Neuromorphic Engineering: Neuromimetic Computation for Understanding the Brain

By Mostafa Rahimi Azghadi, Giacomo Indiveri, and Derek Abbott

Neuromorphic engineering attempts to understand the computational properties of neural processing systems by building electronic circuits and systems that emulate the principles of computation in the neural systems. The electronic systems that are developed in this process can serve both engineering and life sciences in various ways ranging from low-power brain-like computing embedded systems to neural-based control, brain machine interfaces, and neuroprosthesis. To realize such systems, various approaches and strategies with their own advantages and limitations, may be adopted. Here, we provide a summary of our recent article published in the proceedings of the IEEE [1], where we have discussed and reviewed the various approaches to the design and implementation of neuromorphic learning systems, and pointed out challenges and opportunities in these systems.

Hardware or Software Approach: That is the Question.

Perhaps, one of the most important steps required to understand and build a neural processing system is to understand its basic components: the neural cells (neurons) and their synapses. By interconnecting large numbers of these components, and adopting strategies for adapting their network structure and properties (e.g. via plasticity mechanisms), biological neural systems are able to express complex learning and signal processing properties. Similar to their biological counterpart, the neuromorphic emulations of these neural components are typically interconnected in Very Large Scale Integration (VLSI) chips and systems, to form artificial neural processing systems.

Over the last few decades, based on their experiments and observations, neuroscientists postulated various models of neural dynamics and synaptic plasticity to explain the learning and computation phenomena in the brain. When implemented in software, the model neural architecture, neurons, synapses, and their learning mechanisms are simulated on conventional Von Neumann computing architectures. The main advantages of this approach are, shorter design/exploration time compared to the physical implementation of neuromorphic systems, as well as the flexibility and reconfigurability they offer compared to the non-reconfigurable design in most of physical design approaches. On the other hand, software-based approaches have very high power consumption and require bulky computing systems, which severely limit their scaling possibilities. A representative example of such systems is the simulation of a large-scale spiking neural network run on the IBM's Blue-Gene supercomputer that takes 147456 CPUs, 144 TB of memory, which consumes more than 5 MW of power [2].

In contrast to the software approach, neuromorphic electronic systems can provide a full-custom dedicated hardware solution that uses low-power hybrid analog/digital circuits in a highly parallel manner, therefore allowing the emulation of large scale neural processing systems on biological, or even in accelerated time scales [1, 3].

Neuromorphic Electronic Learning Systems

There exists a variety of theoretical models and their corresponding hardware implementations, for neurons and synapses [4, 5]. These models and their relevant physical implementations mimic the functionality and properties of neural components with various degrees of details.

For instance, neuromorphic engineers have implemented various neuron models, ranging from simple Integrate and Fire (IF) neurons to detailed Hodgkin-Huxley (HH) neuron [4]. An IF neuron integrates the input currents, produced by the synapses stimulated by its afferent neurons in the network, and fires a spike once the integrated current exceeds a threshold. This feature can be effectively mimicked using electronic circuits. More elaborate IF neuron circuits directly emulate the properties of the voltage dependent conductances in real neurons, incorporating mechanisms that can give rise to useful functional properties, such as adaptive threshold, spike-frequency adaptation, and refractory period. Figure 1 shows the response of a silicon neuron of this type, to a constant input current stimulus.

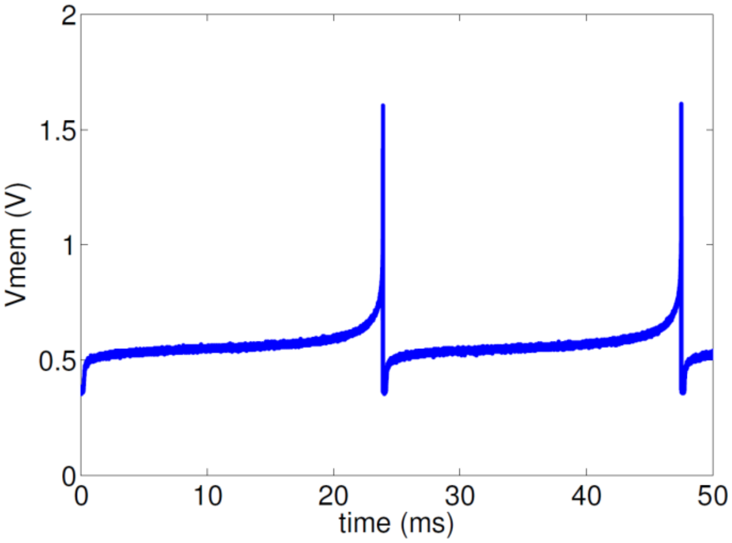

Figure 1: *A silicon neuron generates spikes in response to input stimulation. The vertical axis shows the neuron potential, V_{mem}, changing over time.*

In addition to neurons and synapses, neuromorphic designers have implemented various learning circuits for changing the efficacy of the synaptic circuits, effectively building synaptic plasticity models in silicon [1], [6]. *Spike Timing Dependent Plasticity* (STDP) is one of the most recognized learning mechanisms which has been implemented and utilized in various theoretical models and neuromorphic systems. The change in the synaptic weights in STDP depends on the exact timing of the spikes arriving in a neuron as well as those emitted by the neuron. Besides STDP and its many variants, other learning rules that modify the synaptic efficacy based on other neural parameters have been implemented, such as learning circuits that are affected by currents and voltages representing Calcium concentration in the synapse, and the neuron’s membrane potential[1].

Over the past years, we have developed a number of neuromorphic learning architectures in VLSI technology. These architectures utilize various learning mechanisms, ranging from the classical STDP protocol to triplet STDP [6], and Spike-Driven Synaptic Plasticity (SDSP) [1]. We have shown that these learning networks, which are composed of silicon neurons and synapses equipped with a chosen learning circuit, are capable of not only reproducing the outcomes of various biological experiments [6], but also have promising performance in cognitive task [3, 7]. In addition to a number of full-custom VLSI devices, which are only built to implement a specific learning rule such as STDP or SDSP, we implemented a hybrid hardware-software neuromorphic system that can be interfaced to conventional computers and programmed to implement various spike-based learning algorithms while still doing neural computation in a massively parallel way with silicon neuron circuits [8]. This provides us with flexibility in implementing and studying various synaptic plasticity algorithms, compared to fully hardware neuromorphic designs.

The VLSI technology used for implementing these neural systems results in low-power and real-time structures for these networks, which are convenient for implementing embedded systems and real-time behaving systems that interact with the environment [1], [7]. These neuromorphic systems receive input signals encoded as spike trains, and through their neurons and synapses learn to process the input signals, i.e. modify their weights accordingly. After the learning phase is over, the input signals to the system, which may be extracted from neuromorphic sensors such as silicon retina or cochlea, are processed online and the system generates appropriate output spiking signals in response to specific signals it has learned during the learning phase.

Challenges in Implementing Neuromorphic Learning Systems

As already mentioned, to realize artificial neural networks that can faithfully reproduce the properties of biological neural networks to facilitate the study of the brain and, at the same time, be useful for implementing neural computation, there are different approaches and strategies including those reviewed here, with their own advantages and limitations. In [1], we have discussed and reviewed the design, implementation, applications and challenges of spike-based neuromorphic learning systems. The reader is encouraged to refer to that article to grasp the general idea of neuromorphic learning systems and their limitations and benefits.

Acknowledgment

The authors would like to thank the Neuromorphic Cognitive System group members, at the Institute of Neuroinformatics, University and ETH Zurich, where most of the aforementioned VLSI neuromorphic systems were developed.

For Further Reading

1. **M. Rahimi Azghadi**, et al., "Spike-Based Synaptic Plasticity in Silicon: Design, Implementation, Application, and Challenges.", *Proceedings of the IEEE*, 2014. 102(5): p. 717-737.

2. **R. Ananthanarayanan**, et al. "The cat is out of the bag: cortical simulations with 10^9 neurons, 10^{13} synapses.", In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*. 2009, DOI: 10.1145/1654059.1654124.

3. **E. Chicca**, et al., "Neuromorphic Electronic Circuits for Building Autonomous Cognitive Systems.", *Proceedings of the IEEE*, 2014. DOI: 10.1109/JPROC.2014.2313954.

4. **G. Indiveri**, et al., "Neuromorphic silicon neuron circuits.", *Frontiers in Neuroscience*, 2011. 5:73.

5. **C. Bartolozzi** and **G. Indiveri**, "Synaptic dynamics in analog VLSI.", *Neural Computation*, 2007. 19(10): p. 2581-2603.

6. **M. Rahimi Azghadi**, et al., "A neuromorphic VLSI design for spike timing and rate based synaptic plasticity.", *Neural Networks*, 2013. 45: p. 70-82.

7. **E. Neftci,** et al., "Synthesizing cognition in neuromorphic electronic systems.", *Proceedings of the National Academy of Sciences*, 2013. 110(37): p. E3468-E3476.

8. **S. Moradi** and **G. Indiveri**, "An Event-Based Neural Network Architecture With an Asynchronous Programmable Synaptic Memory.", *IEEE Transactions on Biomedical Circuits and Systems*, 2014. 8(1): p. 98-107.