Software Impacts 15 (2023) 100473

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

Simulation of memristive crossbar arrays for seizure detection and prediction using parallel Convolutional Neural Networks

Chenqi Li [a], Corey Lammie [b], Amirali Amirsoleimani [c], Mostafa Rahimi Azghadi [b,*], Roman Genov [d]

[a] *Division of Engineering Science, University of Toronto, 42 St George St, Toronto, M5S 2E4, Ontario, Canada*
[b] *College of Science and Engineering, James Cook University, 1 James Cook Dr, Townsville, QLD 4811, Queensland, Australia*
[c] *Department of Electrical Engineering and Computer Science, York University, 4700 Keele St, Toronto, M3J 1P3, Ontario, Canada*
[d] *Department of Electrical and Computer Engineering, University of Toronto, 42 St George St, Toronto, M5S 2E4, Ontario, Canada*

ARTICLE INFO

Keywords:
CNN
Seizure detection
Seizure prediction
EEG
RRAM
Memristive crossbar array

ABSTRACT

For epileptic seizure detection and prediction, to address the computational bottleneck of the von Neumann architecture, we develop an in-memory memristive crossbar-based accelerator simulator. The simulator software is composed of a Python-based neural network training component and a MATLAB-based memristive crossbar array component. The software provides a baseline network for developing deep learning-based signal processing tasks, as well as a platform to investigate the impact of weight mapping schemes and device and peripheral circuitry non-idealities.

Code metadata

Current code version	v1
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2022-232
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/7908097/tree/v1
Legal Code License	*GNU v3.0*
Code versioning system used	git
Software code languages, tools, and services used	MATLAB2019b, Python, Bash
Compilation requirements, operating environments & dependencies	Python ≥3.7 and MATLAB ≥ 2019b. Specific Python requirements are listed at: Link
If available Link to developer documentation/manual	Link
Support email for questions	mostafa.rahimiazghadi@jcu.edu.au

1. Introduction

The unpredictability of seizure occurrences and lack of understanding of the underlying mechanism of epilepsy introduce many challenges to managing seizure symptoms [1–3]; especially for patients with drug-resistant epilepsy. An accurate epileptic seizure prediction system would inform patients and first-responders, in a timely manner, to intervene before seizures occur. Electroencephalogram (EEG) is a commonly used device to monitor brain activity and can detect changes in activities associated with seizure events. Deep Learning (DL) has shown to be a promising solution to tackle many engineering problems, outperforming State-Of-The-Art (SOTA) methods. The main advantage of DL networks lies within their ability to automatically extract features [4]. The main drawback of DL networks is increased model complexity and computational time. Through parallelization, Graphics Processing Units (GPUs) can reduce the training and inference time for DL networks, however, within the von Neumann architecture, the need to constantly transfer data between memory and computing units is difficult to parallelize [5,6]. In-Memory Computing (IMC) addresses the aforementioned bottleneck, achieving constant time complexity, $\mathcal{O}(1)$, for Multiply-Accumulate (MAC) operations [7].

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail address: mostafa.rahimiazghadi@jcu.edu.au (M.R. Azghadi).

https://doi.org/10.1016/j.simpa.2023.100473
Received 2 January 2023; Received in revised form 22 January 2023; Accepted 24 January 2023

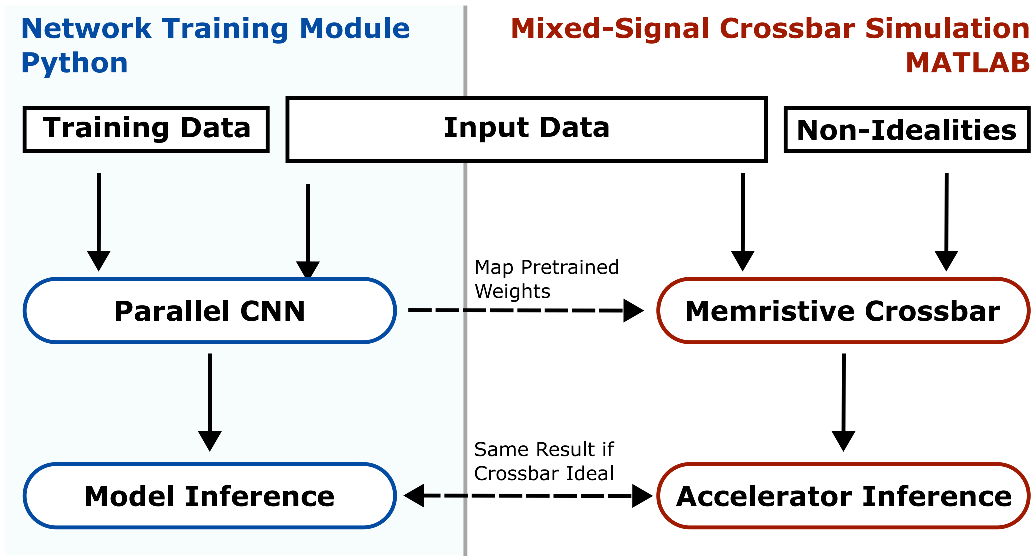

Fig. 1. Software system overview.

Table 1
Seizure detection performance comparison on university of Bonn dataset.
Source: Adapted from [8].

Paper	Pre-processing	Method	Parallelization	Parameters	Accuracy (%)
[9]	✓	1D-CNN	✗	21,436	99.90
[10]	✓	1D-CNN	✗	16,778,144	92.00
[11]	✓	2D-CNN	✗	106,388	98.00
[12]	✓	2D-CNN	✗	1,603,080	99.45
Abdelhameed et al. (2021)	✓	2D-CNN	✗	10,304,467	**100.00**
Ours	✗	1D-CNN	✓	**10,778**	99.84

*Not reported.

Using our software, we simulated a parallel Convolutional Neural Network (CNN) using memristive crossbar arrays employing IMC, and demonstrated that our system is capable of achieving SOTA performance while requiring 2-2800x fewer network parameters and 2 orders of magnitude reduction in latency compared to hybrid memristive-Complementary Metal–Oxide–Semiconductor (CMOS) accelerators [8]. We also investigated the impact of device and circuit non-idealities, and proposed new methods to mitigate such impacts. In this work, we describe the software that enabled the training of the adopted neural networks and the simulations of our hybrid memristive-CMOS accelerator design.

2. Impact overview

The software is composed of two major components, a Python-based parallel CNN training module, and a MATLAB-based inference memristive crossbar simulator (see Fig. 1). Currently, three datasets, SWEC-ETHZ, Bonn, and CHBMIT, are supported.

2.1. Parallel convolutional neural networks

Using Pytorch, a parallel CNN architecture is implemented and tested across all three datasets. This serves to facilitate the development of methods for seizure detection and prediction, by providing a lightweight, SOTA CNN architecture for the research community to deploy or improve upon. Tables 1 and 2 provide a comparison of seizure detection and prediction performance of our proposed parallel CNN against SOTA models in literature. In fact, the parallel network architecture can be applied to various time-series-based classification tasks, such as EEG emotion recognition or Electrocardiogram (ECG) myocardial infarction detection.

Fixed seeds and deterministic algorithms are used to ensure results are reproducible and consistent across runs. This enables a fair comparison and exploration of varying network architectures, hyperparameters, and preprocessing techniques. During training, checkpoints of the model with the best accuracy are saved. A postprocessing script is employed to convert `.h5` checkpoints to text files, as an interface between the Python-based training module and the MATLAB-based crossbar simulator. Saved checkpoints not only enable communication between Python and MATLAB components, but they also enable future hardware deployment investigation and transfer learning.

2.2. Memristive crossbar array simulation

To perform a simulation of memristive crossbar arrays, a crossbar array model by A. Chen that takes into account line resistance and nonlinear device characteristics is employed [20]. The crossbar model source code is implemented in MATLAB, in order to leverage matrix computation efficiency to solve for output crossbar currents at each column with given input voltages at each row.

The pretrained network, in the form of text files, is imported and mapped onto 7 64×64 crossbar arrays. To enable the representation of both negative and positive weights, a differential weight mapping scheme is employed, whereby the left and right columns represent negative and positive weights respectively. The true weight is thus the difference between right and left memristor weights.

While Chen's model takes into account line resistance and nonlinear device characteristics, it fails to consider other device and circuit non-idealities. Our software improves upon Chen's model to take into account several crossbar non-idealities, including input and output resolutions, weight write resolution, weight write deviation, stuck R_{ON}/R_{OFF} devices, line and source resistance, and conductance range

Table 2
Seizure prediction performance comparison on SWEC-ETHZ and CHB-MIT datasets.
Source: Adapted from [8].

Paper	Method	Parallelized	Parameters	Sensitivity (%)	Specificity (%)	Accuracy (%)	FPR[b]
CHB-MIT							
[13]	2D-CNN	✗	N/R[c]	81.20	N/R[c]	N/R[c]	**0.16**
[10][a]	2D-CNN	✗	N/R[c]	N/R[c]	N/R[c]	92.00	N/R[c]
[14]	2D-CNN	✗	49,560	82.71	88.21	98.19	N/R[c]
[15][a]	2D-CNN	✗	N/R[c]	88.80	88.60	88.70	N/R[c]
[16][a]	3D-CNN	✗	28,459,615	96.66	99.14	98.33	N/R[c]
[17][a]	2D-CNN	✗	9,695,012	84.00	99.00	99.00	0.2
[18]	1D-CNN	✓	105,538	95.55	**99.68**	**99.64**	N/R[c]
Ours	1D-CNN	✓	**10,778**	**99.24**	98.68	99.01	0.47
SWEC-ETHZ							
[19][a]	Ensemble HD	✗	N/R[c]	96.38	97.31	96.85	N/R[c]
[18]	1D-CNN	✓	105,538	94.57	**99.86**	**99.81**	N/R[c]
Ours	1D-CNN	✓	**10,778**	**98.22**	97.02	97.54	**0.99**

[a]Indicates the results are reported across the entire dataset and patient-wise performance was not reported.
[b]False positive rate (per hour).
[c]Not reported.

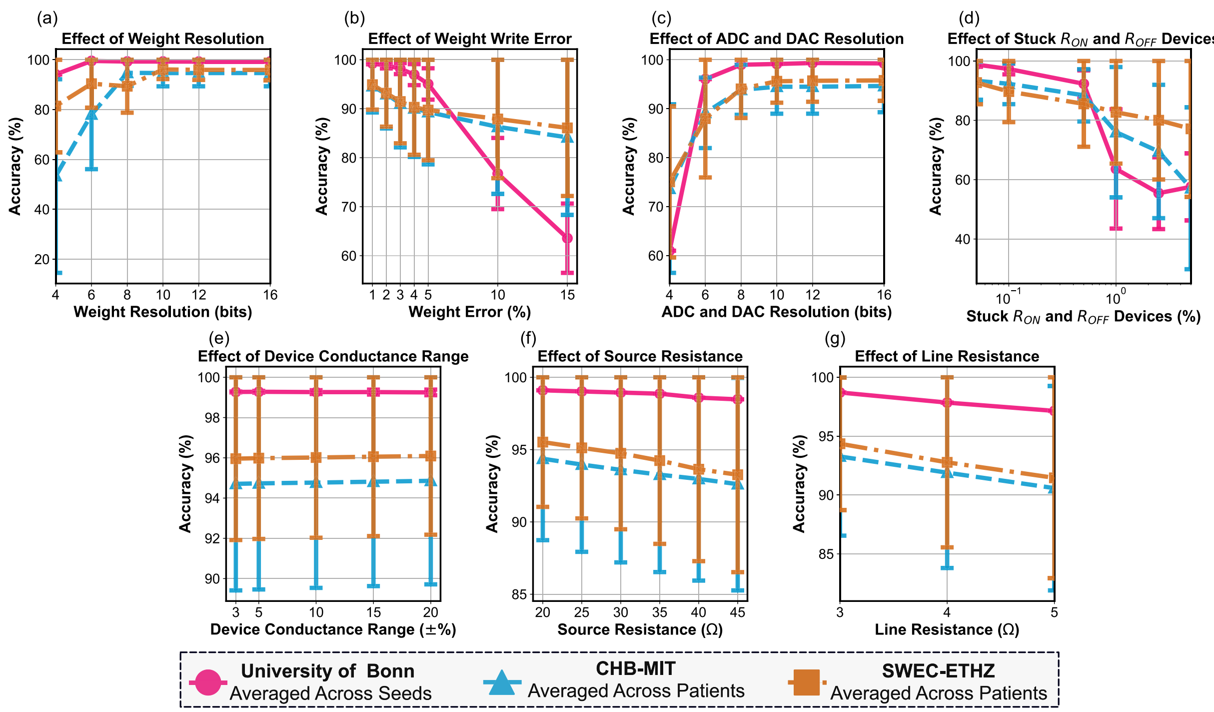

Fig. 2. Impact of hardware non-idealities on performance degradation.
Source: Adapted from [8].

variation. All of the non-idealities are simulated by varying the memristor weights or input voltages and output currents of the crossbars. The degree of variation for non-idealities can be customized through pre-defined variables in the configuration section of different scripts. Therefore, our simulation software can be easily adapted by other researchers to comprehensively simulate their custom crossbar array designs. Fig. 2 summarizes the simulated impact of different non-idealities on the hardware performance of our proposed parallel CNN for seizure detection and prediction. An interesting application of our simulation is to investigate how different network weight mapping schemes and layouts impact the vulnerability to different non-idealities. In other words, how can crossbar weights be mapped to minimize the impact of device non-idealities?

With all crossbar weights mapped and non-idealities considered, output currents can be solved using input voltages and crossbar weights. Digital circuit blocks can also be simulated through corresponding computations and operations on the output currents. Outputs from crossbars are fed to subsequent arrays until the final inference result is computed. The operation can be repeated for the entire dataset, and final metrics can be computed. While we applied our system to the application of epileptic seizure detection and prediction using DL, our simulator can be used to simulate any algorithm that adopts matrix–vector multiplication operations. To comprehensively benchmark our memristive inference accelerator, between 3 to 7 degrees of variation for each non-idealities are simulated using 5 different seeds for all 3 datasets.

3. Conclusion and future improvements

In this work, we presented an end-to-end pipeline for the training and simulation of hybrid memristive-CMOS accelerators for epileptic seizure detection and prediction. The Python-based neural network training and validation component can facilitate future work on DL-based signal processing tasks. The MATLAB-based memristive crossbar array simulation provides a comprehensive benchmark for custom inference accelerators, as well as providing a platform to investigate the impact of weight mapping schemes and layouts on the system's vulnerability to non-idealities. For our application, a fixed network architecture is employed. The MATLAB simulation component relies on the assumption of fixed network architecture and crossbar configurations. Automatic mapping of different network architecture weights onto different crossbar array configurations would greatly facilitate an end-to-end verification pipeline, from network training to simulation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work of Corey Lammie was supported in part by JCU DRTPS and in part by IBM Ph.D. Fellowship Program. The work of Mostafa Rahimi Azghadi was supported by a JCU Rising Star ECR Fellowship. The work of Roman Genov and Amirali Amirsoleimani was supported by NSERC. The work of Roman Genov was also supported by CIHR under Grants 149090, 153037, 146465, and 149090.

References

[1] D.C. Patel, B.P. Tewari, L. Chaunsali, H. Sontheimer, Neuron-glia interactions in the pathophysiology of epilepsy, Nat. Rev. Neurosci. 20 (5) (2019) 282–297.
[2] G.P. Brennan, D.C. Henshall,
[3] S. Gasparini, E. Ferlazzo, C. Sueri, V. Cianci, M. Ascoli, S.M. Cavalli, E. Beghi, V. Belcastro, A. Bianchi, P. Benna, R. Cantello, D. Consoli, F.A. De Falco, G. Di Gennaro, A. Gambardella, G.L. Gigli, A. Iudice, A. Labate, R. Michelucci, M. Paciaroni, P. Palumbo, A. Primavera, F. Sartucci, P. Striano, F. Villani, E. Russo, G. De Sarro, U. Aguglia, Hypertension, seizures, and epilepsy: a review on pathophysiology and management, Neurol. Sci. 40 (9) (2019) 1775–1783.
[4] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, http://dx.doi.org/10.1038/nature14539.
[5] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C.E. Graves, Z. Li, J.P. Strachan, P. Lin, Z. Wang, M. Barnell, Q. Wu, R.S. Williams, J.J. Yang, Q. Xia, Analogue signal and image processing with large memristor crossbars, Nat. Electron. 1 (1) (2018) 52–59, http://dx.doi.org/10.1038/s41928-017-0002-z.
[6] C. Lammie, O. Krestinskaya, A. James, M.R. Azghadi, Variation-aware binarized memristive networks, in: 2019 26th IEEE International Conference on Electronics, Circuits and Systems, ICECS, 2019, pp. 490–493, http://dx.doi.org/10.1109/ICECS46596.2019.8964998.
[7] C. Lammie, W. Xiang, M.R. Azghadi, Towards memristive deep learning systems for real-time mobile epileptic seizure prediction, in: 2021 IEEE International Symposium on Circuits and Systems, ISCAS, 2021, http://dx.doi.org/10.1109/ISCAS51556.2021.9401080.
[8] C. Li, C. Lammie, X. Dong, A. Amirsoleimani, M.R. Azghadi, R. Genov, Seizure detection and prediction by parallel memristive convolutional neural networks, IEEE Trans. Biomed. Circuits Syst. (2022).
[9] I. Ullah, M. Hussain, E. ul Haq Qazi, H. Aboalsamh, An automated system for epilepsy detection using EEG brain signals based on deep learning approach, Expert Syst. Appl. 107 (2018) 61–71, http://dx.doi.org/10.1016/j.eswa.2018.04.021.
[10] T. Wen, Z. Zhang, Deep convolution neural network and autoencoders-based unsupervised feature learning of EEG signals, IEEE Access 6 (2018) 25399–25410.
[11] A. Abdelhameed, M. Bayoumi, A deep learning approach for automatic seizure detection in children with epilepsy, Front. Comput. Neurosci. 15 (2021) 29.
[12] Ö. Türk, M.S. Özerdem, Epilepsy detection by using scalogram based convolutional neural network from EEG signals, Brain Sci. 9 (5) (2019) 115.
[13] N.D. Truong, A.D. Nguyen, L. Kuhlmann, M.R. Bonyadi, J. Yang, S. Ippolito, O. Kavehei, Convolutional neural networks for seizure prediction using intracranial and scalp electroencephalogram, Neural Netw. 105 (2018) 104–111.
[14] M.S. Hossain, S.U. Amin, M. Alsulaiman, G. Muhammad, Applying deep learning for epilepsy seizure detection and brain mapping visualization, ACM Trans. Multimedia Comput. Commun. Appl. (TOMM) 15 (1s) (2019) 1–17.
[15] J. Cao, J. Zhu, W. Hu, A. Kummert, Epileptic signal classification with deep EEG features by stacked CNNs, IEEE Trans. Cogn. Dev. Syst. 12 (4) (2019) 709–722.
[16] X. Tian, Z. Deng, W. Ying, K.-S. Choi, D. Wu, B. Qin, J. Wang, H. Shen, S. Wang, Deep multi-view feature learning for EEG-based epileptic seizure detection, IEEE Trans. Neural Syst. Rehabil. Eng. 27 (10) (2019) 1962–1972.
[17] W. Liang, H. Pei, Q. Cai, Y. Wang, Scalp EEG epileptogenic zone recognition and localization based on long-term recurrent convolutional network, Neurocomputing 396 (2020) 569–576.
[18] X. Wang, X. Wang, W. Liu, Z. Chang, T. Kärkkäinen, F. Cong, One dimensional convolutional neural networks for seizure onset detection using long-term scalp and intracranial EEG, Neurocomputing 459 (2021) 212–222.
[19] A. Burrello, S. Benatti, K. Schindler, L. Benini, A. Rahimi, An ensemble of hyperdimensional classifiers: Hardware-friendly short-latency seizure detection with automatic iEEG electrode selection, IEEE J. Biomed. Health Inf. 25 (4) (2020) 935–946.
[20] A. Chen, A comprehensive crossbar array model with solutions for line resistance and nonlinear device characteristics, IEEE Trans. Electron Devices 60 (4) (2013) 1318–1326, http://dx.doi.org/10.1109/TED.2013.2246791.