Guest Editor Introduction: Neuromorphic Computing

Neuromorphic systems have the potential to transform the way we use computers and hold the possibility of breakthroughs in machine understanding and intelligence. Interest in the field is rapidly growing due to the looming end of the Moore's Law era, the end of Dennard scaling, and the explosion of interest in artificial intelligence. Significant research challenges in neuromorphic computers remain in defining the architecture building blocks and configurations of a system, the appropriate learning models, the development system and applications, and, of course, the development of neuromorphic hardware. Neuromorphic systems provide the opportunity to build highly efficient intelligent computing devices that can allow for a general reduction of power costs for computing and that can be deployed to allow for *in situ* machine learning at the edge. Neuromorphic computing systems have the potential to mimic the functionality of neural systems in the brain, which we believe will lead to more powerful and efficient computing paradigms. These new systems have the potential to help scientists pioneer new scientific discoveries or to help intelligently analyze sensor data for improved cyber security or energy management.

This special issue contains articles on a variety of neuromorphic implementations of neural network models and learning approaches, as well as a variety of hardware implementation types and applications of neuromorphic systems. As noted in the Department of Energy Advanced Scientific Computing Research's Workshop on Neuromorphic Computing: Architectures, Models, and Applications and reinforced in the articles presented in this special issue, the following issues remain unclear:

- The classes of models or problems that can be effectively implemented in neuromorphic systems (e.g., spiking or deep learning networks),
- The algorithms used for training these models (e.g., spike-timing dependent plasticity or gradient descent-based methods), and
- The applications that can best leverage these models and algorithms (e.g., traditional machine learning tasks such as speech recognition or non-machine learning approaches such as graph algorithms)

It is not clear that the community will converge on a common approach to models, algorithms, applications, or hardware implementation in the near future. This lack of convergence indicates that it may be time for a paradigm shift in the overall approach, particularly in the areas of learning and applications.

The wide adoption of deep learning approaches and the wide availability of deep learning datasets (e.g., MNIST, ImageNet) has influenced both the types of learning that have been widely used to date in neuromorphic systems (e.g., gradient descent-based methods) and the types of applications that have been widely examined (e.g., image classification). Though it is useful to demonstrate that neuromorphic systems are capable of deploying deep learning solutions on these types of applications, we believe that neuromorphic systems have a much greater potential in other, currently less well-defined, applications, which will require a paradigm shift in the way we approach both learning and applications. In particular, we suspect that the development of algorithmic approaches and learning methods that are native to neuromorphic systems is required in order to

2018 Copyright is held by the owner/author(s).

1550-4832/2018/11-ART39

<https://doi.org/10.1145/3283217>

realize their full potential. We believe that these new approaches should be inspired not only by biology but also by the underlying architectural features that are common to most neuromorphic systems. Deep learning methods, for example, limit neuromorphic systems to methods that are highly tuned to conventional computing hardware (CPUs or GPUs) and are in general a poor fit to the radically different architectures of neuromorphic systems. A better approach may be to take inspiration from both the biological and architectural structure of neuromorphic systems to develop new approaches for learning and training.

The convergence of the neuromorphic community to common approaches will require the development and maintenance of our own suite of datasets and challenge problems that are suited to neuromorphic solutions, and most likely not well addressed by other computing technologies. We expect that these “neuromorphic” datasets should be richly connected, temporal/longitudinal, multi-modal, and/or noisy. We also anticipate that the suite of neuromorphic challenge problems should include control and anomaly detection tasks, require supervised, unsupervised, and/or online learning approaches, and should include non-machine learning tasks as well (e.g., graph algorithm implementation).

Once again, the articles in this special issue demonstrate that there are a variety of approaches for neuromorphic system implementations, algorithms, and applications. The first article, “[Sparse Hardware Embedding of Spiking Neuron Systems for Community Detection](#)” by Hamilton et al., describes an approach for using spiking neurons to detect communities in graphs. The authors are able to demonstrate that mapping fully connected graphs to sparse embeddings does not reduce performance on this task, indicating that it is well suited to neuromorphic implementation. This article demonstrates a non-standard use case for spiking neuromorphic systems, beyond those inspired by machine learning or neural network approaches, which we anticipate will be a growing area of research in the coming years.

In the second article, “[Efficient Memristor-based Architecture for Intrusion Detection and High-Speed Packet Classification](#)” by Bontupalli et al., the authors describe a neuromorphic inspired-memristor crossbar for deep packet inspection, a cyber security application. This represents an application-specific use case inspired by neuromorphic hardware, a marked difference from the typical programmable neuromorphic architecture. However, using this custom hardware, the authors can achieve high throughput, high accuracy, and very low power, demonstrating that some applications may warrant custom implementations.

The third article, “[Power, Performance, and Area Benefit of Monolithic 3D ICs for On-Chip Deep Neural Networks Targeting Speech Recognition](#)” by Chang et al., presents a monolithic 3D IC implementation of deep neural networks that achieves power savings and accuracy improvements over implementations in 2D IC. Though the results were limited to a particular deep neural network structure, this article shows the promise for monolithic 3D IC implementations of neuromorphic systems moving forward.

The fourth article, “[Semi-Trained Memristive Crossbar Computing Engine with In-Situ Learning Accelerator](#)” by Ziyarah and Kudithipudi, presents a memristor-based neuromorphic implementation with on-chip learning. The article presents a compact, efficient on-chip learning circuit that achieves comparable accuracies with previous approaches, but more efficiently. This article sparks an interesting question of when on-chip learning should be implemented and when the tradeoffs in chip size, power efficiency, and accuracy are appropriate.

In the fifth article, “[STDP-based Unsupervised Feature Learning Using Convolution-Over-Time in Spiking Neural Networks for Energy-Efficient Neuromorphic Computing](#)” by Srinivasan et al., the authors demonstrate an unsupervised spike-timing dependent plasticity (STDP) approach for feature learning in images. They are able to show that the resulting convolutional networks require less training images than the fully connected networks and are more energy-efficient in

spintronic neuromorphic systems. This article offers a compelling combination of convolutional and STDP-trained networks that have multiple benefits, including requiring less storage, less training examples, and less computation in a neuromorphic system.

The sixth and seventh articles both address reservoir computing-style implementations, though with two radically different hardware implementations. In “[DFR: An Energy-Efficient Analog Delay Feedback Reservoir Computing System for Brain-Inspired Computing](#),” Yi et al. demonstrate the first known analog implementation integrated circuit for delayed feedback reservoir. In “[An FPGA Implementation of a Time Delay Reservoir Using Stochastic Logic](#),” Loomis et al. demonstrate an FPGA-based implementation for time delay reservoir. These two works demonstrate some of the diversity of hardware implementations in the field of neuromorphic systems, spanning from custom fabricated systems to off-the-shelf commodities.

Finally, the eighth and ninth articles both address FPGA-based cellular neural network implementations. In “[A Multi-Level Optimization Framework for FPGA-Based Cellular Neural Network Implementation](#),” Liu et al. focus on optimizations in the complete workflow for FPGA-based implementations and achieve significant speedup on image segmentation tasks, while in “[Efficient Hardware Implementation of Cellular Neural Networks with Incremental Quantization and Early Exit](#),” Xu et al. focus on incremental quantization of weights to achieve significant speedup and also on image segmentation tasks.

Our goal for the special issue is to provide a window for the greater emerging technologies in computing community into not only some of the ongoing work in the field but also introduce some of the fundamental ongoing questions for the community. We hope to inspire both neuromorphic and non-neuromorphic researchers alike to come together in helping the community pursue the paradigm shift that is required to demonstrate the full capabilities of neuromorphic systems.