

EDITORIAL | JUNE 26 2023

Brains and bytes: Trends in neuromorphic technology

Adnan Mehonic   ; Jason Eshraghian 



APL Machine Learning 1, 020401 (2023)

<https://doi.org/10.1063/5.0162712>



CrossMark

Articles You May Be Interested In

Neotype kuramite optoelectronic memristor for bio-synaptic plasticity simulations

J. Chem. Phys. (May 2023)

Two Zaslavsky maps in pseudorandom byte generation

AIP Conference Proceedings (October 2019)

Cellular automata based byte error correcting codes over finite fields

AIP Conference Proceedings (August 2012)

Brains and bytes: Trends in neuromorphic technology

Cite as: *APL Mach. Learn.* **1**, 020401 (2023); doi: [10.1063/5.0162712](https://doi.org/10.1063/5.0162712)

Submitted: 15 June 2023 • Accepted: 15 June 2023 •

Published Online: 26 June 2023



View Online



Export Citation



CrossMark

Adnan Mehonic^{1,a)}  and Jason Eshraghian² 

AFFILIATIONS

¹ Department of Electronic and Electrical Engineering, University College London, Torrington Place, London WC1E 7JE, United Kingdom

² Department of Electrical and Computer Engineering, University of California, Santa Cruz, Santa Cruz, California 95064, USA

^{a)} Author to whom correspondence should be addressed: adnan.mehonic.09@ucl.ac.uk

<https://doi.org/10.1063/5.0162712>

The term “neuromorphic” was originally introduced by Mead in the late 1980s,¹ referring to devices and systems that imitated certain elements of biological neural systems. However, today the interpretation of the term has diverged across different research communities. Broadly speaking, “neuromorphic” relates to computing approaches that are inspired by the brain. Not exclusively, such approaches could include in-memory computing, continuous learning in hardware, spike-based processing, fine-grained parallelism, reduced precision computing, and asynchronous computing, among others. Many of these concepts are explored independently, not necessarily by linking them to neuromorphic technologies. For example, a comprehensive review of in-memory computing using emerging technologies is provided by Mannocci *et al.* in our first issue.²

There are distinct approaches to neuromorphic research, distinguished by their primary goals. One goal is to provide efficient hardware platforms to deepen our comprehension of biological nervous systems (e.g., the human brain). Conversely, a different objective is to harness brain-inspired principles to create efficient computing applications, without being confined by the need for biological realism. The distinct goals of brain modeling and efficient computing start to intersect when incorporating more biologically credible mechanisms, such as learning processes, which can potentially exceed the capabilities of existing machine learning and deep learning paradigms.³ In connection to this, it is of paramount importance to establish a standardized and unbiased method of evaluating the advantages and strengths of neuromorphic methods compared to conventional deep learning-based techniques.⁴

The neuromorphic community is incredibly diverse and spans across all stacks of computational abstractions. At the top layer, neuromorphic algorithms most commonly refer to spiking neural networks (SNNs)^{5,6} or training models via biologically plausible learning rules. What does it mean for a learning rule to be

biologically plausible? A common criterion is the need for locality in both space and time. While error backpropagation routes a huge number of gradient signals to each model parameter [Fig. 1(a)], the synapses in the brain are thought to only update on the basis of signals that are immediately available to it. For example, the work by McCaughan *et al.* published in this issue of *APL Machine Learning* derives a method to broadcast a global loss signal to all parameters, where each weight in a network selectively “extracts” the component relevant to itself [Fig. 1(b)]. While the technique is likened to wireless communication, it also bears similarities to global mechanisms in the brain, such as dopamine release. Fundamentally, the power savings it can have by reducing the amount of data routing are vast.

In the past several years, an emerging trend among the neuromorphic algorithms community is to hack apart error backpropagation to enable machine learning models to adapt and learn in real-time. Almost all these techniques stem from the original real-time recurrent learning training algorithm (RTRL) proposed back in 1989 by Williams and Zipser.⁸ When training recurrent models, gradient-based updates require storing the state of all neurons throughout all of history [Fig. 1(c)]. The memory demands become prohibitively expensive for training continuous models. The RTRL modifies this by pushing gradient signals forward in time instead, eliminating the need for a historical trace of each neuron [Fig. 1(d)]. Doing so means the cost of training SNNs is no longer limited by sequence length. The computational resources that once had to be reserved for lengthy sequences can be reallocated to scaling up the SNNs we train. Pushing this trend of large-scale SNNs was catapulted by the recent emergence of the first neuromorphic language model, SpikeGPT.⁹

Techniques such as e-prop approximate the RTRL by removing recurrent feedback connections.¹⁰ Deep continuous online learning, or *decolle*, applies local losses to each layer to address the

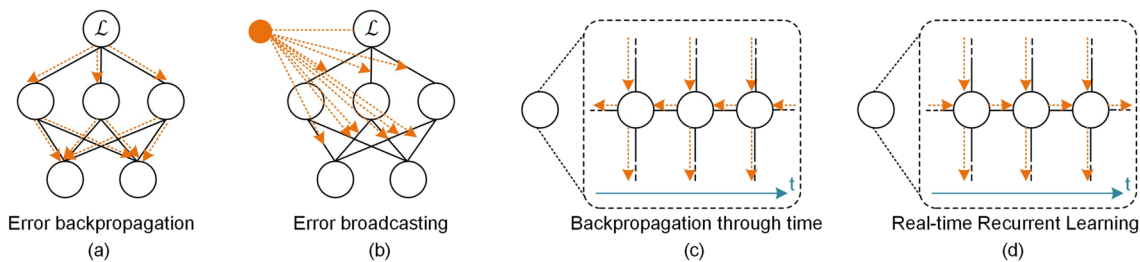


FIG. 1. Computational graphs of various modes of training. (a) Error backpropagation in a standard neural network. (b) Error broadcasting reduces the cost of gradient routing. (c) Backpropagation through time in sequential neural networks. (d) Real-time recurrent learning pushes gradients forward in time.

spatial locality of error signals in addition to temporal locality.¹¹ Forward propagation through time applies the RTRL to time-varying losses with regularization to promote training stability.¹² The general trend is to endow SNN learning with the benefits of the RTRL and stress-testing how tolerant these models are of approximations and alternative loss metrics.

While these are advances made at the software layer, the hardware and devices community directly reaps the benefits. Frenkel and Indiveri integrated forward-mode learning into a CMOS integrated circuit to show that SNNs can be trained at ultra-low power on a variety of dynamical tasks.¹³ This is the only silicon demonstration of a task-agnostic chip that can learn continuously on a range of data modalities. This range of high-performant lightweight learning algorithms signals a shift away from low-level learning rules, such as spike-time dependent plasticity, as we develop deeper insight as to how local learning rules become competitive with their non-local counterparts, especially when modulated by system-level objectives.

The absence of silicon chips in the online learning space is offset when one digs deeper into the abstraction layers. The past decade of neuromorphic devices and materials that have to adapt in real-time to input stimuli opens up a breadth of work that aims to link dynamics to functionality.^{14,15} The development of “neuromorphic devices” might be the closest to the AIPP readership. This involves leveraging the physical mechanisms of electronic, magnetic, and photonic materials to create efficient novel nanodevices that mimic biological functionalities such as those of neurons and synapses. The primary distinction from neuromorphic engineering is the objective to realize these functionalities through the inherent physics of materials instead of relying on individual devices like transistors and aiming for the most compact implementations (e.g., nanoscale devices). Multiple examples of these can be found across the AIPP journals, such as recent work on using memristive devices for neuromorphic applications,¹⁶ optoelectronic devices,^{17,18} magnetic devices,¹⁹ ferroelectric devices,²⁰ and photonic devices,²¹ as well as roadmaps and special issues.^{22–24} *APL Machine Learning* aspires to be the premier and preferred platform for all the aforementioned research. This falls under what we term as “Applied Physics for Machine Learning”—AP for ML, which is a central part of the journal’s scope.

Ultimately, neuromorphic computing is a natural stage for blending the various abstractions. The brain is a physical manifestation of the neural code, and building the most powerful computers will rely on researchers treating the link between physics and applications, and everything in between, in much the same way. If

the above resonates with you and your research, please consider submitting it to *APL Machine Learning*. More information about the journal’s scope can be found on our website, as well as in the first editorial.²⁵

REFERENCES

- C. Mead, “How we created neuromorphic engineering,” *Nat. Electron.* **3**, 434–435 (2020).
- P. Mannocci, M. Farronato, N. Lepri, L. Cattaneo, A. Glukhov, Z. Sun, and D. Ielmini, “In-memory computing with emerging memory devices: Status and outlook,” *APL Mach. Learn.* **1**(1), 010902 (2023).
- S. Schmidgall, J. Achterberg, T. Miconi, L. Krisch, R. Ziaie, S. Pardis Hajiseydrizi, and J. Eshraghian, “Brain-inspired learning in artificial neural networks: A review,” [arXiv:2305.11252](https://arxiv.org/abs/2305.11252) (2023).
- J. Yik *et al.*, “NeuroBench: Advanced neuromorphic computing through collaborative, fair and representative benchmarking,” [arXiv:2304.04640](https://arxiv.org/abs/2304.04640) (2023).
- J. K. Eshraghian *et al.*, “Training spiking neural networks using lessons from deep learning,” [arXiv:2109.12894](https://arxiv.org/abs/2109.12894) (2021).
- W. Maass, “Networks of spiking neurons: The third generation of neural network models,” *Neural Networks* **10**(9), 1659–1671 (1997).
- A. N. McCaughan, B. G. Oripov, N. Ganesh, S. W. Nam, A. Dienstfrey, and S. M. Buckley, “Multiplexed gradient descent: Fast online training of modern datasets on hardware neural networks without backpropagation,” *APL Mach. Learn.* **1**, 026118 (2023).
- R. J. Williams and D. Zipser, “Experimental analysis of the real-time recurrent learning algorithm,” *Connection Sci.* **1**(1), 87–111 (1989).
- R.-J. Zhu, Q. Zhao, and J. R. Eshraghian, “SpikeGPT: Generative pre-trained language model with spiking neural networks,” [arXiv:2302.13939](https://arxiv.org/abs/2302.13939) (2023).
- G. Bellec *et al.*, “A solution to the learning dilemma for recurrent networks of spiking neurons,” *Nat. Commun.* **11**(1), 3625 (2020).
- J. Kaiser, H. Mostafa, and E. Neftci, “Synaptic plasticity dynamics for deep continuous local learning (DECOLLE),” *Front. Neurosci.* **14**, 424 (2020).
- B. Yin, F. Corradi, and S. M. Bohté, “Accurate online training of dynamical spiking neural networks through Forward Propagation Through Time,” *Nat. Mach. Intell.* **5**, 518 (2023).
- C. Frenkel and G. Indiveri, “ReckOn: A 28nm sub-mm² task-agnostic spiking recurrent neural network processor enabling on-chip learning over second-long timescales,” in *2022 IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, 2022), Vol. 65.
- S. Kumar *et al.*, “Dynamical memristors for higher-complexity neuromorphic computing,” *Nat. Rev. Mater.* **7**(7), 575–591 (2022).
- A. Mehonic, A. Sebastian, B. Rajendran, O. Simeone, E. Vasilaki, and A. J. Kenyon, “Memristors—From in-memory computing, deep learning acceleration, and spiking neural networks to the future of neuromorphic and bio-inspired computing,” *Adv. Intell. Syst.* **2**(11), 2000085 (2020).
- Y. Huang, F. Kiani, F. Ye, and Q. Xia, “From memristive devices to neuromorphic systems,” *Appl. Phys. Lett.* **122**(11), 110501 (2023).

- ¹⁷X. Shan, C. Zhao, Y. Lin, J. Liu, X. Zhang, Y. Tao, C. Wang, X. Zhao, Z. Wang, H. Xu, and Y. Liu, "Optoelectronic synaptic device based on ZnO/HfO_x heterojunction for high-performance neuromorphic vision system," *Appl. Phys. Lett.* **121**(26), 263501 (2022).
- ¹⁸X. Dong, W. Wei, H. Sun, S. Li, J. Chen, J. Chen, X. Zhang, Y. Zhao, and Y. Li, "Neotype kuramite optoelectronic memristor for bio-synaptic plasticity simulations," *J. Chem. Phys.* **158**(18), 184702 (2023).
- ¹⁹W. Lv, J. Cai, H. Tu, L. Zhang, R. Li, Z. Yuan, G. Finocchio, S. Li, X. Sun, L. Bian, B. Zhang, R. Xiong, and Z. Zeng, "Stochastic artificial synapses based on nanoscale magnetic tunnel junction for neuromorphic applications," *Appl. Phys. Lett.* **121**(23), 232406 (2022).
- ²⁰X. Li, G. Li, Z. Zhang, W. Zhai, W. Zheng, L. Chen, L. Lin, X. Zhou, Z. Yan, and J.-M. Liu, "Implementation of habituation on single ferroelectric memristor," *Appl. Phys. Lett.* **122**(18), 183505 (2023).
- ²¹L. El Strouji, A. Krishnan, R. Ravichandran, Y. Lee, M. On, X. Xiao, and S. J. Ben Yoo, "Photonic and optoelectronic neuromorphic computing," *APL Photonics* **7**(5), 051101 (2022).
- ²²G. W. Burr, A. Sebastian, E. Vianello, R. Waser, and S. Parkin, "Emerging materials in neuromorphic computing: Guest editorial," *APL Mater.* **8**(1), 010401 (2020).
- ²³I. K. Schuller, A. Frano, R. C. Dynes, A. Hoffmann, B. Noheda, C. Schuman, A. Sebastian, and J. Shen, "Neuromorphic computing: Challenges from quantum materials to emergent connectivity," *Appl. Phys. Lett.* **120**(14), 140401 (2022).
- ²⁴T. Venkatesan and S. Williams, "Brain inspired electronics," *Appl. Phys. Rev.* **9**(1), 010401 (2022).
- ²⁵A. Mehonic, "Editorial: Welcome to APL Machine Learning," *APL Mach. Learn.* **1**(1), 010401 (2023).