Neural Network Acceleration and Voice Recognition with a Flash-based In-Memory Computing SoC

Liang Zhao[1*,3], Shifan Gao[1], Shengbo Zhang[2], Xiang Qiu[2], Fan Yang[1], Jie Li[3], Zezhi Chen[3], Yi Zhao[1*,4]
[1]College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China
[2]Flash Billion Semiconductor Co. Ltd., Shanghai, China [3]Hefei Reliance Memory Ltd., Hefei, China
[4]Nanhu Academy of Electronics and Information Technology, Jiaxing, China
*Email: {lzhao2020, yizhao}@zju.edu.cn

2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS) | 978-1-6654-1913-0/20/$31.00 ©2021 IEEE | DOI: 10.1109/AICAS51828.2021.9458476

***Abstract*—AI inference based on novel compute-in-memory devices has shown clear advantages in terms of power, speed and storage density, making it a promising candidate for IoT and edge computing applications. In this work, we demonstrate a fully integrated system-on-chip (SoC) design with embedded Flash memories as the neural network accelerator. A series of techniques from device, design and system perspectives are combined to enable efficient AI inference for resource-constrained voice recognition. 7-bit/cell storage capability and self-adaptive write of novel Flash memories are leveraged to achieve state-of-the-art overall performance. Also, model deployment techniques based on transfer learning are explored to significantly improve the accuracy loss during weight data deployment. Integrated in a compact form factor, the whole voice recognition system can achieve >10 TOPS/W energy efficiency and ~95% accuracy for real-time keyword spotting applications.**

I. Introduction

Recently, artificial intelligence of things (AIoT) has emerged as a major development trend in the technology and business worlds. By moving computing power closer to the edge terminal and augmenting it with sensors and AI algorithms, AIoT aims to deliver cognitive and learning systems on a massive scale. If it prevails, AIoT can offer significantly better quality of service and real-time decision making for applications ranging from personal wearable devices to smart home, smart city and industry automation. Accompanied by this trend, new technological challenges are posed to the circuits and system designers today, including but are not limited to:

- The constrained communication bandwidth even with the adoption of 5G.
- The insufficient supply of power to meet the demands of all IoT devices in the future.
- The explosive growth of the amount of data to be collected and processed through AIoT.

To tackle these challenges, conventional hardware design methodologies sometimes fell short while new computing paradigms and methodologies (e.g. in-memory computing) are gaining more and more traction. For example, computing systems based on the traditional von-Neumann architecture have separated computing and memory components. Thus, they typically underperform in data-intensive tasks such as computing deep neural networks (DNN), due to the need to frequently transfer data between processor and memory. The limited memory bandwidth created the so-called "memory wall" problem and increasingly became the performance bottleneck of such systems [1]. In contrast, in-memory computing methodology merges the computing and memory functions into the same Compute-In-Memory (CIM) block and consequently overcomes the memory wall and offers some great advantages in terms of energy/cost efficiency and computing speed [2]. For AIoT in particular, CIM devices of non-volatile nature are preferred due to the elimination of static power and the instant-on feature when activated from the sleep mode [3].

In this work, we shall demonstrate the design and implementation of a voice-recognition SoC with novel embedded Flash memories acting as the neural network accelerator. A series of techniques from technology, circuit-design and system perspectives are combined to enable efficient voice recognition under constrained resources. **The major contributions are as follows:**

- State-of-the-art 7-bit/cell capability and micro-second adaptive write of novel Flash devices are leveraged to maximize the storage density of weight data and cost efficiency of product testing.
- Model deployment techniques based on transfer learning concepts are explored to significantly improve the accuracy loss after the deployment of weight data; in particular, a residual model approach is applied for model fine-tuning after trial deployment.
- The in-memory computing SoC chip is integrated in a compact form factor and ships as a multi-keyword spotting product for smart home; the integrated voice-recognition system can achieve >10 TOPS/W energy efficiency and 94.8%+ accuracy in real time processing.

II. Flash-based In-memory Computing SoC Design

A. Flash-based computing paradigm for DNN acceleration

Flash memory, based on floating-gate transistors, is well known for its multi-level storage capability. Utilizing bi-directional F-N tunneling, a novel split-gate Flash memory cell has been proposed which enables very gradual program and erase operations (Fig. 1(a-c)), while maintaining good retention and endurance (more details can be found in [4]). This innovative device structure, also named programmable linear random-access memory (PLRAM), is an ideal candidate for synaptic devices in neural network accelerators. With a PLRAM array, highly efficient in-memory computing of matrix-vector multiply (MVM) can be achieved with the help of parallel input/output configuration and in-situ storage of weights. Using adaptive program/erase conditions, 7-bit/cell precision on MB size array (Fig. 1(d)) can be achieved with affordable program time (< 10 microseconds per cell).

This work was supported by the National Key Research and Development Program of China (No.2020AAA0109000) and the Key R&D Program of Zhejiang Province (No.2021C01039).

978-1-6654-1913-0/21/$31.00 ©2021 IEEE
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on October 10,2023 at 10:29:16 UTC from IEEE Xplore. Restrictions apply.

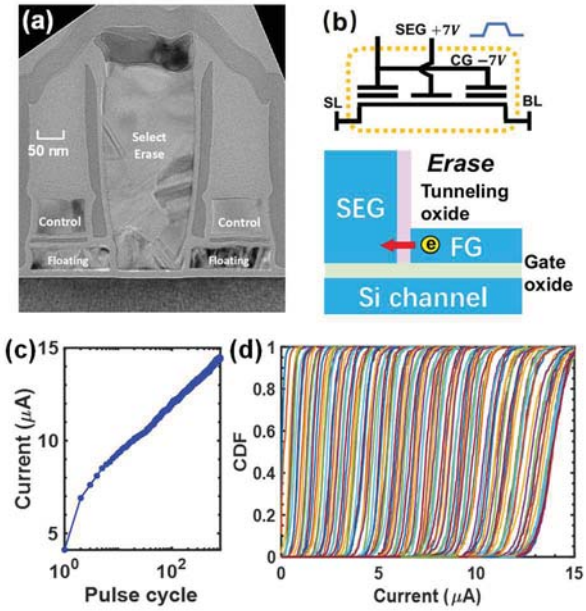


Fig. 1 (a) TEM cross-section image of a novel PLRAM device [4]; **(b)** Operation scheme for the gradual erase; **(c)** Single cell current response of PLRAM to constant-voltage erase pulses; **(d)** Programming results of an 876x128 sector, demonstrating 7-bit/cell capability.

B. *System architecture*

Fig. 2(a) shows the system architecture of the Flash-based in-memory computing SoC, which consists of a PLRAM compute-in-memory array, analog input (including DAC), analog output (including ADC), control logic and interface I/O blocks. The input signals such as voice or video streams are passed through the interface and processed by the analog input block before feeding into the CIM array. The MVM calculation results are detected and converted by the analog output block before output through the interface.

The full SoC design is implemented on a 90-nm CMOS technology platform offering embedded PLRAM solution (Fig. 2(b)). It features a memory array size of 2048x3096, totaling 6M novel Flash memory cells. Assuming 7 bit/cell capability, the total weight storage capacity is 42Mbit per die, ideal for many neural network models for voice recognition.

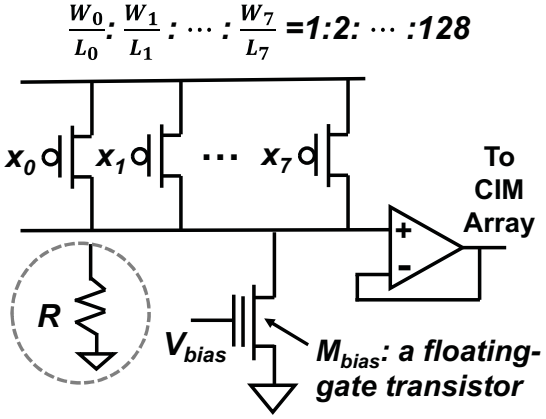


Fig. 3 Circuit diagram of the improved analog input block, including DAC based on pull-up transistors and a trimmable floating-gate transistor to generate the output voltage. (In the dotted circle: initial design in which the output voltage was generated by a resistor.)

The detailed designs of the analog input and output blocks are discussed in the following section.

C. *Analog input and output design*

Fig. 3 demonstrates the analog input block design with input buffer and 8-bit DAC functionality. Here the digital inputs are separated by bits and applied to the gates of 8 pull-up transistors in parallel, each with varying W/L. The pull-up current determined by the inputs can be sensed with a resistor (as shown in the dotted circle), and then applied to the input line buffers of the CIM array. However, since each input in parallel requires one DAC/buffer, there could be significant mismatch across the many input lines, causing degradation of CIM's accuracy. In order to offset the input mismatch, an improved design of the analog input block is shown in Fig. 3, which uses a floating-gate transistor to replace the resistor and generate the input voltage. This way, the resistances of the pull-down transistors (Mbias) can be trimmed by adjusting the floating-gate transistor's V_{th} to eliminate mismatch.

Fig. 4 demonstrates the design of the analog output block, featuring a cascode current mirror to generate output voltages and a differential ADC for output sensing. The cascode current mirror can precisely duplicate the current accumulated

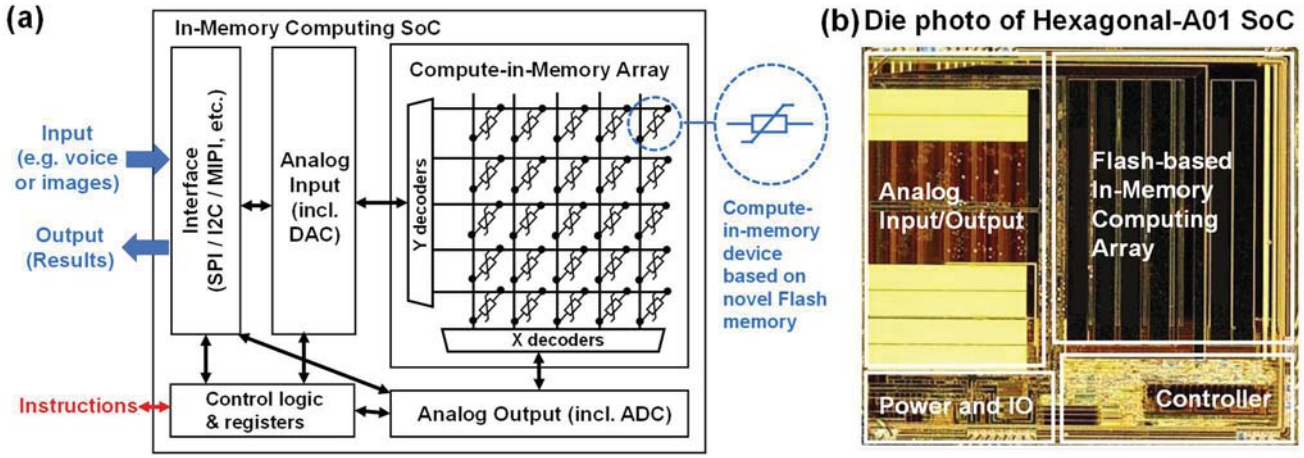


Fig. 2 (a) System architecture of the in-memory computing SoC designed based on novel Flash memory; **(b)** die photo of the in-memory computing SoC chip (Hexagonal-A01) based on novel Flash memory.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on October 10,2023 at 10:29:16 UTC from IEEE Xplore. Restrictions apply.

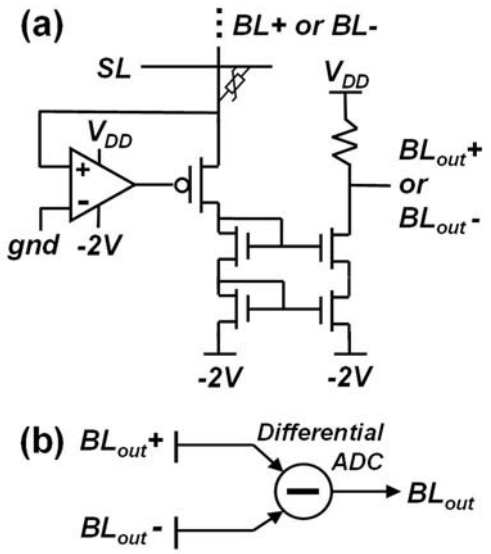


Fig. 4 Circuit diagrams of the analog output block: **(a)** sense amplifier with cascode current mirror for output voltage sensing; **(b)** schematics of positive and negative outputs subtraction with differential ADC.

on the bitline (BL). The separation of positive and negative weights on a pair of BLs and the differential ADC reduces the range of weights on each BL and allows higher precision of weight storage [5]. Moreover, by subtracting the currents on two BLs, some systematic errors of current sensing can be canceled out to produce more accurate digital output.

III. Neural Network Model Deployment Techniques

A. Sources of error in DNN computation with CIM array

In order to achieve highly accurate inference for AIoT applications, it is important to understand the non-ideal factors in the CIM-based paradigm and analyze the source of errors. These non-ideal factors include but are not limited to: device non-linearity [6], IR drop [7], non-ideal write [8], circuits mismatch, and die-to-die or device-to-device variations, etc.

By performing physical analysis of the errors, researchers sometimes came up with novel ideas targeted to mitigate/ eliminate a specific source of error. For example, the idea of differential representation of weights was proposed to offset the asymmetric programming behaviors of PCM-based CIM array and extend the dynamic range [5]. In another example, researchers use modeling and simulations to predict and mitigate the IR drop in a ReRAM-based binary neural network accelerator [7]. On the other hand, a realistic CIM system may contain multiple non-ideal factors, making it challenging to decouple and mitigate these factors one by one. We need a model deployment approach that is highly accurate, cost-effective and yet doesn't require much prerequisite knowledge about the model and the hardware platform.

B. On-chip transfer learning for model deployment and fine-tuning

Fig. 5(a) summarizes the concept of transfer learning and the corresponding domains in the problem of neural network model deployment in CIM arrays. As an example, the source domain can be the purely mathematical problem of keyword spotting (KWS) from voice signals, while the target domain is the optimization of weight data to be written into the CIM array (also called hardware-optimized weights), which can generate as close computation results as possible to those predicted by the pure mathematical model in the source domain. Compared to the source domain, the target-domain problem involves more complicated factors stemming from the non-ideality of the CIM array, such as: IR drop, non-linearity, variation, mismatch, etc. Yet, the two domains share significant similarities in terms of the mathematical structure of the problem. According to the theory and practices of transfer learning, this kind of problems are often called "domain adaption" and addressed by the approach of "model fine-tuning" [9].

In this work, we generalized the deep residual learning techniques proposed by K. He et al. [10] to realize model fine-

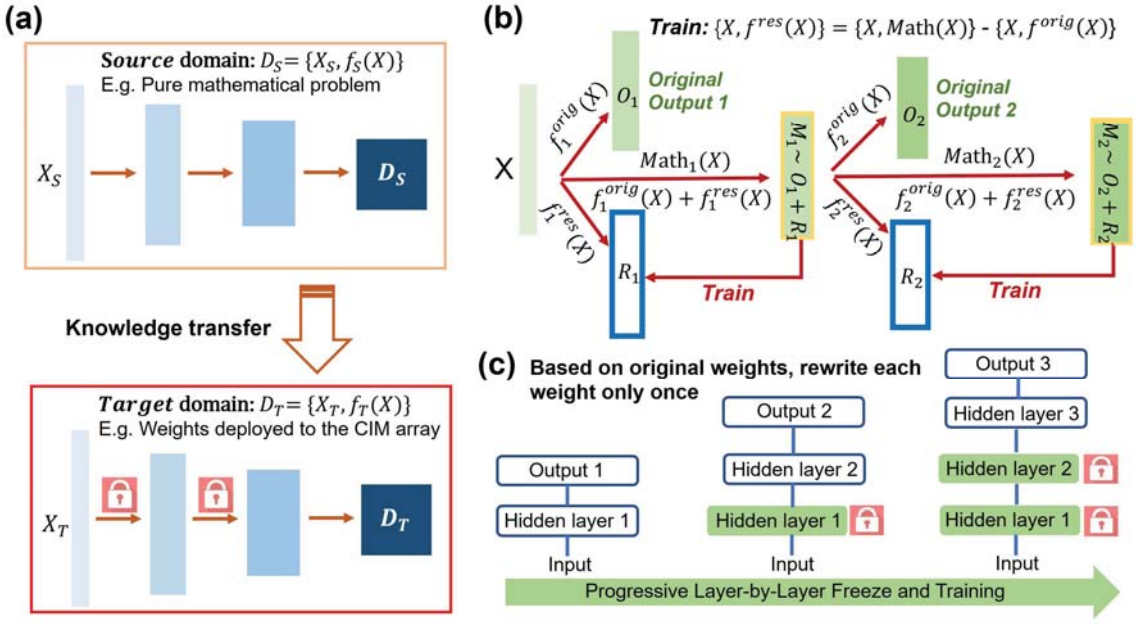


Fig. 5 (a) Transfer learning concepts and its application in model deployment; **(b)** Method to train a residual model which only needs one-time update after the initial deployment; **(c)** Progressive layer-by-layer freeze and training approach.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on October 10,2023 at 10:29:16 UTC from IEEE Xplore. Restrictions apply.

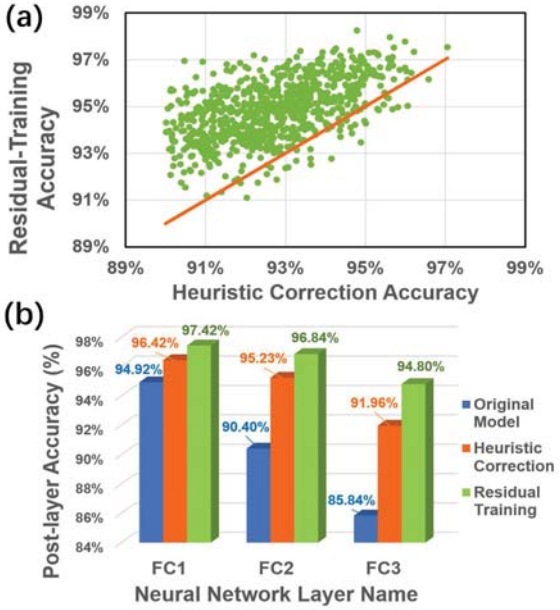
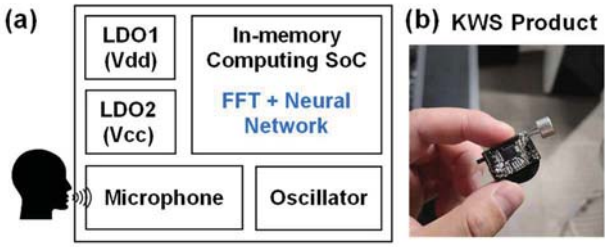


Fig. 6 (a) Comparison of post-FC3 accuracy between the residual-training approach and heuristic correction, each point corresponds to one input vector in the test set; **(b)** Post-layer accuracy improvements for each FC layer with the residual-training approach, compared to the original model and the heuristic correction approach.

tuning. This approach was originally proposed to counter the problem of vanishing gradients and reduce the training time for DNN. As shown in Fig. 5(b), instead of directly learning the hardware-optimized weights on chip, we train the residual function which is the difference between the hardware-optimized weights and the original weights. Compared to traditional training approaches [11,12], this residual training approach can help to reduce the frequency of writing memory cells and achieve more efficient training for non-volatile CIM arrays. The detailed approach is shown in Figs. 5(b-c), which indicate the training of residual model is done in a layer-by-layer fashion. Firstly, we deploy the theoretical model (fully-trained with GPU) into the CIM array and use the output of the 1st layer to train the residual model of the 1st layer. Secondly, we update the residual model only to the 1st layer and use the output of the first 2 layers to train the residual model of the 2nd layer. This procedure can be proceeded until the entire DNN is trained in the CIM arrays. During this process, we only need to rewrite each weight data once, minimizing the endurance concerns for Flash memories.

Fig. 6 summarizes the results of deployment accuracy using the residual-training approach and the comparison with deploying original weights or using an IR-drop based heuristic correction [7]. A simple voice-recognition model (3-layer perceptron with Mel coefficients as inputs) is applied to benchmark the inference accuracy in terms of the output's correlation coefficients with pure mathematical calculation. For each layer, the training of the residual model is carried out using a small training set of 10,000 input vectors and back propagation. The results suggest that the residual-training approach can strongly improve the CIM array's inference accuracy. Moreover, the progressive layer-by-layer freeze and training inspired by transfer learning is confirmed to be effective, resulting in similar improvements across every layer (improved ~3% per layer compared to original and ~1% per layer compared to heuristic correction). Finally, the overall



(c) Performance Benchmark across Hardware Platforms

Hardware	Running (W)	Dynamic (W)	Inf./Sec	Joules/ Inf.	Cost $
GPU	37.83	22.86	770.39	0.0298	$250
CPU	28.48	11.47	1813.6	0.0063	$200
JETSON	4.98	2.34	419	0.0056	$300
MOVIDIUS	0.647	0.437	300	0.0015	$99
LOIHI	0.110	0.081	296	0.00027	N/A
Hexagonal	0.03	0.015	101	0.00015	$1

Fig. 7 (a) Architecture of the fully integrated voice-recognition system (including SoC, microphone, power supplies, etc.); **(b)** a photo of the final product for KWS applications; **(c)** Performance benchmarks of the Flash-based in-memory computing SoC versus some existing solutions for multi-keyword spotting applications [13].

inference accuracy of the deployed voice-recognition model is improved from ~86.0% to ~94.8%, making it feasible for use in real products.

IV. System Integration and Benchmark

The developed Flash-based in-memory computing SoC can be integrated in a compact form factor for product design. As shown in Fig. 7(a), one such system has been demonstrated for multi-keyword spotting applications and is being offered for smart home applications. It is worth noting that the in-memory computing SoC also performs the computation of FFT and Mel coefficients in addition to DNN inference. The final product, as shown in Fig. 7(b), is powered by one 3V button cell battery to perform continuous speech recognition. The power efficiency of the entire system is characterized to be >10 TOPS/W for neural network inference. Fig. 7(c) summarizes the performance benchmark results of the multi-keyword spotting solution based on in-memory computing SoC, and compares it with existing solutions using CPU/GPU/CMOS-based neuromorphic chips [13]. From this comparison, the in-memory computing SoC shows an obvious cost advantage and also delivers better energy efficiency compared to conventional solutions.

V. Conclusion

We have demonstrated a fully integrated Flash-based in-memory computing SoC for voice-recognition applications. We combined a series of techniques from device, circuit-design and system perspectives to enable efficient neural network inference and signal processing. 7-bit/cell storage capability and adaptive write of novel PLRAM devices were leveraged to achieve state-of-the-art overall performance. Also, model deployment techniques based on transfer learning concepts are explored to significantly improve the accuracy loss during weight data deployment. The integrated voice-recognition system demonstrated >10 TOPS/W energy efficiency and ~95% inference accuracy as a real-time multi-keyword spotting engine.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on October 10,2023 at 10:29:16 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] D. Ielmini and H.-S. Philip Wong, "In-memory computing with resistive switching devices", Nature Electronics 1, 333-343(2018).

[2] S. Yu, "Neuro-inspired computing with emerging nonvolatile memory", Proceedings of the IEEE 106, 260-285(2018).

[3] A. Lee, M.-F. Chang, C.-C. Lin, C.-F. Chen, et al., "RRAM-based 7T1R nonvolatile SRAM with 2x reduction in store energy and 94x reduction in restore energy for frequent-off instant-on applications", Symposium on VLSI Circuits (VLSI Circuits), pp. 76-77, 2015.

[4] S.Gao, G. Yang, X. Qiu, C. Yang, et al., "Programmable linear RAM: A new flash memory-based memristor for artificial synapses and its application to speech recognition system", IEEE International Electron Devices Meeting (IEDM), pp. 14.1, 2019.

[5] G. W. Burr, R. M. Shelby, C. di Nolfo, J. W. Jang, et al., "Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses), using phase-change memory as the synaptic weight element", IEEE International Electron Devices Meeting (IEDM), pp. 29.5, 2014.

[6] T. Kim, H. Kim, J. Kim and J.-J. Kim, "Input voltage mapping optimized for resistive memory-based deep neural network hardware", IEEE Electron Device Lett. 38, 1228-1231(2017).

[7] S. Lee, G. Jung, M. E. Fouda, et al., "Learning to predict IR drop with effective training for ReRAM-based neural network hardware", 57th ACM/ IEEE Design Automation Conference (DAC), 2020.

[8] K. Roy, I. Chakraborty, M. Ali, A. Ankit, et al., "In-memory computing in emerging memory technologies for machine learning: an overview", 57th ACM/ IEEE Design Automation Conference (DAC), 2020.

[9] Y. Guo, H. Shi, A. Kumar, K. Grauman, et al., "SpotTune: transfer learning through adaptive fine-tuning", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4805-4814, 2019.

[10] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.

[11] M. Cheng, L. Xia, Z. Zhu, Y. Cai et al., "TIME: A training-in-memory architecture for memristor-based deep neural networks", 54th ACM/ IEEE Design Automation Conference (DAC), 2017.

[12] C. Li, D. Belkin, Y. Li, P. Yan et al., "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks", Nature Communications 9, 2385(2018).

[13] P. Blouw, X Choo, E. Hunsberger and C. Eliasmith, "Benchmarking keyword spotting efficiency on neuromorphic hardware", Proceedings of the 7th Annual Neuro-inspired Computational Elements Workshop, 2019.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on October 10,2023 at 10:29:16 UTC from IEEE Xplore. Restrictions apply.