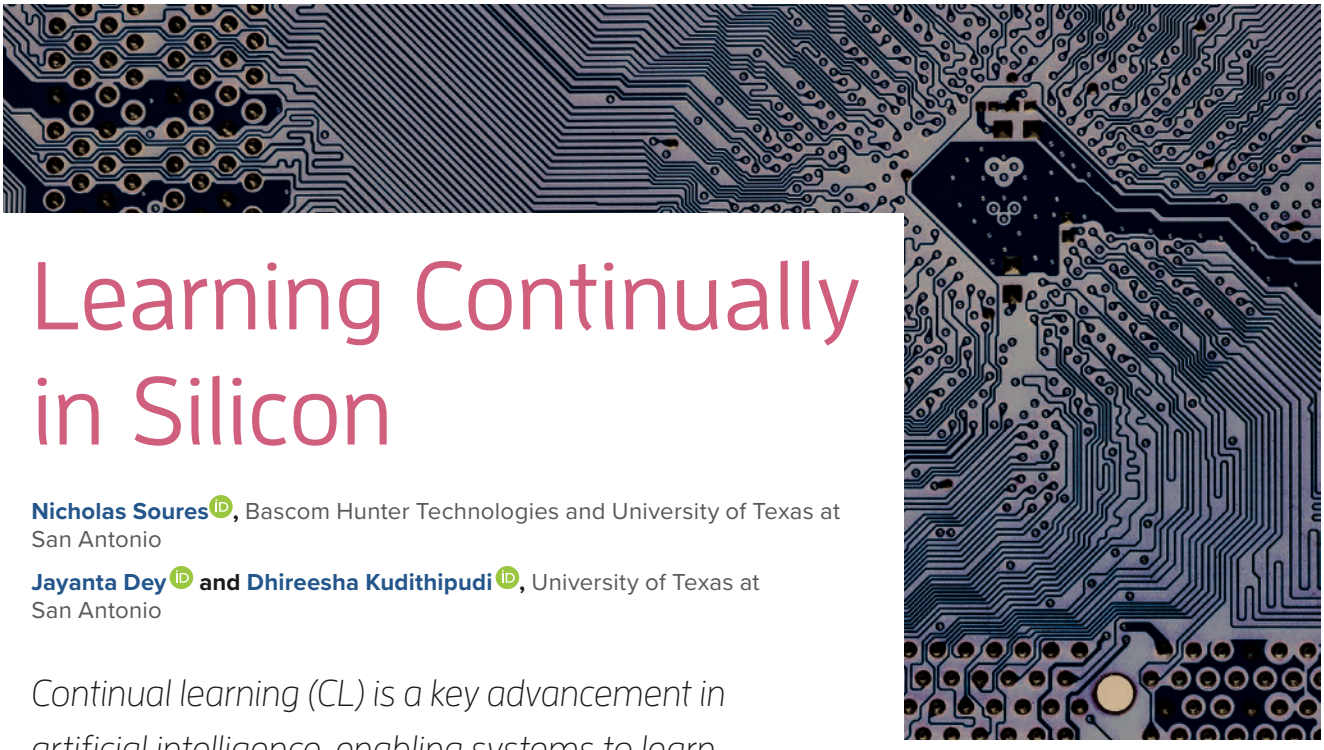

Learning Continually in Silicon

Nicholas Soures, Bascom Hunter Technologies and University of Texas at San Antonio

Jayanta Dey and **Dhireesha Kudithipudi**, University of Texas at San Antonio

Continual learning (CL) is a key advancement in artificial intelligence, enabling systems to learn and adapt over time. We introduce the concept and features of how to approach CL in silicon, with early examples from the literature.

Continual learning (CL) is a challenge in which an agent must learn from sequential, nonstationary data from changing distributions. The agent is expected not only to learn new tasks more efficiently, consuming fewer resources, such as storage, computing time, and energy, but also to retain and even improve performance on previously learned tasks. Although training on interleaved task data can improve the learner's performance on all of the tasks, the sequential nature of the task data in CL complicates the problem (see Figure 1). Moreover, interleaved training would be untenable when scaling over an entire lifetime.

The literature is replete with examples of CL algorithms that can be broadly classified into two groups: resource constrained and resource growing. Resource-constrained algorithms[1,2,3,4,5] use fixed-capacity schemes, reallocating or regularizing model parameters to learn new tasks. Fixed-capacity algorithms are useful when the learning environment has finite complexity. By contrast, resource-growing approaches[6,7,8] expand the model's capacity as more tasks are encountered and thus can cope with a learning environment with growing or infinite complexity. In both groups, an important strategy to improve CL algorithms includes replaying or rehearsing old data to mitigate forgetting of old tasks.[9,10,11] However, all algorithms target specific aspects of CL rather than approaching it holistically, leading to a continuous development of new concepts and methods. This presents a significant challenge in designing CL hardware architectures for existing and evolving algorithms.[12]

On the other hand, the current artificial intelligence (AI) accelerators primarily concentrate on a single task, which contrasts with the sequential multitask challenge faced by continual learners. While these accelerators support some aspects of CL, they lack many essential features required for successful deployment (see Table 1). For instance, edge AI accelerators, similar to CL systems, have limited computational capabilities, often operate on battery power, and encounter several other constraints. In

Digital Object Identifier 10.1109/MC.2024.3437828
Date of current version: 23 September 2024

0018-9162/24©2024IEEE

EDITOR **CONRAD JAMES**
Sandia National Laboratories, USA;
cdjame@sandia.gov

line with Occam's razor, the design of CL accelerators can be regarded as a type of multiscale optimization with simple plasticity and learning blocks.

In this article, we review some of the key features in designing CL accelerators. Extensive details on CL can be found in Kudithipudi et al.[12,13]

CL ON SILICON

Current AI accelerators predominantly support inference and plasticity (see Table 1 for a subset of machine learning and spiking edge accelerators). This can be partly attributed to the siloed evolution of the hardware and models, which often led to fragmented solutions. We can achieve progress in specialized hardware for CL with co-design approaches that encompass plasticity blocks coupled with novel learning architectures. Recently, we introduced common features for designing such accelerators.[12] Here, we highlight key aspects to consider for building such

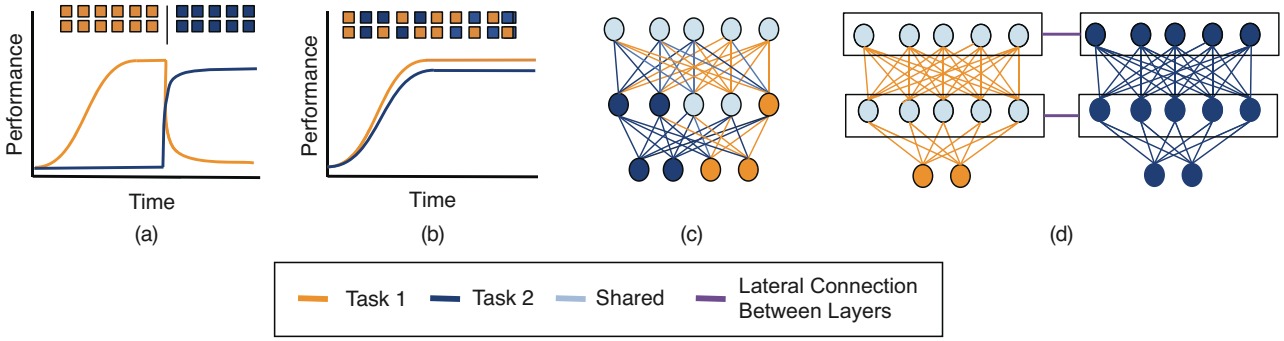

FIGURE 1. (a) Sequential training from different tasks results in catastrophic forgetting of older tasks for traditional deep learning algorithms. b) Forgetting is mitigated when the learner is trained on task data in an interleaved fashion. CL strategies include (c) the *resource-constrained* scheme, in which multiple tasks are learned with fixed number of parameters, and (d) the *resource-growing* scheme, in which new parameters are added with new tasks. (Adapted from van de Ven et al.[14])

TABLE 1. Overview of recent AI accelerators with on-device training which support single-task learning but only partially support CL. Attributes such as quantization, sparsity, and on-chip memory configurations support different resource-growing and resource-constrained plasticity mechanisms.

Chip	Quantization	Neural network(s)	Power	Throughput	On-chip memory
Chimera[15]	INT 8	DNN, CNN	418 mW (40 nm)	0.92 TOP/s	2 MB[c]
Tenstorrent Wormhole[16]	FP16, FP8, bfloat16	DNN, CNN, LSTM	80 W (12 nm)	430 TOP/S	120 MB
SIGMA[17]	FP16/32	DNN, RNN, CNN	22.3 W (28 nm)	10.8 TFLOPS	68 MB
Loihi[18]	INT1-INT9[a]	Spiking CNN, SNN, LSTM	420 mW (14 nm)	50 FPS (10 kHz)	33 MB
SpiNNaker 2[19]	FXP32, FP32	DNN, CNN, SNN	~0.72 W (22 nm)[b]	4.6 TOP/s (250 MHz)	18 MB SRAM, 8 GB off-chip DRAM
SCOLAR[20]	FxP16	SNN	21.25 mW (65 nm)	250 MOP/s (10 MHz)	100 KB (SRAM)

[a]Signed or unsigned integer or mixed-precision number is supported.
[b]Processing element power.
[c]RRAM instead of SRAM.
FPS: frames/s; FP: floating point; FX: fixed point; DNN: deep neural network; CNN: convolutional neural network; SNN: spiking neural network; LSTM: long short-term memory; DRAM: dynamic random-access memory; SRAM: static random-access memory; RRAM: resistive random-access memory.

accelerators, including on-device learning, memory topologies, programmability, and reconfigurable architectures, as shown in Figure 2.

On-device plasticity

Many classes of CL algorithms exhibit diverse forms of plasticity, enabling the design of adaptive architectures that facilitate both local and global learning across multiple tasks. Examples of this are changes in synaptic strength (local) or in network architecture (global and structural). To facilitate this, CL architectures should support various loss/regularization functions, learning from replayed data, growth and/or pruning of networks at different abstractions (synapse, neuron, layer, or network), and neuromodulation for context-dependent processing.

Memory topologies

In both resource-growing and resource-constrained CL algorithms, memory topology plays a key role. Models must store dynamic parameters to regulate learning, store previous activations or parameters, and store previously seen data or generative models scaling with task complexity. The data communication costs for these techniques can be reduced by near-memory or in-memory computing approaches. Based on sample size, access frequency, and replay stage (sleep or wake), the data can be located at higher levels of the memory hierarchy or in buffers close to the compute substrate. There is also an opportunity to apply conventional techniques for power gating to memory, such as in sleep modes. Hardware state checkpointing, to recover from catastrophic deviations caused by changes in a new stimulus, is also important for CL models. While common in software, implementing tracking and checkpointing of model changes imposes several challenges in hardware. Coalescing data flow and layout as well as recomputation to reduce memory accesses can be beneficial in such scenarios.

Reconfigurable architectures

CL models frequently experience dynamic changes, requiring fine-grain reconfigurability at runtime. The system should transition learning from streaming data to accessing, processing, and learning from batched samples of previous experiences. The architecture should offer on-demand access to new computing or memory resources at runtime and generate new resources that were not previously available (such as potential synapse) as needed.

Programmability

CL accelerators will need to support flexible operations within a fixed resource or energy budget. There is, however, a tradeoff between the programmability and the energy efficiency of accelerators. Translating this into accelerator design requires flexibility in reassigning resources under size, weight, and power constraints, where several features are necessary but need not be active simultaneously. For example, a system might periodically alternate between structural plasticity, regularization, and replay, or activate them all in tandem.

CO-DESIGN

When developing CL accelerators, optimizations informed by both hardware and algorithms are critical to enhance performance and scalability. Such a co-design process is adopted in SCOLAR for CL with a resource-constrained algorithm (metaplasticity). At the algorithmic level, SCOLAR[20] introduced parameter sharing to reduce memory overhead, constraining the hyperparameter search space to powers of two where beneficial.

A similar effort[21] with emerging devices took advantage of the inherent properties of memristive devices to implement resource-constrained learning (metaplasticity). Instead of using a dynamic learning rate, the probabilistic nature of state changes in memristor devices regulated learning in response to plasticity. The system operates with low-precision weights, parameter sharing, and sparse weight updates. Error thresholding and the accumulation of gradient signals further reduced data flow and the overall write frequency to memristor devices. Such examples illustrate the effectiveness of co-design for CL hardware.

As the algorithmic space for CL grows, so will the scope of features for CL accelerators and their design complexity. There is an increasingly greater need for adopting a co-design paradigm rather than siloed development. For instance, communities such as TinyML have put forth a set of guidelines for general edge AI accelerators targeting submilliwatt power consumption. Such criteria will probably be essential for designers of CL accelerators. Although some preliminary criteria are given in Kudithipudi et al. 2023,[12] they are likely to expand as the algorithmic space advances. Critical issues moving forward include 1) demonstrating scalable on-device plasticity for large-scale applications, 2) understanding the relation and integrating different CL mechanisms to design reconfigurable architectures, 3) efficient memory topologies that facilitate CL and minimize energy consumption, and 4) developing new evaluation criteria and metrics to assess CL accelerators. An avenue of research in advancing the state of CL accelerators will revolve around emerging technology beyond traditional CMOS, such as resistive random-access memory (RAM). Finally, biological brains can provide inspiration for new forms of plasticity and learning, not available in current AI models, as CL is a natural phenomenon seen in biological systems.

> "Computers are incredibly fast, accurate, and stupid. Human beings are incredibly slow, inaccurate, and brilliant. Together they are powerful beyond imagination."
>
> —*Albert Einstein*

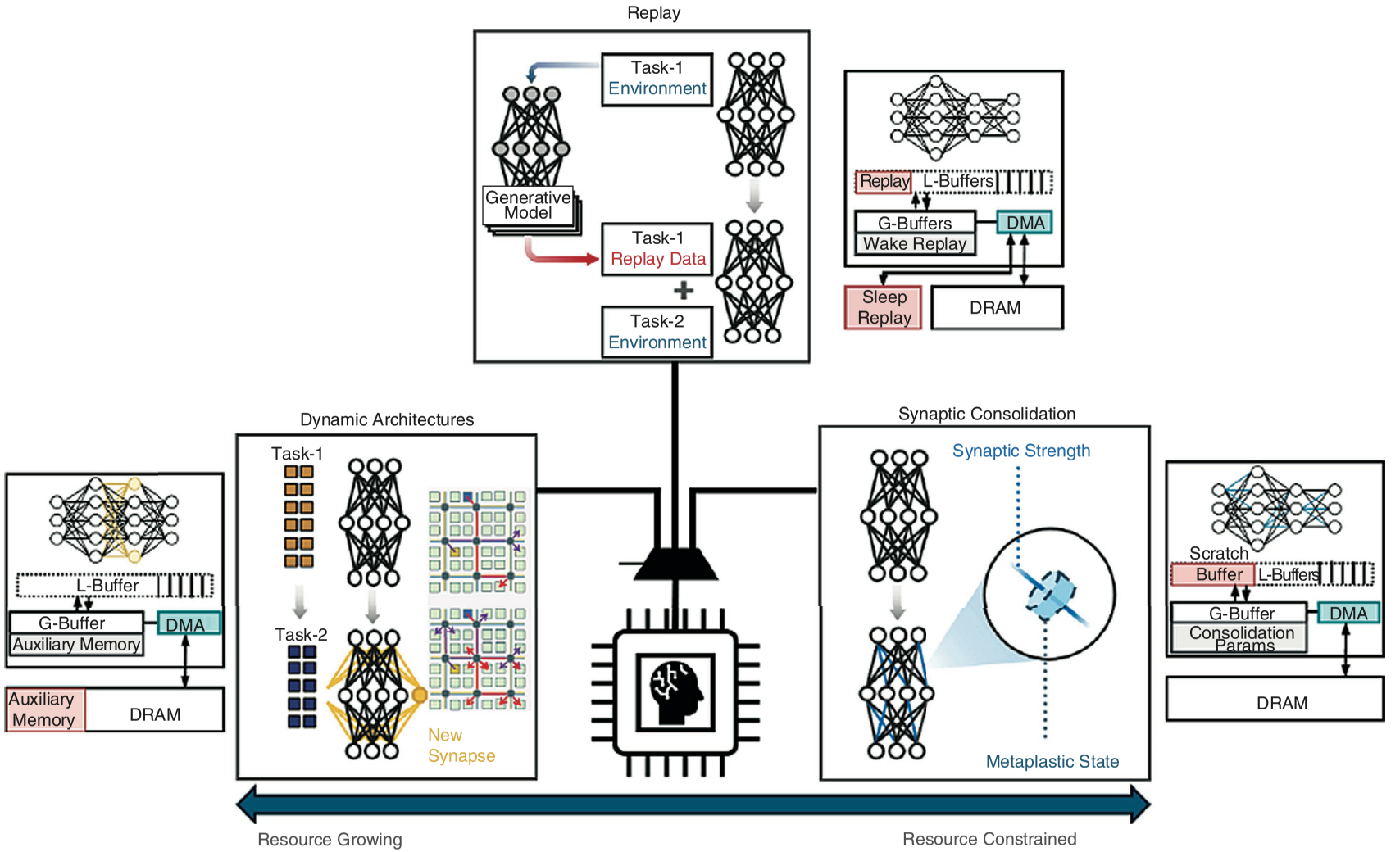

FIGURE 2. Overview of CL algorithm approaches and the associated architectures that can be selected based on the complexity of the problem. The architectures feature potential memory topologies and fine-grain reconfigurability to incorporate different plasticity and learning methods. DRAM: dynamic random access memory; DMA: direct memory access. (Adapted from Kudithipudi et al. 2023.[12])

ACKNOWLEDGMENT

This effort is partially supported by the National Science Foundation (NSF) Emerging Frontiers in Research and Innovation Brain-Inspired Dynamics for Engineering Energy-Efficient Circuits and Artificial Intelligence Award 2317706, the NSF Partner Neuro-Inspired AI for the Edge at the University of Texas at San Antonio Award 2332744, and the Air Force Research Laboratory under agreement FA8750-24-2-0151.

REFERENCES

1. J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, 2017, doi: 10.1073/pnas.1611835114.
2. F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. 34th Int. Conf. Mach. Learn*, 2017, vol. 70, pp. 3987–3995.
3. Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2017, doi: 10.1109/TPAMI.2017.2773081.
4. J. Schwarz et al., "Progress and compress: A scalable framework for continual learning," 2018, *arXiv:1805.06370*.
5. C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, "Online meta-learning," in K. Chaudhuri and R. Salakhutdinov, Eds., *Proc. Int. Conf. Mach. Learn.*, Long Beach, CA, USA: PMLR, Jun. 2019, vol. 97, pp. 1920–1930.
6. A. A. Rusu et al., "Progressive neural networks," 2016, *arXiv:1606.04671*.
7. J. T. Vogelstein et al., "Representation ensembling for synergistic lifelong learning with quasilinear complexity," 2020, *arXiv:2004.12908*.
8. R. Ramesh and P. Chaudhari, "Model Zoo: A growing brain that learns continually," in *Proc. Int. Conf. on Learn. Representations*, 2021.
9. G. M. van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature Commun.*, vol. 11, no. 1, 2020, Art. no. 4069, doi: 10.1038/s41467-020-17866-2.
10. A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Sci.*, vol. 7, no. 2, pp. 123–146, 1995, doi: 10.1080/09540099550039318.
11. H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2990–2999.
12. D. Kudithipudi et al., "Design principles for lifelong learning AI accelerators," *Nature Electron.*, vol. 6, no. 11, pp. 807–822, 2023, doi: 10.1038/s41928-023-01054-3.
13. D. Kudithipudi et al., "Biological underpinnings for lifelong learning machines," *Nature Mach. Intell.*, vol. 4, no. 3, pp. 196–210, 2022, doi: 10.1038/s42256-022-00452-0.
14. G. M. van de Ven, N. Soures, and D. Kudithipudi, "Continual learning and catastrophic forgetting," 2024, *arXiv:2403.05175*.
15. K. Prabhu et al., "CHIMERA: A 0.92-TOPS, 2.2-TOPS/W edge AI accelerator with 2-MByte on-chip foundry resistive RAM for efficient training and inference," *IEEE J. Solid-State Circuits*, vol. 57, no. 4, pp. 1013–1026, Apr. 2022, doi: 10.1109/JSSC.2022.3140753.
16. D. Ignjatovic´, D. W. Bailey, and L. Bajic´, "The wormhole AI training processor," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Piscataway, NJ, USA: IEEE Press, 2022, vol. 65, pp. 356–358, doi: 10.1109/ISSCC42614.2022.9731633.
17. E. Qin et al., "SIGMA: A sparse and irregular GEMM accelerator with flexible interconnects for DNN training," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Piscataway, NJ, USA: IEEE Press, 2020, pp. 58–70, doi: 10.1109/HPCA47549.2020.00015.
18. M. Davies et al., "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan./Feb. 2018, doi: 10.1109/MM.2018.112130359.
19. C. Mayr, S. Hoeppner, and S. Furber, "SpiNNaker 2: A 10 million core processor system for brain simulation and machine learning-keynote presentation," in *Communicating Process Architectures 2017 and 2018*, Amsterdam, The Netherlands: IOS Press, 2019, pp. 277–280.
20. V. Karia, F. T. Zohora, N. Soures, and D. Kudithipudi, "SCOLAR: A spiking digital accelerator with dual fixed point for continual learning," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Piscataway, NJ, USA: IEEE Press, 2022, pp. 1372–1376, doi: 10.1109/ISCAS48785.2022.9937294.
21. F. T. Zohora, V. Karia, N. Soures, and D. Kudithipudi, "Probabilistic metaplasticity for continual learning with memristors," 2024, *arXiv:2403.08718*.
22. A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Netw.*, vol. 111, pp. 47–63, Mar. 2019, doi: 10.1016/j.neunet.2018.12.002.

NICHOLAS SOURES is a senior artificial intelligence/machine learning engineer at Bascom Hunter Technologies, Baton Rouge, LA 70809 USA, and a postdoctoral fellow at the Neuromorphic Artificial Intelligence Lab, University of Texas at San Antonio, San Antonio, TX 78249 USA. Contact him at nms9121@rit.edu.

JAYANTA DEY is a postdoctoral fellow in the Neuromorphic Artificial Intelligence Lab at the University of Texas at San Antonio, San Antonio, TX 78249 USA. Contact him at jayanta.dey@utsa.edu.

DHIREESHA KUDITHIPUDI is the founding director of the Matrix AI Consortium, the Robert F. McDermott Endowed Chair in Engineering, and a professor of electrical and computer engineering/computer science at the University of Texas at San Antonio, San Antonio, TX 78249 USA. Contact her at dhireesha.kudithipudi@utsa.edu.