matrix to reduce the ADC resolution requirement. Kim *et al.* [7] improved the ReLU function to reduce the ADC resolution requirement from 8 to 4 bit [7]; however, this article did not consider ADC variation [8] and used a very large crossbar (512 × 512), which may cause considerable accuracy degradation. Tang *et al.* [9] proposed to deploy a trained BNN network by splitting weights and mapping to multiple crossbar array with a 4-bit ADC [9]. Sun *et al.* [8] described how to optimize the accuracy and chip area when mapping the weights to different crossbar sizes and different ADC precisions; however, these works did not consider partial sum quantization during training, which may cause accuracy loss in silicon realization.

In this article, we explore the advanced techniques involved in the design of the structure of nvCIM macro. According to the analysis of nvCIM, we propose a quantization scheme that accounts for the hardware limitations of nvCIM. Our main contributions are as follows.

1) An overall analysis of the hardware limitations of ReRAM-based computing-in-memory (CIM) chip design.
2) A proposed quantization training algorithm that involves input/weight quantization, ReRAM convolution (ReConv), and ADC quantization.
3) To examine the effectiveness of various network architectures, the proposed method is applied to different benchmark data sets (MNIST and CIFAR-10).

The remainder of this article is organized as follows. Section II introduces the hardware limitations of the current nvCIM designs. Section III introduces an nvCIM-aware quantization method based on hardware limitations. Section IV presents the experimental results. The concluding remarks are presented in Section V.

II. ReRAM-BASED CIM

In recent years, many ReRAM-based CIM chips have been proposed, with attempts made to balance between accuracy and efficiency while achieving higher precision. A trend to increase the number of I/O and precision of weights has been observed for operations ranging from 1b-input, ternary-weight, 3b-output [3] and 1b-input, 8b-weight, 1b-output [4] to 2b-input, 3b-weight, 4b-output [5] (see Fig. 1). More bits can be used to represent a value for improving the precision and complexity of operation. The techniques for achieving these improvements involve the design of the entire nvCIM macro, physical characteristics of ReRAM, and precision of ADC outputs. Consequently, the accuracy of each chip is improved continuously, and more complicated inferencing work can be performed. However, increasing the numbers of I/O, WL, and ADC outputs brings new challenges and

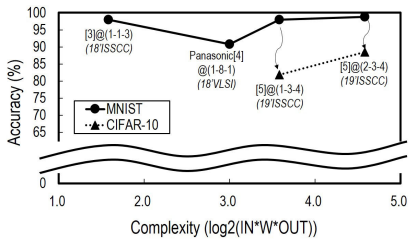


FIGURE 1. Hardware complexity and accuracy of nvCIM in recent works.

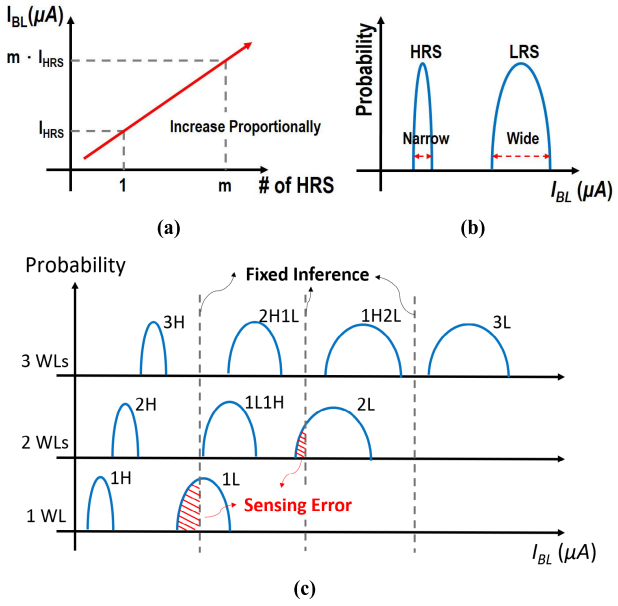


FIGURE 2. (a) I_{BL} increases proportionally with the # of ReRAM in HRS. (b) Narrow and wide distributions of I_{BL} in the HRS and LRS, respectively. (c) Fixed reference and different sensing results.

hardware limitations. In this section, these limitations are explained, highlighting the need for hardware and software codesign.

A. INPUT LIMITATIONS

To overcome the limitation of input precision on WLs, two critical factors must be considered: the input pattern variation and process variation of ReRAM. In the case of input pattern variation, the bitline current (I_{BL}) is highly influenced by the current (I_{HRS}) through the number of ReRAM cells in the high-resistance state (HRS) [see Fig. 2(a)]. In particular, when the R-ratio (R_H/R_L) is small, I_{HRS} is significant. Fig. 2(b) shows the different distributions of I_{BL} corresponding to HRS and low-resistance state (LRS) due to the process variation of ReRAM. Fig. 2(c) shows the different input patterns of multibit MAC values. Each row represents the number of turned-on WLs. H or L represents weight 0 or 1. If the reference is fixed for different turned-on WLs, different sensing results might be obtained, resulting in incorrect outputs. To solve this problem, Mochida *et al.* [4] proposed

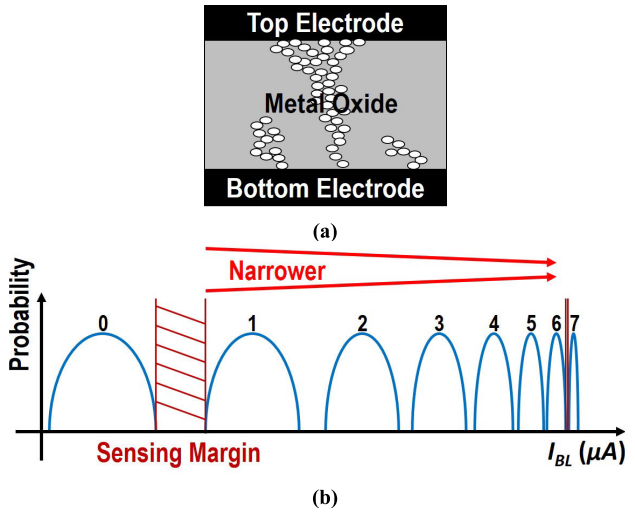


FIGURE 3. **(a) Conductive filament of ReRAM. (b) Narrower sensing margin due to input pattern and process variation.**

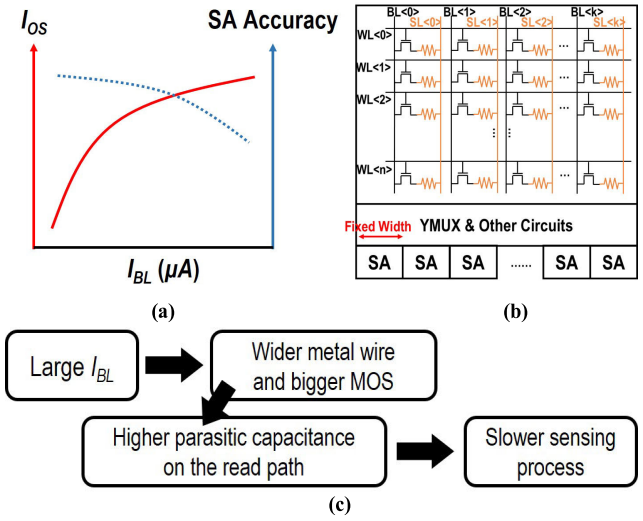


FIGURE 4. **(a) Relation between I_{BL}, I_{OS}, and SA accuracy. (b) Limited width for the SA schematic. (c) Influences of large I_{BL}.**

a dynamic I_{REF} generation scheme. However, this scheme requires a large area when the required precision of WL increases.

In the HRS, I_{BL} passing through the ReRAM cell is stable and has low variation. However, in the LRS, I_{BL} is widely distributed due to different widths and numbers of conductive filaments, which are generated during the SET stage in the middle layer of ReRAM [see Fig. 3(a)]. Considering the input pattern and process variation discussed earlier, achieving a higher precision of input would lead to very close distributions of I_{BL} when the MAC value is high [see Fig. 3(b)]. Therefore, the MAC precision of hardware is limited. The inputs precision should be optimized with software codesign to maintain inference accuracy.

B. WEIGHT LIMITATIONS

For the MAC operation on a macro, the higher number of bits used to represent a weight value, the higher is the inference accuracy. Two methods can be used to meet the required weight precision. One is to simply use multiple single-level-cell (SLC) ReRAM to represent one weight value. However, this increases the area cost and complexity of the MAC operation, which is related to the power and latency of the entire input process. The larger the size of the memory, the longer the metal wire. Moreover, the problem of IR drop and latency may occur. Therefore, simply increasing the number of ReRAM cells is not a practical way. The number of ReRAM cells should be optimized with software codesign.

The other method involves making a multilevel cell (MLC) ReRAM to represent the weight value with more than one bit. In [4], a ReRAM with eight levels that achieve 256 times higher precision of weight is proposed. However, due to the process variation in ReRAM, precisely controlling the resistance of ReRAM with 256 levels is difficult. The reported accuracy of MNIST with MLC ReRAM is 90.8%, less than the reported accuracy using SLC ReRAM [3], [5].

C. ADC OUTPUT PRECISION LIMITATIONS

Increasing the precision of ADC outputs involves many challenges related to the input offset, area, parasitic capacitance, and sense margin. First, with increasing I_{BL} due to multibit MAC operations, the input offset of ADC increases, which reduces the accuracy of the sensing output [see Fig. 4(a)]. Second, to meet the precision requirement, the number of ADC outputs must be increased. Consequently, the area of the ADC becomes too large [see Fig. 4(b)]. Moreover, when I_{BL} is large, a wide metal wire and large MOS are required, resulting in increased parasitic capacitance on the read path, thus decreasing the speed of the ADC sensing process [see Fig. 4(c)]. Last but not least, achieving a large sensing margin is always a circuit design challenge. With the process variation of ReRAM and the input pattern, ADC has a very small sensing margin for obtaining a correct output.

Due to the challenges mentioned earlier, the precision of ADC outputs is considerably limited. Moreover, the precision of ADC outputs limits the precision of the entire MAC operation. Therefore, how to do quantization in software become critical to compensate for the limited precision of ADC and to improve the inference accuracy for CIM.

III. nvCIM QUANTIZATION TRAINING

Considering the hardware limitation mentioned in Section II, the proposed nvCIM quantization comprises three steps: input and weight quantization, ReConv, and ADC quantization. The mathematical details of the proposed method are introduced in this section.

A. INPUT AND WEIGHT QUANTIZATION

The ReLU function is widely used as the activation function in DNNs. To satisfy the input limitation of nvCIM, the activation function is modified for quantization of the inputs as

Algorithm 1 ReRAM Convolution, ReConv

Input: At $l-$th layer, a minibatch of b_A-bit quantized inputs $\boldsymbol{A}^q_{l-1}$, b_W-bit quantized weights $\boldsymbol{W}^q_l$, ADC precision b_{AD}-bit, $\mathrm{N_{WL}}$, filter size f.

Output: Full-precision group convolution result $\boldsymbol{A}_l$.

$\mathrm{K} \leftarrow \frac{(\text{input channels of } \boldsymbol{W}^q_l, \boldsymbol{A}^q_{l-1}) \cdot f^2}{\mathrm{N_{WL}}}$

$\boldsymbol{W}^q_{l,k}, \boldsymbol{A}^q_{l-1,k} \leftarrow \mathrm{split}(\boldsymbol{W}^q_l, \mathrm{K}), \mathrm{split}(\boldsymbol{A}^q_{l-1}, \mathrm{K})$

for $k = 1$ **to** K in $\boldsymbol{W}^q_{l,k}, \boldsymbol{A}^q_{l-1,k}$ and set $\boldsymbol{A}_l = 0$ **do**

  $\boldsymbol{A}_{l,k} \leftarrow$ **Convolution**$(\boldsymbol{A}^q_{l-1,k}, \boldsymbol{W}^q_{l,k})$

  $\boldsymbol{A}^q_{l,k} \leftarrow \boldsymbol{Q}_{\mathbf{AD}}(\boldsymbol{A}_{l,k})$

  $\boldsymbol{A}_l \leftarrow \boldsymbol{A}_l + \boldsymbol{A}^q_{l,k}$

end for

follows:

$$Q_A(A_l) = \frac{\mathrm{round}(\min(\max(0, A_l), 1) \cdot (2^{b_A} - 1))}{(2^{b_A} - 1)} \tag{1}$$

where A_l denotes the inputs (or activations) of the lth layer in the DNN and b_A denotes the bit of input.

Due to the limitation of a ReRAM cell and model accuracy, the proposed weight quantization focuses on the design of SLC ReRAM. The SLC ReRAM structure usually uses different groups of BLs to represent integer weight values. Thus, the quantization level must follow asymmetric quantization. The weight quantization is defined as follows:

$$Q_{\mathrm{W}}(W_l) = \frac{\mathrm{round}(\tanh(\bar{W}_l) \cdot (2^{b_W - 1} - 1))}{(2^{b_W - 1} - 1)} \tag{2}$$

where W_l denotes the weight matrix of the lth layer in DNN. The normalized $\bar{W}_l = \tanh(W_l)/\max(\mathrm{abs}(\tanh(W_l)))$ and b_w denotes the weight bit.

B. ReRAM CONVOLUTION

Considering the features of the crossbar architecture, the strategy of weight reuse has considerable benefits for accelerating convolutional computation by operating nvCIM. However, because of the low resolution of ADC, the prediction accuracy would degrade. Therefore, the proposed quantization-aware training algorithm focuses on overcoming this drawback.

The proposed ReConv is summarized in Algorithm 1. ReConv simulates the behavior of ADC and the range of I_{BL}. Moreover, the behavior in ReConv is maintained the same as that in the original convolution. First, the inputs and weights are split into several groups. The number of split groups, K, is determined by the number of inputs and the number of WLs (NWL). Second, the grouped weights and inputs perform the convolution computation. The convolution results are then transferred to perform the proposed ADC quantization (see Section III-C). The final step of ReConv takes the quantized result into partial summation.

C. ADC QUANTIZATION

ADC quantization is included in ReConv to control the range of I_{BL} in the training algorithm. Two types of ADC quantization methods are proposed for the nvCIM quantization training algorithm. The first method, ReRAM quantization ($\mathrm{R^2Q}$), involves clip-based ADC quantization. The second method, ReRAM relaxed quantization ($\mathrm{R^3Q}$), involves concrete-based ADC quantization. $\mathrm{R^2Q}$ focuses on controlling the range of I_{BL}, while $\mathrm{R^3Q}$ optimizes the effect of noise through a probability model.

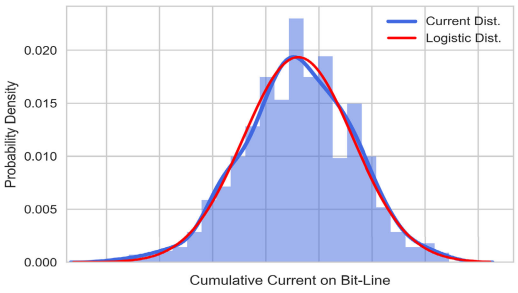


FIGURE 5. Distribution of I_{BL}. Blue bars: simulation data for different input/weight patterns and variations. Blue line: distribution of data with KDE. Red line: logistic distribution corresponding to the data.

1) CLIP-BASED ADC QUANTIZATION

The first ADC quantization method involves clipping I_{BL} within a suitable range and then performing quantization. The mathematical formula of clip-based ADC quantization is as follows:

$$Q_{\mathrm{AD}}(A_{l,k}) = \frac{\mathrm{round}(\bar{A}_{l,k} \cdot (2^{b_{\mathrm{AD}} - 1} - 1))}{(2^{b_{\mathrm{AD}} - 1} - 1)} \tag{3}$$

where b_{AD} denotes the ADC bit and $\bar{A}_{l,k}$ denotes the inputs clipped into a suitable range. In practice, we set $\bar{A}_{l,k} = \min(\max(A_{l,k}, -1), 1)$. Only the overload I_{BL} is considered in ADC quantization performed using the clip method; however, the sensing margin problem mentioned in Section II-C still needs to be solved.

2) CONCRETE-BASED ADC QUANTIZATION

In [10], a relaxed quantization (RQ) technique is proposed, in which random noises are introduced to disturb the quantization process and then replace the sampling process with the Gumbel-softmax trick [11], [12]. This method provides a noise-aware strategy; however, the ADC sampling noise needs to be established through probability distribution in the nvCIM system.

With regard to the process variation of ReRAM discussed in this article, Fig. 5 shows the measured I_{BL} distribution, which closely follows the logistic distribution. Therefore, the logistic distribution can be utilized for training. As mentioned in Section II, the sensing margins of ADC are very small, which leads to a sampling error between multiple distributions.

Unlike RQ [10], in this article, the sampling noise is applied to every quantized value r_i. The noise distribution is $p(\tilde{r}_i)$, where $\tilde{r}_i = r_i + \epsilon$ and $\epsilon \sim L(0, \sigma)$, and $p(\tilde{r}_i) = L(r_i, \sigma)$.

Algorithm 2 Concrete-Based ADC Quantization During Training

Input: Input a,, temperature λ, variation of noise σ.
Output: Quantized outputs a^q.
1: $r = [r_0, \ldots, r_{N-1}] = [-2^{b_{AD}}+1, \ldots, 0, \ldots, 2^{b_{AD}}-1]$
2: $\pi_i = \begin{cases} 1 - sigmoid\left(\frac{a-r_i}{\sigma_i}\right); & \text{if } a > r_i \\ sigmoid\left(\frac{a-r_i}{\sigma_i}\right); & \text{if } a \le r_i \end{cases}$
3: $s_i \sim \text{Concrete}(\pi_i, \lambda)$
4: $a^q = \sum_i s_i r_i$

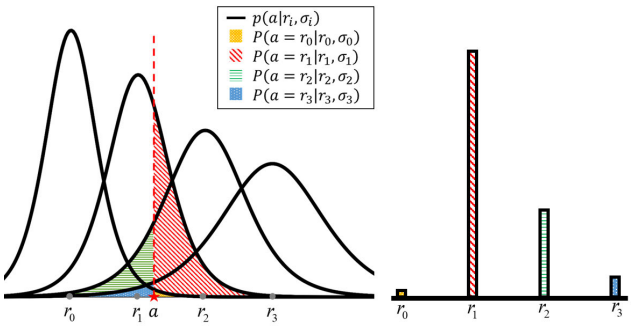


FIGURE 6. Categorical probability of the ADC quantization. (a) Shaded area corresponds to the tail probability of each representative distribution in which *a* falls. (b) Categorical probability over the quantized values. Each probability of the quantized values r_i is equal to the tail probability of *a* corresponding to $p(a = r|r, \sigma)$.

Algorithm 3 R^2Q Training With L-layers Network

Input: A minibatch of b_A-bit quantized inputs $\boldsymbol{A}^q$, current weights $\boldsymbol{W}$, weight precision b_W-bit, ADC precision b_{AD}-bits, number of word-line N_{WL}, learning rate η.
Output: update weights $\boldsymbol{W}^{updated}$.
1: **for** $l = 1$ **to** L **do**
2: $\quad \boldsymbol{W}_l^q \leftarrow \boldsymbol{Q_W}(\boldsymbol{W}_l)$
3: $\quad \boldsymbol{A}_l \leftarrow \textbf{ReConv}\left(\boldsymbol{A}_{l-1}^q, \boldsymbol{W}_l^q, b_{AD}, N_{WL}\right)$
4: $\quad \boldsymbol{A}_l^q \leftarrow \boldsymbol{Q_A}(\boldsymbol{A}_l)$
5: $\quad$ Optionally apply pooling
6: **end for**
7: $g_{A_L} \leftarrow \textbf{STE}(\frac{\partial C}{\partial A_L^q}, \boldsymbol{W}_l^q)$
8: $\boldsymbol{W}^{updated} \leftarrow \textbf{UpdateParameters}(\boldsymbol{W}, g_{A_L}, \eta)$

According to the aforementioned assumption, the second ADC quantization method in Algorithm 2 consists of two elements: the categorical probability and the Gumbel-softmax trick. The categorical probability is determined by the tail probability of each distribution corresponding to input data a, as shown in Fig. 6. The Gumbel-softmax trick (also known as a concrete distribution) replaces the discrete part in DNN with the concrete distribution. With the temperature parameter in the softmax function that approaches zero during training, the procedure makes the softmax function become the argmax function.

TABLE 1. Test error (%) with R^3Q for MNIST.

b_W-b_A-b_{AD}-N_{WL}	MNIST	Gap
32-32-32-ideal	0.57	–
8-8-32-ideal	0.65	-0.08
4-4-32-ideal	0.64	-0.07
2-2-32-ideal	0.65	-0.08
1-T-32-ideal	0.85	-0.28
2-2-32-ideal	0.65	–
2-2-8-ideal	0.73	-0.08
2-2-4-ideal	0.75	-0.1
2-2-3-ideal	0.79	-0.14
2-2-2-ideal	1.05	-0.4
2-2-4-36	0.76	-0.11
2-2-4-18	0.75	-0.1
2-2-4-9	0.7	-0.05

The proposed ReRAM-based nvCIM quantization training algorithm is summarized in Algorithm 3. By following the proposed approach, the trained weights of a DNN can meet the specifications of state-of-the-art nvCIM chips [3]–[5]. Moreover, the number of I/O, WLs, and ADCs can change arbitrarily before training to assist in future tape-out simulation.

IV. EXPERIMENTAL RESULTS

A. SOFTWARE EXPERIMENTAL SETUP

The proposed algorithm is based on the hardware specifications of [5]. The initial hyperparameters setting of concrete-based ADC quantization in R^3Q follows the instruction introduced in [10]. We initialized the parameter α in Q_{AD} according to the first batch of maximum and minimum values of the input A_l and divided α by b_{AD}^2. To represent the zero value of the quantized value, the bias β was not used. Moreover, we initialized λ of the Gumbel-softmax trick to 1.

We applied a variant of the LeNet-5 network with the 32C3-MP-64C3-MP-512FC-Softmax architecture to the MNIST data set. The learning rate was maintained constant at 1 over all 200 epochs. In the case of the CIFAR-10 data set, we considered a small VGG network [15] with the 2x(16C3)-MP-2x(32C3)-MP-2x(64C3)-MP-128FC-Softmax architecture and ResNet18 [5] and trained them with a mini-batch size of 100 and 400 epochs, respectively. The learning rate was divided by 10 after every 100 epochs. All the experiments were trained from scratch and implemented using Tensorflow [14]. LeNet-5 and small VGG were optimized using Adam [13] with a learning rate of 10^{-4}. ResNet18 was optimized using SGD with a learning rate of 10^{-1}.

B. EXPLORING THE QUANTITY SPACE OF WL AND ADC QUANTIZATION

The ideal resolution of a MAC operation is the sum of b_W and b_A. Moreover, a convolution operation consisting of many

TABLE 2. Test error (%) with R^{3}Q for CIFAR10.

b_W-b_A-b_{AD}-N_{WL}	sVGG	Gap	ResNet18	Gap
32-32-32-ideal	11.73	—	7.99	—
8-8-32-ideal	13.29	-1.56	8.67	-0.68
4-4-32-ideal	13.76	-2.03	9.31	-1.32
2-2-32-ideal	16.64	-4.91	10.38	-2.39
1-T-32-ideal	19.93	-8.20	18.83	-10.84
2-2-32-ideal	16.64	—	10.38	—
2-2-8-ideal	17.36	-0.72	11.36	-0.98
2-2-4-ideal	18.43	-1.79	12.53	-2.15
2-2-3-ideal	21.08	-4.44	20.28	-9.90
2-2-2-ideal	23.65	-7.01	29.16	-18.78
2-2-4-36	19.03	-2.39	12.44	-2.06
2-2-4-18	18.02	-1.38	18.85	-8.47
2-2-4-9	17.95	-1.31	16.63	-6.25

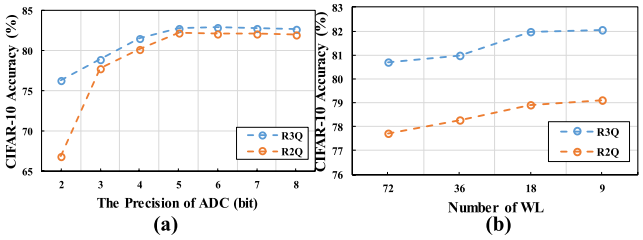


FIGURE 7. (a) Different precision of the ADC when N_{WL} is ideal. (b) Different N_{WL} of ReRAM convolution when ADC precision = 4 bits.

MAC operations requires a floating-point representation. The proposed ReConv can decrease the size of the set of MAC operations. However, the ideal resolution of ReConv must be reduced to satisfy hardware limitations. Thus, ADC quantization in the inference process must be simulated. In this section, we explore the reconfigurable nature of R^{3}Q. The testing results are reported in Tables 1 and 2, and the results can be divided into three parts.

In the first part, R^{3}Q exhibits flexibility for different bits of weight and activation. R^{3}Q supports weight/activation quantization in the range of binary/ternary bit to 8 bit. To meet the I/O limitations of the SOTA ReRAM chip such as [3], we explore the precision space of Q_{AD} by considering 2-bit weights and 2-bit activations. The results are reported in the second part of Tables 1 and 2. The benchmark error rates of LeNet, the small VGG network, and ResNet18 are 0.65%, 16.64%, and 10.38%, respectively. The network error rate only increases by 0.08% for the MNIST task and 0.72% (0.98%) for the CIFAR-10 task, with $b_{\mathrm{AD}} = 8$. The error rate for CIFAR-10 degrades to less than 2.5% at $b_{\mathrm{AD}} = 4$; however, the network performance deteriorates at extremely small ADC resolutions. Moreover, the condition $b_{\mathrm{AD}} = 4$ is suitable for the SOTA ReRAM CIM.

TABLE 3. Performance comparisons (test error %).

	b_W-b_A-b_{AD}-N_{WL}	MNIST (%)	
QRN[16]	2-32-32-ideal	1.25	—
R^{2}Q	2-32-32-ideal	1.13	+0.12
R^{2}Q	2-2-4-9	1.14	+0.11
R^{3}Q	**2-2-4-9**	**0.70**	**+0.55**
R^{2}Q	4-32-32-ideal	0.82	+0.43
R^{2}Q	4-4-4-9	0.87	+0.38
R^{3}Q	4-4-4-9	0.86	+0.39

TABLE 4. Generalization comparisons.

	Resolution of			N_{WL}	Training Complexity
	Weight	Input	ADC		
QRN[16]	✓	×	×	×	1x
R^{2}Q	✓	✓	✓	✓	1.44x
R^{3}Q	✓	✓	✓	✓	27.6x

The third part of Tables 1 and 2 indicates the relationship between N_{WL} and the error rate. The results suggest that as N_{WL} increases, the test accuracy decreases. Therefore, the design of nvCIM should consider the NWL with a precision-limited ADC. Moreover, the error rate can be reduced by quantizing the output values.

As shown in Table 1, the accuracy under the specification 2-2-4-9 is lower than that of ideal nvCIM (2-2-32-ideal) by only 0.05% for the MNIST. For the CIFAR10 in Table 2, small VGG and ResNet18 are lower than the ideal case by 1.31% and 6.25%, respectively. The results indicate that the proposed algorithm can effectively maintain the accuracy.

C. COMPARISON OF R^{2}Q AND R^{3}Q

The major difference between R^{2}Q and R^{3}Q is the quantization modeling for ADC. The performance of R^{3}Q is superior to that of R^{2}Q for every hardware limitation (see Fig. 7). In contrast to QRN [16], R^{3}Q supports not only weight quantization but also input quantization, ADC quantization, and I/O limitation. Thus, the proposed method outperforms QRN, as presented in the top half of Table 3.

To verify the general scalability of the proposed method, different precisions of input, weight, and ADC are shown in the lower half of Table 3. The proposed method still demonstrates very good performance and outperforms QRN. However, R^{3}Q requires a long training time, as shown in Table 4. The training time of R^{3}Q is 27.6 times greater than that of QRN [16] because the computation is slowed down by the concrete-based quantization.

V. CONCLUSION

This article thoroughly discusses the hardware limitations of ReRAM-based CIM. According to these limitations, a novel quantization training method is proposed for the ReRAM-based CIM system. The proposed method involves

three steps: input and weight quantization, ReConv, and ADC quantization. Moreover, two ADC quantization methods are introduced: R^2Q and R^3Q. Compared with previous training methods, the R^3Q method proposed in this article is more robust and comprehensive. The simulation results indicate that the R^3Q method can avoid the sampling errors occurring in ADC in nvCIM. The accuracy of R^3Q is lower than that of ideal nvCIM (2-2-32-ideal), whose I/O limitation and ADC are ideal, by only 0.05% and 1.31% for the MNIST and CIFAR-10 data sets, respectively, under the specification 2-2-4-9. The proposed training method can meet the specifications of state-of-the-art nvCIM chips; the number of I/O, WLs, and ADCs can also be changed arbitrarily before training to assist future tape-out simulation.

REFERENCES

[1] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[2] M. Price, J. Glass, and A. P. Chandrakasan, "A scalable speech recognizer with deep-neural-network acoustic models and voice-activated power gating," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 244–245.

[3] W.-H. Chen *et al.*, "A 65nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 494–495.

[4] R. Mochida *et al.*, "A 4M synapses integrated analog ReRAM based 66.5 TOPS/W neural-network processor with cell current controlled writing and flexible network architecture," in *Proc. IEEE Symp. VLSI Technol.*, Jun. 2018, pp. 175–176.

[5] C.-X. Xue *et al.*, "24.1 a 1Mb multibit ReRAM computing-in-memory macro with 14.6ns parallel MAC computing time for CNN based AI edge processors," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers* , Feb. 2019, pp. 388–389.

[6] X. Xu *et al.*, "Scaling for edge inference of deep neural networks," *Nature Electron.*, vol. 1, no. 4, pp. 216–222, Apr. 2018.

[7] Y. Kim, H. Kim, D. Ahn, and J.-J. Kim, "Input-splitting of large neural networks for power-efficient accelerator with resistive crossbar memory array," in *Proc. Int. Symp. Low Power Electron. Design*, Jul. 2018, pp. 1–6.

[8] X. Sun, S. Yin, X. Peng, R. Liu, J.-S. Seo, and S. Yu, "XNOR-RRAM: A scalable and parallel resistive synaptic architecture for binary neural networks," in *Proc. DATE*, Mar. 2018, pp. 1423–1428.

[9] T. Tang, L. Xia, B. Li, Y. Wang, and H. Yang, "Binary convolutional neural network on RRAM," in *Proc. 22nd Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2017, pp. 782–787.

[10] C. Louizos, M. Reisser, T. Blankevoort, E. Gavves, and M. Welling, "Relaxed quantization for discretized neural networks," 2018, *arXiv:1810.01875*. [Online]. Available: http://arxiv.org/abs/1810.01875

[11] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," in *Proc. ICLR*, 2017.

[12] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," in *Proc. ICLR*, 2017.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–41.

[14] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: https://arxiv.org/abs/1603.04467

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[16] Q. Yang, H. Li, and Q. Wu, "A quantized training method to enhance accuracy of ReRAM-based neuromorphic systems," in *Proc. ISCAS*, May 2018, pp. 1–5.

WEI-CHEN WEI received the B.S. degree in communication engineering from National Central University, Taoyuan, Taiwan, in 2017, and the M.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2019.

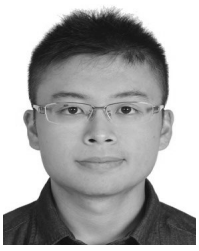
His current research interests include deep learning and model compression algorithm.

CHUAN-JIA JHANG received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2020, where he is currently pursuing the Ph.D. degree with the Institute of Electrical Engineering.

His current research interests focus on emerging nonvolatile memory circuit design and neuromorphic computing-in-memory hardware structure.

YI-REN CHEN received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2018, where he is currently pursuing the M.S. degree in electrical engineering.

His current research interests include deep learning and model compression algorithm.



CHENG-XIN XUE received the B.S. and M.S. degrees from the Department of Electrical Engineering, National Chung Cheng University, Chiayi, Taiwan, in 2015 and 2017. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan.

His current research interests include emerging nonvolatile memory, embedded memory, and deep neural network circuit design.

SYUAN-HAO SIE is currently pursuing the master's degree in electrical engineering with National Tsing Hua University, Hsinchu, Taiwan.

His current research interests include in-memory computing-based artificial intelligence accelerator and sparsity accelerator architecture.

JYE-LUEN LEE received the B.S. degree in industrial engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2018, where he is currently pursuing the M.S. degree in electrical engineering.

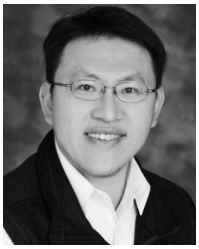
His current research interests include deep learning and model compression algorithm.

HAO-WEN KUO received the B.S. degree in communication engineering from National Central University, Taoyuan, Taiwan, in 2019. He is currently pursuing the M.S. degree in electrical engineering with National Tsing Hua University, Hsinchu, Taiwan.

His current research interests include deep learning and model compression algorithms, especially the integration of quantization and pruning algorithms.

CHIH-CHENG LU received the B.S. degree in electronics engineering from National Chung Kung University (NCKU), Tainan, Taiwan, in 1997, and the M.S. and Ph.D. degree in electrical engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 1997 and 2010, respectively.

From 2002 to 2005, he was with an IC design house, ELAN Microelectronics Corporation, Hsinchu, Taiwan, where he worked on the chipset design for optical storage products. From 2010 to 2012, he was with Taiwan Semiconductor Manufacturing Company (TSMC), Hsinchu, where he worked on digitally assisted sigma–delta converters and high-speed I/O for DDR III with 65- and 28-nm technology, respectively. In 2012, he joined the Industrial Technology Research Institute (ITRI), where he is currently the Project Manager of neuromorphic AI chip. His research interests include mixed-signal and analog circuit design, readout circuit and related applications for optical sensor, and probabilistic algorithm in VLSI.



MENG-FAN CHANG (Fellow, IEEE) received the M.S. degree from Pennsylvania State University, State College, PA, USA, in 1996, and the Ph.D. degree from the National Chiao Tung University, Hsinchu, Taiwan, in 2005. Currently, he is a Full Professor at National Tsing Hua University (NTHU), Taiwan.

Before 2006, he has worked in industry for over ten years. From 1996 to 1997, he designed memory compilers at Mentor Graphics, Bedminster, NJ, US. From 1997 to 2001, he designed embedded SRAMs and Flash at the Design Service Division (DSD), Taiwan Semiconductor Manufacturing Company, Hsinchu. From 2001 to 2006, he was the Co-Founder and the Director with IPLib Company, Hsinchu, where he developed embedded SRAM and ROM compilers, Flash macros, and flat-cell ROM products. His research interests include circuit designs for volatile and nonvolatile memory, ultralow-voltage systems, 3-D memory, circuit-device interactions, spintronics circuits, memristor logics for neuromorphic computing, and computing-in-memory for artificial intelligence.

Dr. Chang has been serving on technical program committee for ISSCC, IEDM as an Ex-com and the MT Chair, DAC, as the Sub-Com Chair, ISCAS as the Track Co-Chair, A-SSCC, and numerous international conferences. He was a recipient of several prestigious national-level awards in Taiwan, including the Outstanding Research Award of MOST-Taiwan, Outstanding Electrical Engineering Professor Award, the Academia Sinica Junior Research Investigators Award, and the Ta-You Wu Memorial Award. He has also been serving as the Program Director of the Micro-Electronics Program of Ministry of Science and Technology (MOST) in Taiwan during 2018–2020 and the Associate Executive Director for Taiwan's National Program of Intelligent Electronics (NPIE) and NPIE bridge program during 2011–2018. He has been serving as an Associate Editor for the IEEE Transactions on Very Large Scale Integration Systems and the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. He has also been serving as a Guest Editor for the IEEE Journal of Solid-State Circuits, the IEEE Transactions on Circuits and Systems—I: Regular Papers, the IEEE Transactions on Circuits and Systems—II: Express Briefs, and the IEEE Journal on Emerging and Selected Topics in Circuits and Systems. He has been a Distinguished Lecture (DL) speaker for the IEEE Solid-State Circuits Society (SSCS) and the Circuits and Systems Society (CASS), a technical committee member of CASS, and an Administrative Committee (AdCom) member of the IEEE Nanotechnology Council.

KEA-TIONG TANG (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the California Institute of Technology, Pasadena, CA, USA, in 2001.

His research interests include bioinspired learning chip, miniature electronic system, and biomedical implantable prosthetic device.

Dr. Tang is a TC Member of the IEEE Biomedical and Life Science Circuits Systems Technical Committee (BioCAS) and currently serving as the TC Chair.was a recipient of numerous awards, including the Outstanding Young Scholar Award, the Wu Ta-You Memorial Award, the National Innovation Award, and the Outstanding Electrical Engineering Professor Award. He was the Chair of the IEEE CAS Chapter, Taipei Section, from 2017 to 2018. He is also the Vice President of the IEEE Taipei Section. He is serving as the Board of Governor (BoG) of the CAS Society. He is also the Associate Editor-in-Chief of the IEEE Transactions on Biomedical Circuits and Systems (TBioCAS), an Associate Editor of the IEEE Sensors Journal, and a Guest Editor of the IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS).