IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 5, May 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105362 |

Prospects for Analog Circuits in Deep Networks

Prof. Manish Choudhary, Prof.Akansha Dixit

Department of Electronics & Communication Engineering, Baderia Global Institute of Engineering & Management, Jabalpur (M.P.), India

ABSTRACT: Operations commonly used in machine learning algorithms, such as additions and softmax, can be effectively implemented using compact analog circuits. Analog Application-Specific Integrated Circuit (ASIC) designs leverage techniques like charge-sharing circuits and subthreshold transistors to achieve high power efficiency. With recent advances in deep learning algorithms, there has been a shift towards digital hardware accelerator designs that focus on matrix-vector multiplication operations. In these designs, power consumption is primarily dominated by the memory access power required for off-chip DRAM used to store network weights and activations. Emerging dense non- volatile memory technologies offer the potential for on- chip memory, and analog circuits are well-suited to perform the necessary multiplication-vector operations in conjunction with in-memory computing approaches. This paper provides a brief review of analog designs that implement various machine learning algorithms and offers an outlook on the use of analog circuits in low- power deep network accelerators suitable for edge or tiny machine learning applications.

I. INTRODUCTION

Machine-learning systems produce state-of-art results for many applications including data mining and machine vision. They extract features from the incoming data on which de- cisions are made, for example, in a visual classification task. Current deep neural network approaches in machine learning (ML) [1] produce state-of-art results in many application domains including visual processing (object detection [2], face recognition [3] etc), audio processing (speech recognition [4], keyword spotting etc), and natural language processing [5], to name a few. This improvement in algorithms and software stack has eventually led to a drive for better hardware that can run such ML workloads efficiently.

In recent times, edge computing has become a big research topic and edge devices that first process the local input sensor data are being developed within different sensor domains. Many current deep network hardware accelerators designed for edge devices are implemented through digital circuits. Less explored are analog/mixed-signal designs that can provide high energy-efficient implementations of deep network architec- tures. Custom analog circuits can exploit the physics of the transistors in implementing computational primitives needed in deep network architectures. For example, operations such as summing can be implemented by simpler analog transistor circuits rather than digital circuits. These designs can provide better area and energy efficiencies than their digital coun- terparts in particular for edge inference applications which require only small networks. With the increasing availability of dense new non-volatile memory (NVM) technology, the com- putation can be co-localized with memory, therefore further savings in power can be obtained by the reduction of off-chip memory access.

This article reviews analog and mixed-signal hardware chips that implement ML algorithms including the deep neural network architectures. We discuss the advantages of analog circuits and also the challenges of using such circuits. While other reviews have focused on analog circuits for on-chip learning [6], we focus more on implementations of generic building blocks that can be used in both training and inference phases.

The paper organization is as follows: First, Sec. II provides a brief review on analog circuits used for neuromorphic comput- ing and how these circuits can implement computational prim- itives useful for neural processor designs and ML algorithms. It is followed by Sec. III that describes basic operations (such as vector-matrix multiplications) being accelerated by analog building blocks, and Sec. IV that reviews the field of non- volatile memory technologies for deep network accelerators. Section V reviews recent Application-Specific Integrated Cir- cuits (ASICs) that implement certain ML algorithms. Finally, we end the paper with a discussion about future prospects of analog-based ML circuits (e.g. in-memory computing).

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105362 |

II. REVIEW OF CIRCUITS FOR ANALOG COMPUTING

With great advances made in digital circuit design through the availability of both software and hardware design tools, analog circuits have taken a back seat in mainstream circuits and are now primarily used in high speed analog domains such as RF, power regulation, PLLs, and interface of sensors and actuators (e.g. in the analog-digital converter (ADC) readout circuits of analog microphone or biochemical sensors).

When it was proposed that the exponential properties of run- ning the transistors in subthreshold can be useful for emulating the structure of nervous system in the field of neuromorphic engineering [7], analog circuits became popular for emulating biological structures like the retina [8], implementing gen- eralized smoothing networks [9], [10], and both simplified and complex biophysical models of biological neurons. The exponential properties of a subthreshold transistor also make it easier to design analog VLSI circuits for implementing basic functions used in many mixed-signal neuromorphic systems such as the sigmoid, similarity [11], charge-based vector- matrix calculations [12], and highly distributed operations such as the winner-take-all [13].

Analog ML designs using these basic functions included a low-power analog radial basis function programmable sys- tem [14]; a sub-microWatt Support Vector Machine classifier design [15] with a computational efficiency of ~ 1 TOp/s/W; and the analog deep learning feature extractor system [16] that processes over 8K input vectors per second and achieves 1 TOp/s/W. The power efficiency of [16] is achieved by running the transistors in subthreshold and using floating-gate non-volatile technology for both storage, compensation and reconfiguration.

Even though analog neural processor circuits were already reported in the 1990s [17], they had taken a back seat partially because of the ease of doing digital designs; and the care needed for good matching in analog circuits. Although mismatch is usually reduced by using larger transistors, there are circuit strategies to minimize the mismatch effect, for example, by techniques to simplify the circuit complexity (Sec. III), floating-gate techniques to compensate for the mismatch, and training methods for deep networks to account for this mismatch as will be discussed in Sec. V.

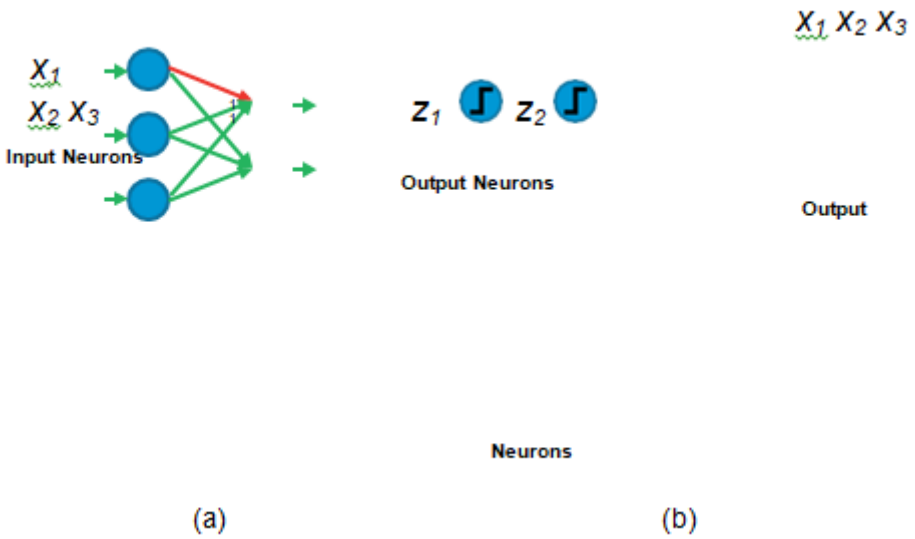


Fig. 1. Simple two-layer Artificial Neural Network (ANN) using a non-linear transfer function (a) and an implementation using a resistive crossbar array with differential weights: two non-negative resistors are used to represent a bipolar weight (b).

III. ANALOG CIRCUITS FOR MATRIX-VECTOR MULTIPLICATION

In a multi-layered neural network, the output of a neuron I in layer l is given by $y^l = g(z^l) = g(\sum w_{ij}x^l)$, where $g()$ is a nonlinear function and the j-th input to the l-th layer x^l is the j-th output from the previous layer (Fig. 1 (a)). The basic operations: summation, multiplication by scalars, and simple nonlinear transformations such as sigmoids can be imple- mented in analog VLSI circuits very efficiently. Dropping the index 1 denoting layer, we can write the most

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

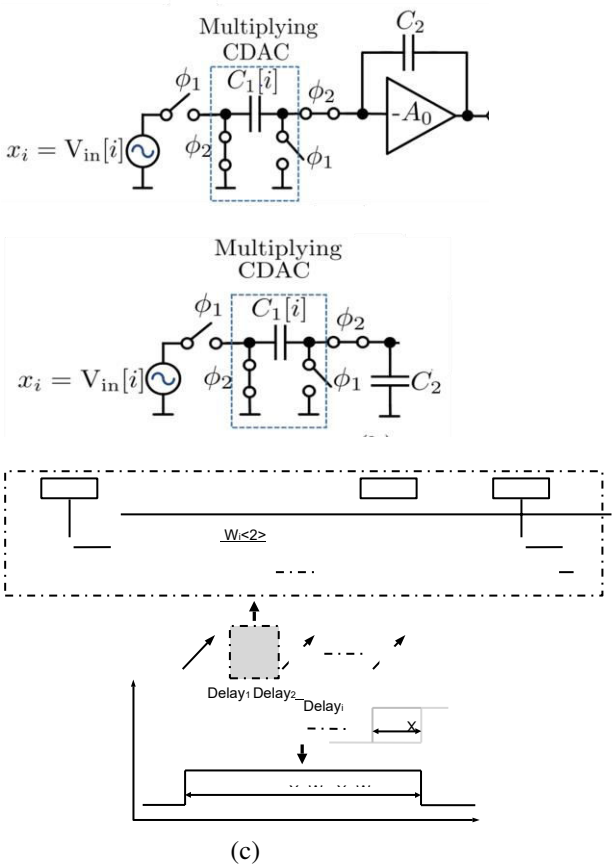
|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105362 |

computationally intensive part as a matrix-vector multiplication (MVM) as follows:

$$z = \mathbf{W}x \quad (1)$$

where x is the vector of inputs to a layer, **W** denote the matrix of weights for the layer and z denotes the vector of linear outputs which is then passed through a nonlinear function g() to produce the final neuronal output. Figure 2 illustrates how the key core computation of weighted summation can be



(c)

Fig. 2. Three modes of matrix-vector multipliers using analog circuits: (a) Charge (b) Time and (c) Current. Figures adapted from [18], [19] and [20].

performed in three modes: (a) Charge, (b) time and (c) current. Further storage of weight coefficients can be performed in a volatile or non-volatile manner. More details about these different possibilities are described next.

a. Charge: Figure 2(a) depicts a capacitive digital analog converter (CDAC) based charge mode MVM circuit. In this circuit, the weight is stored in the form of binary weighted capacitor sizes (shown as C1[i] in the figure) and hence weighting of input happens naturally by charging the desired capacitor value with the input voltage. Summation happens through the process of charge redistribution among capacitors via switched capacitor circuit principles. The charge redistri- bution can be more accurate if done by using an amplifier in a negative feedback configuration (as shown on top of Fig. 2(a))–this is referred to as an active configuration [18], [21]. On the other hand, the passive configuration without an amplifier (shown below the active one) has the benefit of high speed operation due to the absence of settling time limitations from the amplifier. However, it was shown in [18] that the errors in the passive multiplication amount to a change in the coefficients of the weighting matrix **W** and can be accounted for. A disadvantage of this approach is that it requires N clock cycles to perform a dot product on two N dimensional vectors. Static random access memory (SRAM) is used to store the weights–hence, it suffers from volatile storage issues. A 6-b input, 3-b weight implementation of this approach achieved ≈ 7.7 − 8.7 TOp/s/W energy efficiency at clock frequencies of 1 − 2.5 GHz clock frequency [18]. Other recent implementations using this approach are reported in [22] and [23].

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105362 |

b. Time: Another way of implementing addition in the analog domain is by using time delays. Figure 2(b) shows how addition can be implemented through the accumulation of delays in cascaded digital buffer stages [19]. By modifying the delay of each stage according to the weight, a weighted summation can be achieved. Figure 2(b) shows one example of a delay line based implementation of this approach. Here, the weight storage is in volatile SRAM cells. The authors used a thermometer encoded delay line as an unit delay cell to avoid nonlinearity at the expense of area. A 1-b input, 3-bit weight architecture achieved very high energy efficiency of ≈ 105 TOp/s/W at 0.7 V power supply [19].

c. Current: By far, the most popular approach for analog VMM implementation is the current mode approach. This architecture offers flexibility in choosing the way input is applied (encoding magnitude in current [24], voltage [25] or as pulse-width of a fixed amplitude pulse [26]) and non- volatile weight storage (Flash [27], PCM [28], RRAM [25], MRAM [29], Ferroelectric FETs [30], ionic floating-gate [31], transistor mismatch [32], etc). An example implementation using standard CMOS compatible Flash transistors shown in Fig. 2(c) was presented as early as 2004 [20].

In these approaches, a crossbar array of NVM devices are used to store the weights **W**. In the case of Flash, the inputs xj can be encoded in current magnitude ($I^{+} - I^{-}$) and presented j j along the columns (word-lines) while current summation occurs along the rows (bit-lines) by virtue of Kirchhoff's Current Law (KCL) to produce the output zj. Here, the weighting happens through a sub-threshold current mirror operation of the input diode-connected Flash transistor and the NVM device in the crossbar with weight being encoded as charge difference between the two devices. Since then, several variants of these Flash-based VMM architectures have been published using amplifier based active current mirrors [33] for higher speed, source coupled VMM for higher accuracy [34] and special Flash process based VMM [27] with promise of scaling down to 28nm. These approaches generally achieve energy
efficiencies in the range of 5 − 10 TOp/s/W.

An interesting alternative is to use the threshold voltage mismatch in-built in transistors as a weight quantity. This leads to ultra-compact arrays for VMM operation [24]. However, since these weights are random, they may only be used to implement a class of randomized neural networks such as reservoir computing, neural engineering framework, extreme learning machines etc. While these networks are typically only two layers deep, they have advantages of good generalization and quick retraining. Such approaches have been used for brain-machine interfaces [35], tactile sensing [36] and image classification [37].

The recent trend in this architecture is to use resistive memory elements, sometimes dubbed memristors, as the NVM device in the crossbar. They are less mature but offer ad- vantages of longer retention, higher endurance, low write energy, and promise of scaling to sizes smaller than Flash transistors. They have also been shown to support back- propagation based weight updates with small modifications to the architecture. Due to their increasing popularity in deep network implementations, we will discuss them further in the next section.

IV. NON-VOLATILE RESISTIVE CROSSBARS

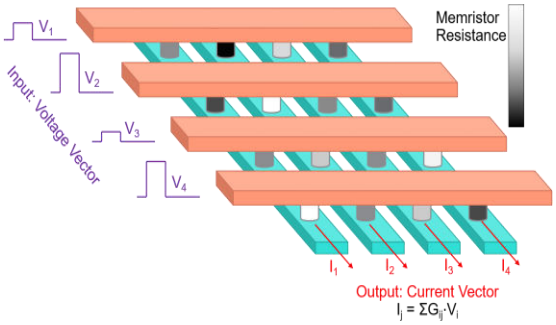
A popular approach for performing the matrix-vector multi- plications in ML - and especially in the deep neural networks of today - is to leverage emerging non-volatile memory technologies such as phase change (PCM), oxide-based re- sistive RAM (RRAM or memristors), and spin-torque mag- netic technologies (STT-MRAM). The equivalent of an ANN implemented in a resistive memory array is shown in Fig. 1 (b). As further illustrated in Fig. 3, a dense crossbar circuit is constructed with these resistive elements or memristors at every junction. Each resistor is analog tunable to encode mul- tiple bits of information (from binary to over seven bits [38], [39]). Most importantly, the devices retain this analog resistive programming in a non-volatile manner. This is appropriate for encoding the heavily reused weights in deep neural networks, such as convolution kernels or fully connected layers. And, as noted earlier, this allows for the dominant vector-matrix computations to be performed in-memory without the costly fetching of neural network (synaptic) weights.

To take full advantage of analog non-volatile crossbars in deep neural networks, it is necessary to devise a larger architecture that supplements the vector-matrix computations performed in-memory, with a range of additional data orches- tration and processing performed by traditional digital circuits. These include operations such as data routing, sigmoidal or other nonlinear activation functions, max-pooling, and nor- malization. Several prior works have explored this design- space [40], [41] for resistive crossbars, with the inclusion of schemes to encode arbitrarily high precision weights across multiple resistive elements ("bit-slicing") and the overhead in constantly converting all intermediate calculations performed within analog resistive crossbars back into the digital domain

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105362 |

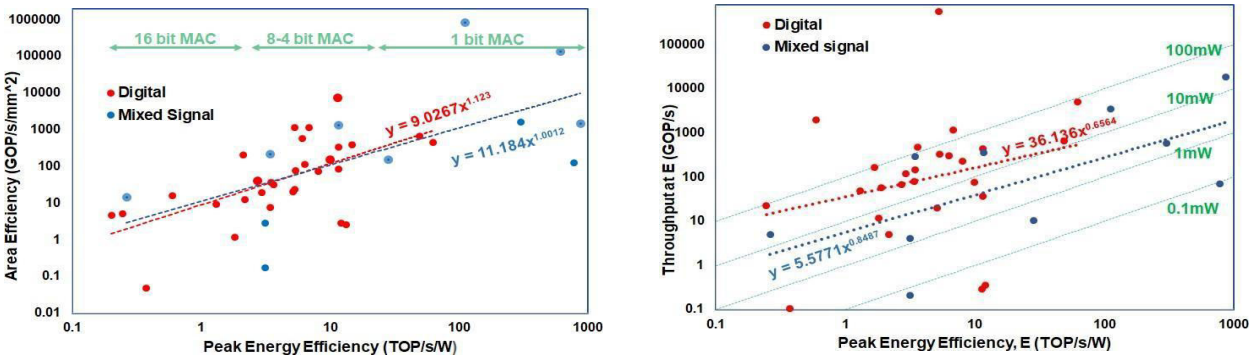


(a)

Fig. 3. Performing vector-matrix multiplication in non-volatile resistive (memristive) crossbar arrays.

The input can be encoded in voltage amplitude with fixed pulse width, as shown here, or other approaches such as fixed amplitude and variable pulse width. At each crossbar junction, a variable resistor is present representing a matrix weight element, modulating the amount of current injected into the column wires through Ohm's law. On each column, the currents from each row are summed through Kirchhoff's current law, yielding the desired output vector representing the vector-matrix product. Pairs of differential weights can be utilized as in Fig. 1(b) to allow bipolar representations. to avoid error accumulation in deep networks of potentially tens of layers. An architecture and performance analysis was first done for CNNs [40], and then extended to nearly all modern deep neural networks including flexible compiler support [42].

To date, non-volatile resistive crossbars have been explored for both the inference and training modes of deep neural networks. In the case of inference, the main advantage from NVM comes from reduced data-fetching and movement, while reprogramming of the crossbars is infrequent, requiring lower endurance from the resistive technology (PCM or RRAM), but potentially higher retention and programming yield ac- curacy [43]. To handle device faults, programming errors, or noise, novel techniques for performing error-detection and cor- rection for the computations performed within these crossbar arrays have been devised recently [44], [45] and experimen- tally demonstrated [46]. On the other hand, implementation of training in NVM crossbars [47], [48], [49] can take advantage of efficient update schemes for tuning each of the resistive weights in parallel (e.g., "outer product" updates such as in [50]). In turn, the implementation of neural network training puts a larger burden on the underlying NVM technology from an endurance perspective, and requirements for a symmetric and linear change in the device resistance when programmed up or down. While the technology remains highly promis- ing, the current state of resistive NVM suffers from large programming asymmetries and uniformity challenges. As the technology continues to mature, some current approaches are able to mitigate these setbacks by complementing the resistive technology with short term storage in capacitors [51], or oper- ating in a more binary mode [52], but with cost of either area or prediction accuracy. Nonetheless, practical demonstrations using resistive crossbar networks have already included, to name just a few, image filtering and signal processing [53], fully connected and convolutional networks [48], [39], [51], LSTM recurrent neural networks [25], and reinforcement learning [54].



(a) (b)

Fig. 4. Machine learning hardware trends: Peak energy efficiency vs throughput (a) and area efficiency (b) for recent ASIC implementations reported in ISSCC, SOVC, and JSSC (see main text).

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105362 |

V. FUTURE OF ANALOG DEEP NEURAL NETWORK ARCHITECTURES

Analog circuits can be more compact than digital circuits, especially for certain low-precision operations like the add operation needed in neural network architectures or the in- memory computing circuits for summing up weighted currents [55]. While mismatch variations could be a feature for certain network architectures like the Extreme Learning Net- works, they need to be addressed in analog implementations of deep networks by using design techniques such as the schemes described in Sec. III or through training methods that account for the statistics of the fabricated devices.
The work of [56] shows that it is possible to use neural net- work training methods as an effective optimization framework to automatically compensate for the device mismatch effects of analog VLSI circuits. The response characteristics of the individual VLSI neurons are added as constraints in an off- line training process, thereby compensating for the inherent variability of chip fabrication and also taking into account the particular analog neuron's transfer function achievable in a technology process. The measured inference results from the fabricated analog ASIC neural network chip [56] matches that from the simulations, while offering lower power consumption over a digital counterpart. A similar retraining scheme was used to correct for memristor defects in a memristor-based synaptic network trained on a classification task [57].

A. Trends in machine learning ASICs

To understand the energy and area efficiency trends in recent ASIC designs of ML systems, we present a survey [58] of papers published in IEEE ISSCC and IEEE SOVC conferences from 2017-2020 (similar to the ADC survey in [59]), and in the IEEE Journal of Solid-state Circuits (JSSC).

Using the data from [60], we show two plots. The first plot shows the throughput vs. peak energy efficiency tradeoff for digital and mixed-signal designs (Fig. 4(a)). It can be seen that while throughput is comparable for analog and digital approaches, the energy efficiency for mixed-signal designs is superior to their digital counterparts. Figure 4(b) plots the area efficiency vs peak energy efficiency for these designs and further classifies the digital designs according to the bit-width of the multiply-accumulate (MAC) unit. It can again be seen that the mixed-signal designs have better tradeoff than the digital counterparts. Furthermore, as expected, lower bit widths lead to higher efficiencies in terms of both area and energy. Lastly, the efficiency of mixed-signal designs seem significantly better than digital designs only for these with very low-precision 1-bit MACs.

VI. CONCLUSION

While deep network accelerator designs are implemented primarily using digital circuits, in-memory computing systems can benefit from analog and mixed-signal circuits for niche ultra low-power edge or biomedical applications. The peak energy efficiency trends in Fig. 4 show that mixed-signal designs can be competitive at lower-bit precision parameters for network accelerators. Some of the analog machine learning circuit blocks such as the sigmoid and the winner-take-all de- signs can implement directly the sigmoid transfer function of the neuron, and the soft-max function of the final classification layers of a deep network. It will be interesting to see how these circuits will be incorporated further into in-memory computing systems in the future.

ACKNOWLEDGMENT

The authors would like to thank Mr. Sumon K. Bose and Dr. Joydeep Basu for help with figures; and T. Delbruck for comments.

REFERENCES

1. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.
2. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classifica- tion with deep convolutional neural networks," in Neural Information Processing Systems (NIPS), 2012, pp. 1106–1114.
3. Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in CVPR, 2014.
4. G. Hinton, L. Deng, and D. Y. et. al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal Process. Mag., vol. 29, no. 6, p. 82–97, 2012.
5. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," Journal of Machine Learning Research, vol. 12, p. 2493–2537, 2011.A. Basu, J. Acharya,

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105362 |

and et. al., "Low-power, adaptive neuromorphic systems: Recent progress and future directions," IEEE Journal of Emerg- ing Topics in Circuits and Systems, vol. 8, no. 1, pp. 6–27, 2018.
6. C. A. Mead, Analog VLSI and Neural Systems. Addison-Wesley, 1989.
7. C. A. Mead and M. A. Mahowald, "A silicon model of early visual processing," Neural Networks, vol. 1, no. 1, pp. 91–97, 1988.
8. S.-C. Liu and J. G. Harris, "Generalized smoothing networks in early vision," in Proceedings CVPR'89: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1989, pp. 184–191.
9. J. G. Harris, S.-C. Liu, and B. Mathur, "Discarding outliers using a nonlinear resistive network," in IJCNN-91-Seattle International Joint Conference on Neural Networks, vol. i, July 1991, pp. 501–506.
10. T. Delbruck and C. Mead, "Bump circuits," in Proceedings of Interna- tional Joint Conference on Neural Networks, vol. 1, 1993, pp. 475–479.
11. R. Genov and G. Cauwenberghs, "Charge-mode parallel architecture for vector-matrix multiplication," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 48, no. 10, pp. 930–936, 2001.
12. S.-C. Liu, J. Kramer, G. Indiveri, T. Delbru¨ck, and R. Douglas, Analog VLSI: Circuits and principles. MIT Press, 2002.
13. S.-Y. Peng, P. E. Hasler, and D. V. Anderson, "An analog programmable multidimensional radial basis function based classifier," IEEE Transac- tions on Circuits and Systems I: Regular Papers, vol. 54, no. 10, pp. 2148–2158, 2007.
14. S. Chakrabartty and G. Cauwenberghs, "Sub-microwatt analog VLSI trainable pattern classifier," IEEE Journal of Solid-State Circuits, vol. 42, no. 5, pp. 1169–1179, May 2007.
15. J. Lu, S. Young, I. Arel, and J. Holleman, "A 1 TOPS/W analog deep machine-learning engine with floating-gate storage in 0.13 µm cmos," IEEE Journal of Solid-State Circuits, vol. 50, no. 1, pp. 270–281, Jan 2015.
16. P. Masa, K. Hoen, and H. Wallinga, "A high-speed analog neural processor," Micro, IEEE, vol. 14, no. 3, pp. 40–50, 1994.
17. E. H. Lee and S. S. Wong, "Analysis and design of a passive switched- capacitor matrix multiplier for approximate computing," IEEE Journal of Solid State Circuits (JSSC), vol. 52, no. 1, pp. 261–271, 2017.
18. L. Everson, M. Liu, N. Pande, and C. H. Kim, "A 104.8TOPS/W one-shot time-based neuromorphic chip employing dynamic threshold error correction in 65nm," in Proceedings of Asian Solid-State Circuits Conference (ASSCC), 2018.
19. R. Chawla, A. Bandyopadhyay, V. Srinivasan, and P. Hasler, "A 531 nW/MHz, 128 × 32 current-mode programmable analog vector-matrix multiplier with over two decades of linearity," in Proceedings of Custom Integrated Circuits Conference (CICC), 2004.
20. E. H. Lee and S. S. Wong, "A 2.5GHz 7.7TOPS/W switched-capacitor matrix multiplier with co-designed local memory in 40nm," in Proceed- ings of International Solid-State Circuits Conference (ISSCC), 2016.
21. K. A. Sanni and A. G. Andreou, "A Historical Perspective on Hardware AI Inference, Charge-Based Computational Circuits and an 8 bit Charge- Based Multiply-Add Core in 16 nm FinFET CMOS," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 3, pp. 532–543, 2019.
22. S. Joshi, C. Kim, S. Ha, and G. Cauwenberghs, "From algorithms to devices: Enabling machine learning through ultra-low-power VLSI mixed-signal array processing," in Proceedings of Custom Integrated Circuits Conference (CICC), 2017.
23. Y. Chen, Z. Wang, A. Patil, and A. Basu, "A 2.86-TOPS/W current mirror cross-bar based machine-learning and physical unclonable func- tion engine for internet-of-things applications," IEEE Transactions on Circuits and Systems-I, vol. 66, no. 6, 2019.
24. C. Li and et. al, "Long short-term memory networks in memristor crossbar arrays," Nature Machine Intelligence, vol. 1, no. 9, pp. 49–57, 2019.
25. M. J. Marinella, S. Agarwal, A. Hsia, I. Richter, R. Jacobs-Gedrim,J. Niroula, S. J. Plimpton, E. Ipek, and C. D. James, "Multiscale co- design analysis of energy, latency, area, and accuracy of a ReRAM analog neural training accelerator," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 8, no. 1, pp. 86–101, 2018.
26. F. Merrikh-Bayat, X. Guo, and et.al, "High-performance mixed-signal neurocomputing with nanoscale floating-gate memory cell arrays," IEEE Transactions on Neural Networks and Learning Systems (TNNLS), vol. 29, no. 10, pp. 4782–90, 2018.
27. M. Suri, O. Bichler, D. Querlioz, O. Cueto, L. Perniola, V. Sousa,D. Vuillaume, C. Gamrat, and B. DeSalvo, "Phase change memory as synapse for ultra-dense neuromorphic systems: Application to complex visual pattern extraction," in 2011 International Electron Devices Meet- ing, 2011, pp. 4–4.
28. J. Grollier, D. Querlioz, and M. D. Stiles, "Spintronic nanodevices for bioinspired computing," Proceedings of the

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105362 |

IEEE, vol. 104, no. 10, pp. 2024–2039, 2016.

29. M. Jerry, P.-Y. Chen, J. Zhang, P. Sharma, K. Ni, S. Yu, and S. Datta, "Ferroelectric fet analog synapse for acceleration of deep neural network training," in 2017 IEEE International Electron Devices Meeting (IEDM). IEEE, 2017, pp. 6–2.
30. E. J. Fuller, S. T. Keene, A. Melianas, Z. Wang, S. Agarwal, Y. Li,Y. Tuchman, C. D. James, M. J. Marinella, J. J. Yang et al., "Parallel programming of an ionic floating-gate memory array for scalable neu- romorphic computing," Science, vol. 364, no. 6440, pp. 570–574, 2019.
31. E. Yao and A. Basu, "VLSI extreme learning machine: A design space exploration," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 1, pp. 60–74, 2017.
32. A. Basu and et. al, "A floating-gate based field programmable analog array," IEEE Journal of Solid State Circuits (JSSC), vol. 45, no. 9, pp. 1781–94, 2010.
33. C. Schlottmann and P. E. Hasler, "A highly dense, low power, pro- grammable analog vector-matrix multiplier: The FPAA implementation," IEEE Journal of Emerging and Selected Topics in Circuits and Systems (JETCAS), vol. 1, no. 3, pp. 403–410, 2011.
34. Y. Chen, E. Yao, and A. Basu, "A 128-channel extreme learning machine-based neural decoder for brain machine interfaces," IEEE Transactions on Biomedical Circuits and Systems, vol. 10, no. 3, pp. 679–692, 2016.
35. M. Rasouli and et. al., "An extreme learning machine-based neuromor- phic tactile sensing system for texture recognition," IEEE Transactions on Biomedical Circuits and Systems, vol. 12, no. 2, pp. 313–325, 2018.
36. A. Patil and et. al., "Hardware architecture for large parallel array of ran- dom feature extractors applied to image recognition," Neurocomputing, vol. 261, pp. 193–203, 2017.
37. F. Alibart, L. Gao, B. D. Hoskins, and D. B. Strukov, "High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm," Nanotechnology, vol. 23, no. 7, p. 075201, 2012.
38. M. Hu, C. E. Graves, C. Li, Y. Li, N. Ge, E. Montgomery, N. Davila,H. Jiang, R. S. Williams, J. J. Yang et al., "Memristor-based analog com- putation and neural network classification with a dot product engine," Advanced Materials, vol. 30, no. 9, p. 1705914, 2018.
39. A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Stra- chan, M. Hu, R. S. Williams, and V. Srikumar, "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," ACM SIGARCH Computer Architecture News, vol. 44, no. 3, pp. 14–26, 2016.
40. L. Song, X. Qian, H. Li, and Y. Chen, "Pipelayer: A pipelined ReRAM- based accelerator for deep learning," in 2017 IEEE International Sym- posium on High Performance Computer Architecture (HPCA). IEEE, 2017, pp. 541–552.
41. A. Ankit, I. E. Hajj, S. R. Chalamalasetti, G. Ndu, M. Foltin, R. S. Williams, P. Faraboschi, W.-m. W. Hwu, J. P. Strachan, K. Roy et al., "PUMA: A programmable ultra-efficient memristor-based accelerator for machine learning inference," in Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, 2019, pp. 715–731.
42. J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," Nature nanotechnology, vol. 8, no. 1, p. 13, 2013.
43. R. M. Roth, "Fault-tolerant dot-product engines," IEEE Transactions on Information Theory, vol. 65, no. 4, pp. 2046–2057, 2018.
44. R. M. Roth, "Analog error-correcting codes," IEEE Transactions on Information Theory, vol. 66, no. 7, pp. 4075–4088, 2020.
45. C. Li, R. M. Roth, C. Graves, X. Sheng, and J. P. Strachan, "Analog error correcting codes for defect tolerant matrix multiplication in crossbars," in 2020 IEEE International Electron Devices Meeting (IEDM). IEEE, 2020, pp. 36–6.
46. G. W. Burr, R. M. Shelby, S. Sidler, C. Di Nolfo, J. Jang, I. Boybat,R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti 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 Transactions on Electron Devices, vol. 62, no. 11, pp. 3498–3507, 2015.
47. M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromor- phic network based on metal-oxide memristors," Nature, vol. 521, no. 7550, pp. 61–64, 2015.
48. A. Sebastian, I. Boybat, M. Dazzi, I. Giannopoulos, V. Jonnalagadda,V. Joshi, G. Karunaratne, B. Kersting, R. Khaddam-Aljameh, S. R. Nandakumar, A. Petropoulos, C. Piveteau, T. Antonakopoulos, B. Ra- jendran, M. L. Gallo, and E. Eleftheriou, "Computational memory-based inference and training of deep neural networks," in 2019 Symposium on VLSI Technology, June 2019, pp. T168–T169.
49. S. Agarwal, T.-T. Quach, O. Parekh, A. H. Hsia, E. P. DeBenedictis,C. D. James, M. J. Marinella, and J. B.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105362 |

Aimone, "Energy scaling advantages of resistive memory crossbar based computation and its application to sparse coding," Frontiers in Neuroscience, vol. 9, p. 484, 2016.

50. S. Ambrogio, P. Narayanan, H. Tsai, R. M. Shelby, I. Boybat,C. di Nolfo, S. Sidler, M. Giordano, M. Bodini, N. C. Farinha et al., "Equivalent-accuracy accelerated neural-network training using ana- logue memory," Nature, vol. 558, no. 7708, p. 60, 2018.
51. W.-H. Chen, C. Dou, K.-X. Li, W.-Y. Lin, P.-Y. Li, J.-H. Huang, J.-H. Wang, W.-C. Wei, C.-X. Xue, Y.-C. Chiu et al., "CMOS-integrated memristive non-volatile computing-in-memory for AI edge processors," Nature Electronics, vol. 2, no. 9, pp. 420–428, 2019.
52. C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song,N. Da´vila, C. E. Graves et al., "Analogue signal and image processing with large memristor crossbars," Nature Electronics, vol. 1, no. 1, p. 52, 2018.
53. Z. Wang, C. Li, W. Song, M. Rao, D. Belkin, Y. Li, P. Yan, H. Jiang,P. Lin, M. Hu et al., "Reinforcement learning with analogue memristor arrays," Nature Electronics, vol. 2, no. 3, p. 115, 2019.
54. A. Biswas and A. P. Chandrakasan, "Conv-RAM: An energy-efficient SRAM with embedded convolution computation for low-power CNN- based machine learning applications," in 2018 IEEE International Solid- State Circuits Conference-(ISSCC), 2018, pp. 488–490.
55. J. Binas, D. Neil, G. Indiveri, S.-C. Liu, and M. Pfeiffer, "Analog electronic deep networks for fast and efficient inference," in Proc Conf. Syst. Mach. Learning, 2018.
56. C. Liu, M. Hu, J. P. Strachan, and H. Li, "Rescuing memristor-based neuromorphic design with high defects," in 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), 2017, pp. 1–6.
57. S. K. Bose, J. Acharya, and A. Basu, "Is my neural network neuromor- phic? Taxonomy, recent trends and future directions in neuromorphic engineering," in Asilomar Conference on Signals, Systems, and Com- puters, 2019.
58. B. Murmann, "ADC performance survey 1997-2019," http://web. stanford.edu/∼murmann/adcsurvey.html, 2019.
59. S. K. Bose, J. Acharya, and A. Basu, "Survey of neuromorphic and machine learning accelerators in SOVC, ISSCC and Nature/Science series of journals from 2017 onwards," https://sites.google.com/view/ arindam-basu/neuromorphic-survey-asilomar, 2019.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.379

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details