An Interview with Katie Schuman: On the Future of Neuromorphic Computing

INNOVATION LEADERS

by Bushra Anjum

Editor's Introduction

In this series of interviews with innovation leaders, Ubiquity Associate Editor and software engineer, Dr. Bushra Anjum, sits down with Katie Schuman to discuss computers inspired by biological neural systems as an alternate promising architecture.

An Interview with Katie Schuman: On the Future of Neuromorphic Computing

INNOVATION LEADERS

by Bushra Anjum

Catherine (Katie) Schuman is a research scientist in the Computational Data Analytics group at Oak Ridge National Laboratory (ORNL). Katie was born and raised in a small town in East Tennessee. She attended the University of Tennessee (UT), where she received her B.S. in computer science and mathematics in 2010 and her Ph.D. in computer science in 2015. Her passion for nature-inspired machine learning techniques, such as neural networks and genetic algorithms, led her to the study of neuromorphic computing as a graduate student at UT. She co-founded the TENNLab neuromorphic computing research team there before graduating and moving to ORNL. She is continuing her research of neuromorphic computing at Oak Ridge and still co-leads the TENNLab research group at UT as a joint faculty member. Katie has co-authored more than 35 publications in the field of neuromorphic computing and is the co-inventor on five neuromorphic patents. She can be found on Twitter @cdschuman.

Bushra Anjum (BA): What is your big concern about the future of computing to which you are dedicating yourself?

Katie Schuman (KS): My biggest concern with the future of computing is how the computing community will deal with the looming end of Moore's Law and the end of Dennard scaling. Moore's law said that the number of transistors on a chip doubles roughly every two years, and what the concept of Moore's law gave us is increased speeds in computing from year to year, due to improvements in hardware. Dennard scaling is a scaling law, which was tied to Moore's law and said that the amount of power consumed by a computer chip was a consequence of the size of the chip rather than the number of transistors. Moore's law depended on Dennard scaling, and because Dennard scaling effectively ended around 2006, it has affected the computing performance improvements that we see today. That is, we do not see the types of increases in computation speed that we enjoyed under the reign of Moore's law and Dennard scaling. To deal with this, today's computer architects are looking to different architectures that specialize in specific application workloads to allow for continued improvements to

computing speed, but the incorporation of new computer architectures and technologies is not easy.

My research centers around the neuromorphic computer; a computer whose architecture and functionality are inspired by biological neural systems. Neuromorphic computer systems are massively parallel computer systems where the computation and memory units are collocated. They are typically event-driven and asynchronous. Neuromorphic computers are targeted primarily towards implementing neural networks efficiently in hardware. Since neural networks underlie much of our machine learning capabilities today, neuromorphic computers can be used in everyday devices, such as mobile phones, to allow for more efficient computing.

Bringing neuromorphic systems into production will require tremendous investment [1]. One of the biggest challenges is the development of new programming models and languages for these machines. The computational thinking we have learned is based on the von Neumann architecture, which features programs stepping through instruction sequences, and transferring data between computational units and memory. Neuromorphic computing requires a different kind of thinking that does not rely on instruction sequences and instead relies on training or learning processes for programming the system. Neuromorphic systems also need new methods for measuring and assessing performance. The increased performance of neuromorphic computers comes at the cost of specialization; they are not meant to be general-purpose computers [2]. However, there is recent work, including some by my colleagues and I, suggesting that these computers can be usefully applied to problems other than machine learning [3].

Why has it taken so long for these non-neural network use cases to emerge? I believe that it is because a new, unfamiliar kind of computational thinking is required to program these systems. Adoption of these new technologies will require a paradigm shift in how we think about computing as a whole, and paradigm shifts are often uncomfortable and challenging. Though this is a difficult task, continued innovation in computing depends on our willingness to move beyond our traditional von Neumann systems.

BA: How did you become interested in computers, and how did that interest lead to the field of neuromorphic computing?

KS: My dad was a middle school math teacher with an interest in computing and electrical engineering. Every few years, he would get the latest computer for our family so that he could keep up with and utilize computing in the classroom, so I don't remember a time in my life where my house didn't have a computer. I got to use the computers, too! My parents invested in educational games for me. Over time, I was able to track the progression of computing technology as I could see how the games I was playing became more complex and sophisticated over time; I could even see how the same game's performance would differ because of a newer, faster computer.

Though computer science courses weren't readily available during my secondary education, I loved other STEM topics: math, biology, and chemistry. When it was time to choose a major in college, computer science was an obvious choice for me. In my later years of undergraduate and graduate studies, studying machine learning (in particular neural networks and genetic algorithms) allowed me to combine my excitement for computing with my love of math and biology—particularly neuroscience and genomics. This combination led me to my dissertation work where I developed a simplified spiking neural network model. A spiking neural network is a neural network model that takes even more inspiration from the biological brain than the traditional model. There is an explicit notion of time in spiking neural networks; that is, when a neuron fires in spiking neural networks, it can take varying amounts of time for the output signal to travel along synapses to reach other neurons. During my dissertation, I demonstrated that the algorithm and model that I had developed could be applied to classification, control, and anomaly detection problems successfully.

As I had grown up watching computers advance in performance, it was easier for me to notice that there had been a plateauing of the performance improvements in new computers [1]. I was introduced to the field of neuromorphic computing by Mark Dean, a professor at the University of Tennessee, during my graduate studies. Dr. Dean, along with my advisor Dr. Doug Birdwell, had recognized the consequences of the looming end of Moore's law and Dennard scaling, and [they] were interested in pursuing the development of new computer architectures and hardware. Together, we developed a computer architecture and hardware system based on my simplified spiking neural network model, which became our first foray into neuromorphic systems.

As I learned more about the field of neuromorphic computing, I realized that it is dominated by hardware developers, neuroscientists, and machine learning researchers, but not many software developers and engineers. We do not have enough computer scientists working towards developing the appropriate system software and applications for these types of systems. There is a need for the development of general-purpose software tools for new architectures such as neuromorphic computers to become usable by the computing community at large. Just like the widespread use of GPUs was enabled by the development of software libraries and APIs, new architectures like neuromorphic computers will need its own set of software systems, APIs, and so on, to make them accessible to new users.

BA: What efforts are you currently leading to make the world of neuromorphic computing more appealing to the computing community as a whole and more inviting to newcomers?

KS: I am currently co-leading a research effort between Oak Ridge National Laboratory and the University of Tennessee to develop common algorithms and system software tools for neuromorphic systems. We work closely with materials scientists and device engineers to learn how to use new neuromorphic implementations effectively. We also collaborate with applications developers on building new neuromorphic software implementations, with a focus on ease of use from the application development perspective. The project is primarily centered on using existing and developing new machine learning techniques to build the “programs” for neuromorphic systems automatically. As part of this project, we have developed the TENNLab software framework, which is a common application programming interface for spiking neuromorphic computing systems [4], as well as EONS, a software system that uses genetic algorithms to assemble spiking neural networks for particular applications [5]. Both of these frameworks are meant to ease the use of neuromorphic computing systems for new users. Our team is also working on developing non-machine learning algorithms and applications for neuromorphic systems. In this case, instead of treating it as a neural network accelerator, we approach the neuromorphic architecture as a computer architecture with specific characteristics, such as massively parallel computation, simple computational units, collocated processing, and memory, etc. that we then exploit to perform certain computations [6].

In addition to my research efforts, with a group of colleagues from across several institutions, I co-organize a conference on neuromorphic systems, the International Conference on Neuromorphic Systems or ICONS, to rally the community to build innovative and usable neuromorphic systems. We are aiming to engage the computing community as a whole with the necessary software and algorithmic support from the neuromorphic computing community to enable these systems to reach their full potential. We encourage researchers who are interested in learning more about neuromorphic computing to engage with our community by joining us at ICONS. ICONS 2019 will be the second year of the conference, following two years of neuromorphic workshops.

I hope that my work will make it easier to use neuromorphic computers, especially for researchers who aren't yet familiar with the field. I believe that the next big leap in computer performance and algorithmic development will come from new architectures such as neuromorphic computers, and my goal is to help enable those innovations to occur. We always welcome collaborations. If you are interested in investigating the use of neuromorphic computers for your applications or in integrating an existing neuromorphic hardware system into our software infrastructure, please do contact me for more information.

References

- [1] Vetter, J. S. et al. Extreme heterogeneity 2018-productive computational science in the era of extreme heterogeneity. Report for DOE ASCR Workshop on Extreme Heterogeneity. USDOE Office of Science. United States. 2018.
- [2] Schuman, C. D. et al. A survey of neuromorphic computing and neural networks in hardware. arXiv preprint arXiv:1705.06963 (2017).
- [3] Aimone, J. B. et al. Non-neural network applications for spiking neuromorphic hardware. Extended abstract. In *Proceedings of the Third International Workshop on Post Moore's Era Supercomputing* (PMES '18); <https://sites.google.com/view/pmes18/>
- [4] Plank, J. et al. The TENNLab exploratory neuromorphic computing framework. *IEEE Letters of the Computer Society* 1, 2 (2018).

- [5] Schuman, C. D. et al. An evolutionary optimization framework for neural networks and neuromorphic architectures. *Neural Networks (IJCNN)*, 2016 International Joint Conference on. IEEE, 2016.
- [6] Schuman, C. D. et al. Shortest path and neighborhood subgraph extraction on a spiking memristive neuromorphic implementation. Extended abstract. In *Proceedings of the Third International Workshop on Post Moore's Era Supercomputing* (PMES '18); <https://sites.google.com/view/pmes18/>

Biography

Bushra Anjum is a software technical lead at Amazon in San Luis Obispo, CA. She has expertise in Agile Software Development for large scale distributed services with special emphasis on scalability and fault tolerance. Originally a Fulbright scholar from Pakistan, Dr. Anjum has international teaching and mentoring experience and has served in academia for over five years before joining the industry. In 2016, she has been selected as an inaugural member of the [ACM Future of Computing Academy](#), a new initiative created by ACM to support and foster the next generation of computing professionals. Dr. Anjum is a keen enthusiast of promoting diversity in the STEM fields and is a mentor and a regular speaker for such. She received her Ph.D. in computer science at the North Carolina State University (NCSU) in 2012 for her doctoral thesis on Bandwidth Allocation under End-to-End Percentile Delay Bounds. She can be found on Twitter [@DrBushraAnjum](#).

DOI: 10.1145/3322097