Hybrid Computing Architecture Based on Analog Phase-Change Memory Chips for Deep Neural Network Training

Zhenhao Jiao*, Xiaogang Chen†, Tao Hong†, Weibang Dai†, Chengcai Tu‡,
Shunfen Li†, Houpeng Chen†, Zhitang Song†
*School of Microelectronics, University of Science and Technology of China, Hefei, China
†Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China
‡College of Physics, Donghua University, Shanghai, China
jiaozhenhao@mail.ustc.edu.cn, chenxg@mail.sim.ac.cn, hongtao@mail.sim.ac.cn, daiweibang@mail.sim.ac.cn,
tuchengcai@mail.dhu.edu.cn, lishunfen@mail.sim.ac.cn, chp6468@mail.sim.ac.cn, ztsong@mail.sim.ac.cn

***Abstract*—Deep neural networks (DNNs) have revolutionized fields like image recognition and natural language processing but face limitations with traditional von Neumann architectures due to high energy consumption and limited computing speed. We propose a hybrid architecture for DNN training combining a digital processing unit (DPU) and analog phase-change memory (PCM) chips using 40 nm CMOS technology. The DPU manages precise computations, while the PCM chip handles matrix-vector multiplication (MVM) with a novel nonlinear pulse scheme for accurate conductance tuning. Our architecture successfully trained a 3-layer fully connected neural network, achieving a classification accuracy of 97.26%, on par with software-based training. Simulations confirm the feasibility of extending this approach to more complex convolutional neural networks, demonstrating its adaptability to PCM device characteristics and potential for high-efficiency DNN training.**

***Index Terms*—phase-change memory, deep neural network, analog in-memory computing**

I. Introduction

Deep neural networks (DNNs) have achieved significant advancements in fields such as image recognition [1], [2], natural language processing [3], intelligent transportation [4], and finance [5]. However, the increasing complexity and scale of these models present substantial challenges for traditional von Neumann architectures, which suffer from high energy consumption and limited computing speed due to the separation of computation and memory units [6]. This bottleneck becomes particularly evident during the training of large-scale models like OpenAI's GPT-3, which requires immense computational resources and time [7].

Currently, most DNN training is conducted using Graphics Processing Units (GPUs) due to their high parallelism. Despite their computational power, GPUs are not ideal for edge computing owing to their high cost and energy consumption. Moreover, the von Neumann bottleneck persists, limiting memory access speed and bandwidth [8]. Analog In-Memory Computing (AIMC) emerges as a promising alternative, integrating computation and memory to overcome these limitations [9]. At the core of AIMC is the use of crossbar arrays of nonvolatile memories (NVMs) for matrix multiplication operations, leveraging Ohm's and Kirchhoff's laws to achieve constant time complexity for these operations. Phase-change memory (PCM), a type of NVM, is particularly suitable for storing neural network weights in an analog form, making it a key component in AIMC.

Despite its potential, AIMC with PCM faces significant challenges, primarily due to the nonideal characteristics of PCM devices, such as drift and variability, which hinder accurate weight mapping. Existing solutions have addressed these issues to some extent. Rasch et al. [10] proposed hardware-aware training methods that accommodate the nonideal characteristics of PCM, achieving inference accuracy comparable to floating-point operations. Joshi et al. [11] demonstrated that training methods could maintain accuracy even when transferring weights to PCM devices. Additionally, introducing noise during training has been shown to enhance inference accuracy [12], [13]. However, these approaches often focus on single factors, limiting their practical applicability.

To address these limitations, we propose a hybrid architecture combining a digital processing unit (DPU) with analog PCM chips using 40 nm CMOS technology. Our architecture leverages the DPU for precise computations and error gradient calculations, while the PCM chip performs matrix-vector multiplication (MVM) operations. A novel nonlinear pulses scheme is introduced to achieve approximate linear tuning of PCM conductance. This architecture not only adapts to the nonideal characteristics of PCM devices but also optimizes the conventional stochastic gradient descent algorithm for the tuning process. Our experiments demonstrate that a 3-layer fully connected neural network trained using this architecture achieves a classification accuracy of 97.26%, comparable to software-based training. Furthermore, simulations confirm the feasibility of extending this approach to more complex convolutional neural networks, highlighting its potential for high-efficiency DNN training.

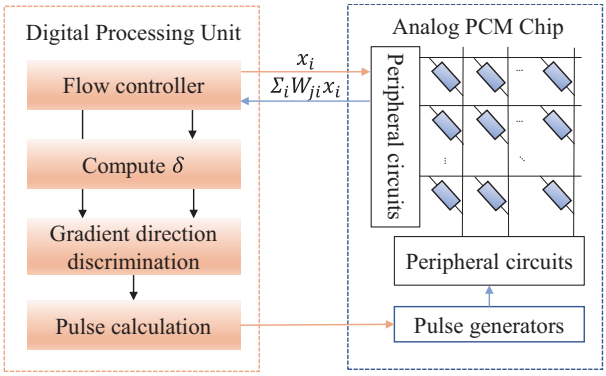


Fig. 1: Schematic diagram of the hybrid architecture for neural network training. The architecture is mainly composed of two parts: the DPU and the analog PCM chip.

II. Hybrid Architecture for Neural Network Training

In several current studies on PCM, digital-analog hybrid computing architectures have made greater research progress in the training as well as inference of deep neural networks [14], [15]. We propose a novel hybrid computing architecture for training deep neural networks using PCM, as shown in Fig. 1. Our proposed system architecture consists of two main components: a digital processing unit (DPU) and an analog PCM chip. The DPU is responsible for high-precision computation, especially for calculating the error gradient, determining the direction of weight update, and pulse calculation. In the training process, the main task of the DPU is to calculate the error δ using the back propagation algorithm. The analog PCM chip is responsible for storing the synaptic weights and performing the analog computations, such as weighted summation, with high energy efficiency. In the forward propagation process, the analog PCM chip calculates the sum of weights and inputs $\sum_i W_{ji} x_i$, where W_{ji} represents synaptic weights, and x_i represents input activations. This operation is carried out in analog mode, fully leveraging the inherent parallelism of the PCM array. A flow controller within the DPU manages data flow between the DPU and PCM chip, ensuring that the operation is synchronized. The gradient direction discrimination is obtained based on the result of the computed error δ and the input activation x. The main purpose is to enable the computed weight gradient to reduce the accuracy and better compensate for the randomness of the device. The pulse calculation is based on the programmed pulse combinations calculated from the obtained weights gradient steps for more accurate conductance tuning.

The mathematical framework and operation of this architecture is as follows.

A. Forward Propagation

The input activation x_i is fed into the analog PCM chip. The chip performs a weighted summation $y_j = \sum_i W_{ji} x_i$. The activation value x_i of the neuron is converted to a voltage V and applied to the rows of the crossbar array. According to the conductance value of each PCM cell, current flows in the column direction. The total current $I_j = \sum_i G_{ji} V_{x_i}$ corresponds to the weighted sum $y_j = \sum_i W_{ji} x_i$ and serves as the input for the next layer of neurons. This operation exploits the analog nature of the analog PCM chip to achieve high parallelism and energy efficiency.

B. Backward Propagation

This part is mainly operated in the DPU.

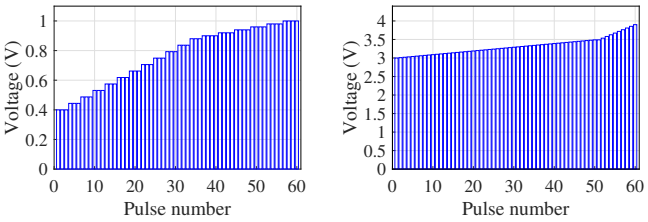
- Error calculation: DPU calculates the error δ for each layer using the output of the forward propagation and the target: $\delta^l = (y^l - \hat{y}) \cdot f'(z^l)$, where f' is the derivative of the activation function, y^l is the output of layer l, and $\hat{y}$ is the target.
- Gradient calculation: DPU calculates the gradient of the loss function with respect to the weights: $\nabla_{w^{(l)}} L = \delta^{(l)} \left(x^{(l-1)} \right)^{\mathrm{T}}$, where $\nabla_{w^{(l)}} L$ is the gradient value of the weights W of the loss function for layer l, $\delta^{(l)}$ is the error for layer l, $x^{(l-1)}$ is the output of layer $(l-1)$.
- Gradient direction discrimination: DPU will add sign discrimination operation to the calculated gradient. The main role is to take out the gradient direction and add constant compensation. The specific formula is as follows:
$$\nabla_{w^{(l)}} L = c \times sign\left(\nabla_{w^{(l)}} L\right) = \begin{cases} c, & \nabla_{w^{(l)}} L > 0 \\ 0, & \nabla_{w^{(l)}} L = 0 \\ -c, & \nabla_{w^{(l)}} L < 0 \end{cases}$$
where c is a constant compensation that we set according to the device characteristics of the analog PCM chip, and $sign(\cdot)$ is the sign discriminant function.
- Pulse Calculation: Set the combination of programmed pulses based on the actual pulse conductance relationship measured by the analog PCM chip.
- Weights update (conductance update): Based on the programmed pulse combinations obtained from the pulse calculation in the DPU, the target weights are set by the pulse generator and the peripheral circuits integrated in the PCM chip.

C. Advantages of Hybrid Computing Architecture

The architecture integrates a digital processing unit and an analog PCM chip to efficiently perform high-precision computations and low-energy weight updates. In the hybrid architecture, the analog PCM chip can utilize its nonvolatile nature to store a large number of network weight parameters with low power consumption and high computational density. The digital processing unit is used to compute the gradients in the backward propagation process, and these operations can be implemented more efficiently in digital circuits. The advantage of this hybrid architecture is that it can fully utilize the analog computational characteristics of PCM and the efficient computational power of digital circuits. Thus, high precision computation and low energy consumption weight updating of neural networks can be realized.

III. NonLinear Programmed Pulses Scheme

The analog PCM chip used in this experiment is based on 40 nm CMOS technology, its interface type is a parallel bus interface of SRAM-like. The chip supports real-time writing of data by address, and the written data will not be lost after power-down. The symmetric and linear tuning of the PCM conductance can be realized by using specific programmed pulses [16], [17]. Through experimental verification, we use the programmed pulses shown in Fig. 2 to realize the approximate linear tuning of the conductance. The programmed potentiation pulses consist of 60 nonlinear pulses, all with a pulse width of 3 μs, as shown in Fig. 2a. The voltage amplitude of the first 36 of these pulses is from 0.44 V to 0.88 V in 12 step stages. The voltage amplitude increases by 0.04 V in each step stage, and the same pulse is repeated 3 times in each stage. The voltage amplitude of the last 24 pulses is from 0.9 V to 1.0 V in 6 step stages. The voltage amplitude is increased by 0.02 V for each step stage and the same pulse is repeated 4 times for each step stage. The programmed depression pulses also consist of 60 nonlinear pulses with a pulse width of 75 ns, as shown in Fig. 2b. Of these 60 pulses, the amplitude of the first 50 pulses increases linearly, with the amplitude of each pulse increasing by 0.01 V. Starting at 3.0 V, it increases to 3.49 V. The last 10 pulses are nonlinear and the amplitude of the final pulse reaches 3.9 V.



(a) (b)

Fig. 2: Diagrams of programmed pulses scheme: (a) The diagram of programmed potentiation pulses; (b) The diagram of programmed depression pulses.

Based on the programmed pulses scheme, we use 120 programmed pulses as a complete conductance tuning cycle to simulate linear, symmetric tuning of conductance. The conductance variations of all PCM cells on the chip for a single read/write cycle are shown in Fig. 3. For a single tuning cycle, the conductance variation of the PCM chip ranges from 0 μS to 120 μS. And this range is also the range for network weights mapping. The simulation results indicate that PCM cells can display approximately linear, symmetric conductance response within a single tuning cycle (120 programmed pulses).

After resetting all the cells in the PCM chip, the programmed pulses scheme is used to perform 30 cycles of continuous conductance tuning experiments on all the cells in the chip, and the simulation results are shown in Fig. 4. It can be seen that the PCM chip has an approximately linear and symmetric conductance response with good consistency and repeatability. This approximate linear and symmetric conductance response is very important for us to train the neural network, especially for adjusting the weights during the training process.

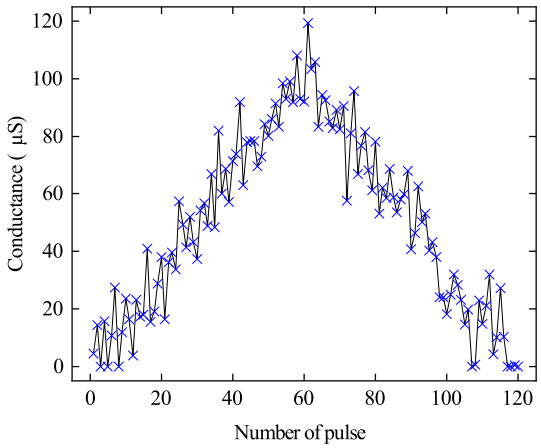


Fig. 3: Conductance response of PCM cells during one complete tuning cycle. There are 120 pulses in total, of which 60 are potentiation pulses and 60 are depression pulses.

IV. Implementation of Fully Connected Neural Network Training

In this experiment, we trained a 3-layer fully connected network using an analog PCM chip for experimental validation, and its network structure is schematically shown in Fig. 5a. This fully connected neural network has 784 input neurons and 10 output neurons, using sigmoid as the activation function. Except for the input layer, each layer has bias neurons, with the numbers of bias neurons being 256, 128, and 10, respectively. The bias neurons are not shown in the network architecture diagram. We also reviewed previous papers [11], [18], [19] and found that using a 2-PCM approach to implement weight mapping can effectively mitigate device noise and drift. Therefore, this experiment maps each weight as a differential PCM unit located on two columns, as shown in Fig. 5b. This fully-connected neural network has 235,146 neural weight synapses, which are mapped into 470,292 PCM devices. Each network weight corresponds to the conductance of 2 PCM devices, i.e., $W \propto [G_+ - G_-]$. A modified stochastic gradient descent (SGD) method is used for network training with a loss function that minimizes the mean square error function. The batch size for network training is set to 32, the learning rate is fixed at 0.1, and the training lasts for 50 epochs. The dataset used in the experiment is the MNIST handwritten digit dataset, which includes 60,000 training samples and 10,000 test samples, all of which are 28×28 grayscale images. It is worth noting that in order to facilitate our training process, the grayscale images need to be normalized before starting the training of the neural network. During the training process of the neural network, the adjustment of the PCM device conductance was performed using our previously proposed

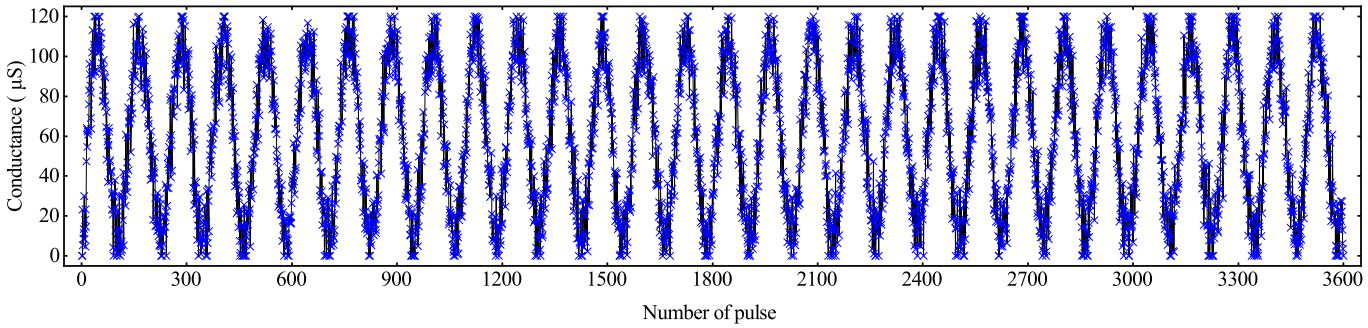
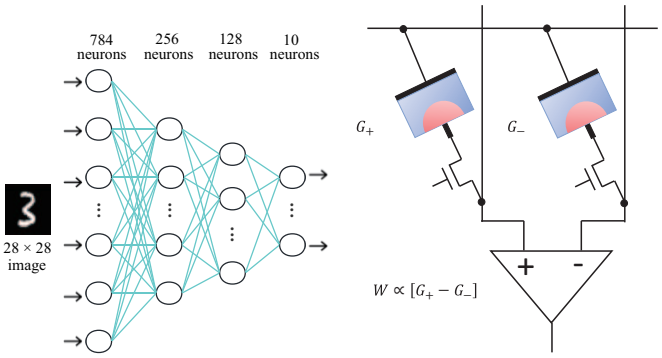


Fig. 4: Conductance response of the PCM device during 30 consecutive conductance tuning cycles of 120 pulses each.

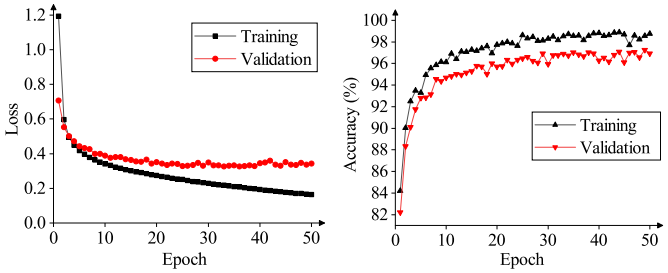
programmed pulses scheme. After 50 training epochs, the training loss of the fully-connected network is shown in Fig. 6a. The experimental results demonstrate that smooth convergence of the network can be achieved using the modified SGD training method and the programmed pulses scheme. The classification performance of this network is shown in Fig. 6b, with a maximum classification accuracy of 97.26% over 50 test epochs, which can basically reach the accuracy of software training.



(a) (b)

Fig. 5: Training experiment verification of the three-layer fully connected network with hybrid architecture: (a) Schematic diagram of network structure; (b) Schematic diagram of differential configuration of weights (2-PCM).

Experiments show that neural networks trained using analog phase-change memory chips can achieve classification accuracy comparable to software training. It is noteworthy that directly training neural networks on analog PCM chips can incorporate the nonideal characteristics of PCM into the training process, which is also a method to mitigate these nonideal characteristics. For the analog PCM chip used in this experiment, we simulated the drift behavior of PCM devices for specific weights, with the drift characteristics for specific weights shown in Fig. 7a. The drift behavior of PCM devices can cause changes in stored neural network weights, thereby affecting the accuracy of neural network [20], [21]. This is also an important factor affecting in-memory computing with PCM and similar analog devices. Using the training method proposed in this paper, after the neural network training



(a) (b)

Fig. 6: Experimental results for 3-layer fully connected networks: (a) Training loss of the network; (b) Classification accuracy of the network.

is completed, the classification accuracy of the network is evaluated within a certain time frame. The results are shown in Fig. 7b, from which it can be seen that the neural network inference accuracy only decreases by about 0.56% over a timeframe of more than one month. The experimental results clearly show that using an analog PCM chip to directly train a deep neural network mitigates the effects of device nonideal characteristics. This is mainly due to the direct incorporation of the nonideal characteristics of the device into the training process. In addition, the training of the network directly on the analog PCM chip does not require separate consideration of the nonideal characteristics of the device. The training process of the neural network will be adaptive to the nonideal characteristics of the device. This also greatly reduces the difficulty of deploying neural networks on nonvolatile memories.

V. Simulation of Convolutional Neural Network Training

Due to the limitation in the number of devices on analog PCM chips, it is currently not possible to directly deploy larger deep neural networks entirely onto analog PCM chips. To demonstrate the generality of using PCM analog arrays for training neural networks, we built an experimental simulation platform based on measured chip data to simulate larger neural networks. It is important to note that the focus of this simulation experiment is to model the weight update behavior of analog PCM chips during deep neural network training. The

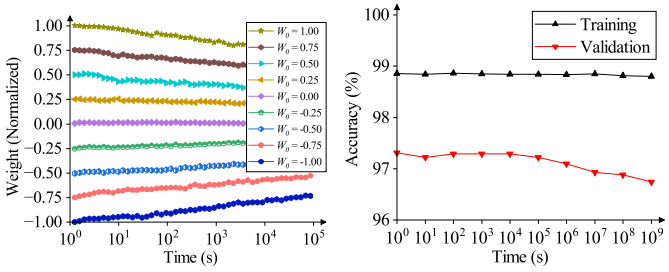


Fig. 7: Study of the drift characteristics of PCM devices: (a) Drift characteristics of normalized weights; (b) Network classification accuracy over a certain time scale.

main purpose is to demonstrate the feasibility of using analog PCM chips for training deep neural networks. Therefore, unlike the previous experimental setup, we simplified the calculations originally done in the digital processing unit to software computations. And the weight update behavior of the analog PCM chip was entirely based on our previous pulse-conductance data. The simulation process was implemented based on a look-up table of the analog PCM chip data. All pulse-conductance data is set up as a lookup table that is queried when a weight update is required.

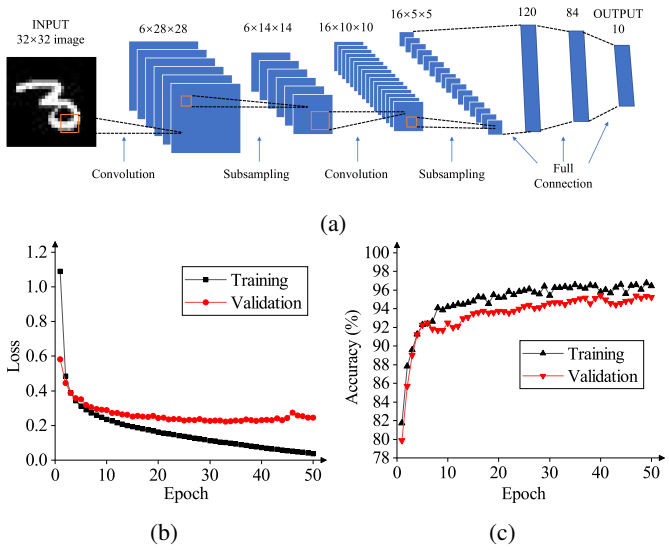


Fig. 8: Training simulation experiments with LeNet-5: (a) Neural network structure diagram of LeNet-5; (b) Training Simulation Losses for LeNet-5; (c) Classification accuracy of training simulation experiments for LeNet-5.

In this simulation experiment, we designed network structures for the commonly used MNIST handwritten digit dataset and CIFAR-10 dataset for image classification tasks. The network structure of LeNet-5 [22] is shown in Fig. 8a. It consists of 2 convolutional layers, 2 pooling layers, and 3 fully connected layers, which can classify handwritten digits. The training loss of this network is shown in Fig. 8b. This network can be directly trained using the analog PCM chip to achieve convergence. The classification accuracy of this network is shown in Fig. 8c. The highest classification accuracy on the training set is 96.76%, and the highest classification accuracy on the validation set is 95.44%. The network structure of VGG-16 [23] is shown in Fig. 9a. This network structure is relatively complex and has more parameters, making it suitable for the classification task of the CIFAR-10 dataset. The training loss of this network is shown in Fig. 9b. The successful convergence of this complex network confirms the feasibility of training deep neural networks with PCM devices. The classification accuracy of this network is shown in Fig. 9c, which is up to 98.26% for the training set and 94.98% for the validation set.

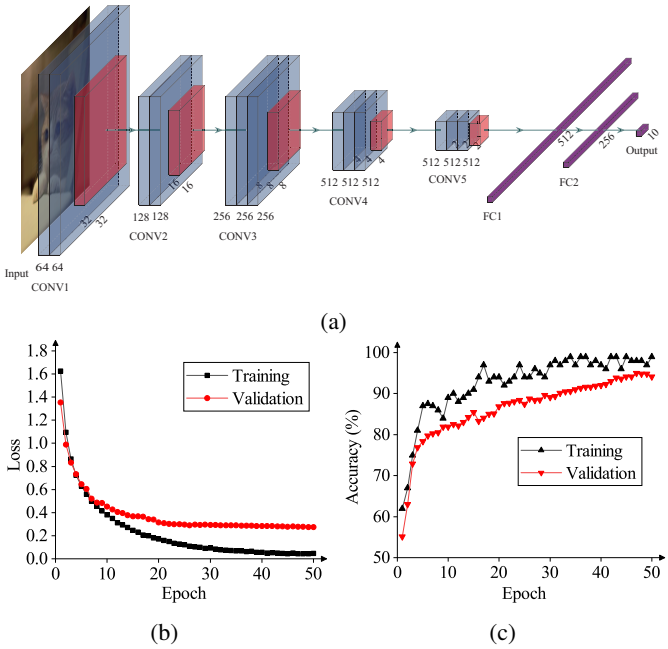


Fig. 9: Training simulation experiments with VGG-16: (a) Neural network structure diagram of VGG-16; (b) Training Simulation Losses for VGG-16; (c) Classification accuracy of training simulation experiments for VGG-16.

VI. Conclusion

In this paper, we proposed a novel deep neural network training architecture combining a digital processing unit with analog phase-change memory chips using 40 nm CMOS technology. Our innovative nonlinear programmed pulses scheme ensures precise adjustment of neural network weights, effectively mitigating the impact of PCM device nonideal characteristics. Experimental results show that our architecture successfully trained a three-layer fully connected neural network with a classification accuracy of 97.26%, comparable to software-based training, and simulations validated its feasibility for more complex convolutional neural networks. Future research directions include optimizing PCM design, developing new algorithms, and validating the architecture for larger network

structures, which collectively aim to address the von Neumann bottleneck and advance high-efficiency DNN training.

ACKNOWLEDGMENT

This research is supported by National Key Research and Development Program of China (grant number 2023YFB4502903) and Strategic Priority Research Program of the Chinese Academy of Sciences (grant number XDB44010200)

REFERENCES

[1] B. N. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, vol. 56, no. 2, 2024.

[2] G. Singh, P. Pidadi, and D. S. Malwad, "A review on applications of computer vision," *Hybrid Intelligent Systems: 22nd International Conference on Hybrid Intelligent Systems (HIS 2022), Lecture Notes in Networks and Systems*, vol. 647, pp. 464–479, 2023.

[3] S. H. Chung, S. Moon, J. Kim, J. Kim, S. Lim, and S. K. Chi, "Comparing natural language processing (nlp) applications in construction and computer science using preferred reporting items for systematic reviews (prisma)," *Automation in Construction*, vol. 154, p. 23, 2023.

[4] D. L. Wu, W. H. Yang, X. Y. Zou, W. Xia, S. Y. Li, Z. B. Hu, W. Z. Zhang, and B. X. Fang, "Smart-DNN plus: A memory-efficient neural networks compression framework for the model inference," *ACM Transactions on Architecture and Code Optimization*, vol. 20, no. 4, p. 24, 2023.

[5] S. Sakri, "Assessment of deep neural network and gradient boosting machines for credit risk prediction accuracy," *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 1–7, 2022.

[6] K. X. Sun, J. S. Chen, and X. B. Yan, "The future of memristors: Materials engineering and neural networks," *Advanced Functional Materials*, vol. 31, no. 8, 2021.

[7] B. D. Lund, T. Wang, N. R. Mannuru, B. Nie, S. Shimray, and Z. Wang, "Chatgpt and a new academic reality: Artificial intelligence-written research papers and the ethics of the large language models in scholarly publishing," *Journal of the Association for Information Science and Technology*, vol. 74, no. 5, pp. 570–581, 2023.

[8] A. Sebastian, M. L. Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnology*, vol. 15, no. 7, pp. 529–544, 2020.

[9] F. Garcia-Redondo, S. Das, and G. Rosendale, "Training DNN IoT applications for deployment on analog nvm crossbars," in *International Joint Conference on Neural Networks (IJCNN) held as part of the IEEE World Congress on Computational Intelligence (IEEE WCCI)*, ser. IEEE International Joint Conference on Neural Networks (IJCNN), 2020.

[10] M. J. Rasch, C. Mackin, M. L. Gallo, A. Chen, A. Fasoli, F. Odermatt, N. Li, S. R. Nandakumar, P. Narayanan, H. Y. Tsai, G. W. Burr, A. Sebastian, and V. Narayanan, "Hardware-aware training for large-scale and diverse deep learning inference workloads using in-memory computing-based accelerators," *Nature Communications*, vol. 14, no. 1, 2023.

[11] V. Joshi, M. L. Gallo, S. Haefeli, I. Boybat, S. R. Nandakumar, C. Piveteau, M. Dazzi, B. Rajendran, A. Sebastian, and E. Eleftheriou, "Accurate deep neural network inference using computational phase-change memory," *Nature Communications*, vol. 11, no. 1, 2020.

[12] S. Kariyappa, H. Tsai, K. Spoon, S. Ambrogio, P. Narayanan, C. Mackin, A. Chen, M. Qureshi, and G. W. Burr, "Noise-resilient DNN: Tolerating noise in PCM-based AI accelerators via noise-aware training," *IEEE Transactions on Electron Devices*, vol. 68, no. 9, pp. 4356–4362, 2021.

[13] X. Yang, C. Wu, M. Li, and Y. Chen, "Tolerating noise effects in processing-in-memory systems for neural networks: A hardware–software codesign perspective," *Advanced Intelligent Systems*, vol. 4, no. 8, p. 2200029, 2022.

[14] S. R. Nandakumar, M. L. Gallo, C. Piveteau, V. Joshi, G. Mariani, and I. B. et al., "Mixed-precision deep learning based on computational memory," *Frontiers in Neuroscience*, vol. 14, 2020.

[15] M. L. Gallo, R. Khaddam-Aljameh, M. Stanisavljevic, A. Vasilopoulos, B. Kersting, and M. D. et al., "A 64-core mixed-signal in-memory compute chip based on phase-change memory for deep neural network inference," *Nature Electronics*, vol. 6, no. 9, pp. 680–688, 2023.

[16] C. Li, D. Belkin, Y. N. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. R. Wang, W. H. Song, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. F. Xia, "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nature Communications*, vol. 9, 2018.

[17] Q. Wang, G. Niu, R. B. Wang, R. Luo, Z. G. Ye, J. S. Bi, X. Li, Z. T. Song, W. Ren, and S. N. Song, "Reliable $Ge_2Sb_2Te_5$ based phase-change electronic synapses using carbon doping and programmed pulses," *Journal of Materiomics*, vol. 8, no. 2, pp. 382–391, 2022.

[18] Q. W. Wang, Y. Park, and W. D. Lu, "Device variation effects on neural network inference accuracy in analog in-memory computing systems," *Advanced Intelligent Systems*, vol. 4, no. 8, 2022.

[19] N. Li, C. Mackin, A. Chen, K. Brew, T. Philip, A. Simon, I. Saraf, J. P. Han, S. G. Sarwat, G. W. Burr, M. Rasch, A. Sebastian, V. Narayanan, and N. Saulnier, "Optimization of projected phase change memory for analog in-memory computing inference," *Advanced Electronic Materials*, vol. 9, no. 6, 2023.

[20] M. J. Rasch, D. Moreda, T. Gokmen, M. L. Gallo, F. Carta, C. Goldberg, K. E. Maghraoui, A. Sebastian, and V. Narayanan, "A flexible and fast pytorch toolkit for simulating training and inference on analog crossbar arrays," in *IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2021.

[21] M. L. Gallo, C. Lammie, J. Büchel, F. Carta, O. Fagbohungbe, C. Mackin, H. Tsai, V. Narayanan, A. Sebastian, K. E. Maghraoui, and M. J. Rasch, "Using the ibm analog in-memory hardware acceleration kit for neural network training and inference," *APL Machine Learning*, vol. 1, no. 4, p. 041102, 2023.

[22] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.