GOMACTech Tutorial on Neuromorphic Computing

Organized by Cliff Lau, Barry Treloar, Gerry Borsuk, Christal Gordon, and Michael Fritze

Neuromorphic computing refers to computational paradigms that are inspired by the way the human brain processes information and thus are intended to be similar to the neuro-biological architectures present in the nervous system. Neuromorphic engineering, which includes neuromorphic computing, was a concept developed by Carver Mead in the late 1980s, describing the use of very-large-scale integration (VLSI) systems containing electronic analog circuits to mimic the signal processing in the brain, although modeling of neuronal computation goes back to the 1940s. In the beginning, neuromorphic systems were single chip devices that emulate peripheral sensory transduction such as silicon retina and silicon cochlea. These artificial neural systems have demonstrated amazing performance in image and speech processing. Gradually the emulation moved further up to the central nervous system such as the olfactory cortex and visual cortex. Many neuromorphic systems have been implemented by software programs in conventional digital computers and applied to a multitude of problems including speech and image recognition. In the last decade, implementation of neuromorphic systems has included the building of artificial brains.

The way the brain computes is very different from conventional digital computers which are based on the von Neumann architecture with the fetch, compute, and store paradigm with arithmetic logic unit and memory units. The brain on the other hand consists of neurons and synapses that connect the neurons together, and the computation and memory are distributed and integrated throughout the brain. While the computational algorithms and information representations are largely unknown, it is clear that instead of binary Boolean logic and precise digital synchronous operations, the brain and central nervous system uses sparse distributed representations, massively parallel mechanisms, extensive adaptations and self-organization and learning. How the brain achieve intelligence is not yet completely understood. But by building computing machine that is similar to the brain, it is hoped that neuromorphic computers would achieve a certain level of intelligence.

Neuromorphic computing is all about neurons, synapses, learning, and memory. The most common model of a neuron is summing amplifier or integrate and fire neuron. Synapse serves to interconnect the neurons together and also serve as storage memory. Most common learning algorithm is error back propagation and steepest descend weight adaptation. In this tutorial course, the attendees will learn what neuromorphic computing is all about, and will be able to apply it to problems such as image processing, object recognition, speech recognition, decision making, machine learning, and autonomous systems. The attendee will learn a short history of neuromorphic computing, various ways of connecting the neurons and synapses, computational architectures and learning algorithms including Deep Learning, application such as object recognition, and some of the research challenges. The attendee will learn about the several state-of-the-art neuromorphic systems, both the hardware and software. Finally the attendee will have an opportunity to experiment and program with these state-of-the-art neuromorphic systems.

GOMACTech Tutorial on Neuromorphic Computing
Monday, March 12, 2018
Hyatt Regency Miami, FL

0730-0800	Registration
0800-0815	Brief history of neuromorphic computing Dr. Clifford Lau, IDA
0815-0915	Survey of neuromorphic computing and neural networks in hardware Dr. Catherine Schuman, ORNL
0915-1015	Hardware implementations Prof. Nathaniel Cady, SUNY Polytechnic Institute
1015-1030	Break
1030-1130	Design and programming methodology Prof. James Plank, Univ. of Tennessee
1130-1230	Roadmap to achieve large neuromorphic hardware systems Prof. Jennifer Hasler, Georgia Tech
1230-1315	Lunch
1315-1415	Deep learning Dr. Wilfried Haensch, IBM Research Yorktown Heights NY
1415-1515	DesignWare EV6x embedded vision processor with DL and CNN for ADAS application Gordon Cooper, Synopsys
1515-1530	Break
1530-1630	Intelligent machines Dr. Winfried Wilcke, IBM Research Almaden CA
1630-1730	Efficient machine learning inference in the cloud Andy Walsh, Xilinx

Abstracts

1. Brief history of neuromorphic computing

Dr. Clifford Lau

Neuromorphic computing (NC) refers to computational architectures and algorithms that are inspired by the way human brain processes information to solve problems and make decisions. Modeling of neuronal computation goes way back to the 1940s. In 1943 McCulloch and Pitts showed that neurons can be modeled as a simple threshold device to perform logic function. By the late 1950s and early 1960s, neuron models were further refined into Rosenblatt's Perceptron and Widrow and Hoff's Adaptive Linear Neuron (Adaline). During the 1970s, Steven Grossberg at Boston University and Teuvo Kohonen at Helsinki University were making significant contributions. Grossberg, together with Gail Carpenter, had developed a model architecture they called adaptive resonance theory (ART) based on the idea that the brain spontaneously organized itself into recognition codes. In the 1980s, neuronal modeling was given an impetus when John Hopfield published a paper in the Proceedings of the National Academy of Sciences followed by another paper in Science. That led to the explosive research growth in artificial neural networks (ANN), including the forever popular Hopfield Nets and Multilayer Perceptrons (MLP). Advances in the very large scale integrated circuits (VLSI) technology ushered in the field of neuromorphic engineering (a term coined by Carver Mead) in the mid-1980s to reflect that the engineered electronic systems are designed to emulate the computational capabilities of the brain and the network of neurons and synapses. Carver Mead, together with a large number of prominent scientists (Max Delbruck, John Hopfield, Richard Feynman, Christof Koch, Terry Sejnowski, Rodney Douglas, Andreas Androu, Paul Mueller, and others), made convincing argument that neuromorphic circuits are ideal for implementing the computational principles exhibited in the brain. Today, the most popular ANN, with application in image recognition, is Deep Learning (DL), which is basically an MLP with lots of layers and millions of synaptic weights, and Convolutional Neural Net (CNN).

2. Survey of neuromorphic computing and neural networks in hardware

Dr. Catherine Schuman

Neuromorphic computing has come to refer to a variety of brain-inspired computers, devices, and models that contrast the pervasive von Neumann computer architecture. This biologically inspired approach has created highly connected synthetic neurons and synapses that can be used to model neuroscience theories as well as solve challenging machine learning problems. The promise of the technology is to create a brainlike ability to learn and adapt, but the technical challenges are significant, starting with an accurate neuroscience model of how the brain works, to finding materials and engineering breakthroughs to build devices to support these models, to creating a programming framework so the systems can learn, to creating applications with brain-like capabilities. In this work, we provide a comprehensive survey of the research and motivations for neuromorphic computing over its history. We begin with a 35-year review of the motivations and drivers of neuromorphic computing, then look at the major research areas of the field, which we define as neuro-inspired models, algorithms and learning approaches, hardware and devices, supporting systems, and finally applications. We conclude with a broad discussion

on the major research topics that need to be addressed in the coming years to see the promise of neuromorphic computing fulfilled. The goals of this work are to provide an exhaustive review of the research conducted in neuromorphic computing since the inception of the term, and to motivate further work by illuminating gaps in the field where new research is needed.

3. Hardware implementations
Prof. Nathaniel Cady

Neuromorphic computing systems seek to emulate biological neural functionality emulated in either software or electrical hardware. A key function for such systems is their ability to learn and adapt. In the human brain, such learning and adaptation is achieved via modulation of synaptic connections between different neurons. My research group has focused on the implementation of non-volatile memory elements (primarily memristors) for synaptic functionality in hardware-based neuromorphic circuits. Memristors, which can be implemented as resistive random access memory (RRAM) are a novel form of non-volatile memory expected to replace a variety of current memory technologies and enabling the design of new circuit architectures. Investigations of ReRAM as a storage technology have shown a combination of high storage density with fast access and write speeds. Recently, the endurance and reliability of ReRAM cells have reached the level at which they are competing with commercially available Flash memory and CMOS technologies, making ReRAM a viable candidate for data storage and novel logic and security architectures.

In this presentation, I will review multiple approaches for integrating non-volatile memory elements (such as memristors) with CMOS, to yield functional neuromorphic circuits. In addition, I will explore the multi-level / analog behavior of some classes of memristors, which can be utilized for high density memory storage and for setting a range of synaptic weight values per individual (or pairs) of memristive devices.

4. Design and programming methodology
Prof. James Plank

Adapting applications to leverage neuromorphic devices is a challenging task. This is both from the application perspective and the device perspective. From the application perspective, application state must be transformed effectively into neuromorphic input, and neuromorphic output must be interpreted effectively by the application. From the design perspective, neuromorphic devices must be "programmed" to control or solve an application. This talk will focus on application design for neuromorphic computing, and on neuromorphic learning/programming techniques. It will not include deep learning, since that will be the topic of another tutorial talk. It will include genetic algorithms, spike timing-dependent plasticity (STDP), and reservoir computing.

5. Roadmap to achieve large neuromorphic hardware systems
Prof. Jennifer Hasler

Neuromorphic systems are gaining increasing importance in an era where CMOS digital computing techniques are reaching physical limits. These silicon systems mimic extremely

energy efficient neural computing structures, potentially both for solving engineering applications as well as understanding neural computation. Toward this end, the authors provide a glimpse at what the technology evolution roadmap looks like for these systems so that Neuromorphic engineers may gain the same benefit of anticipation and foresight that IC designers gained from Moore's law many years ago. Scaling of energy efficiency, performance, and size will be discussed as well as how the implementation and application space of Neuromorphic systems are expected to evolve over time.

6. Deep learning

Dr. Wilfried Haensch

The recent success of machine learning and deep learning networks in image recognition, speech and language processing stress computational resources. Training of these very large networks with millions of parameters (weights) can take weeks on current hardware. Inherent in these workloads is that the underlying algorithms are noise tolerant and amenable to low precision computation. Custom hardware is created to take advantage of this situation. Reduced precision digital solutions are emerging first for inferencing and are expected to tackle training as well soon. The question at hand is: once the avenue of reduced precision has reached its end what is the next step to address performance bottlenecks?

Performance in deep learning applications is determined by two factors: (1) computational efficiency, that is performing the mathematical operation that are required, and (2) bringing the relevant data, the weights on which the computation is performed, from the memory to the compute unit. Reduced precision will address both, however it will never eliminate movement of data.

One intriguing solution to the problem of weight related data movement is to capture the weights in arrays of nonvolatile memory and to perform the required computations locally on these arrays. It turns out, however, that existing memory materials are ill suited for this problem. Memory materials are optimized for a few reproducible bit states, whereas their application in deep learning networks requires almost analog switching behavior with a reversible response to pulse stimulation of opposite polarity.

To take advantage of this new architecture two possible solution can be pursued: (1) adjust the control circuitry to accommodate the imperfect switching behavior of current available materials, like PCM or RRAM; and (2) find a suitable material and architecture that cam maximize the benefit. Initial estimates show that a more than 10,000 times speed increase could be feasible at significantly reduced power if the architecture can take advantage of an optimized switching material. Enhancement factors of this magnitude will open the possibility of local learning on mobile devices or intelligent devices at the edge of the network.

7. DesignWare EV6x embedded vision processor with DL and CNN for ADAS application

Gordon Cooper, Synopsys

The technological demands at the heart of embedded vision applications, in the neural network, require solutions that deliver the combination of high precision and performance with low power and area use. The unique combination of the vector DSPs and programmable CNN engine in the DesignWare EV6x Vision Processor enables developers to implement vision functionality in their embedded devices with much higher performance efficiency than CPU- and GPU-based alternatives.

The DesignWare EV6x Processor family integrates scalar, vector DSP and CNN processing units for highly accurate and fast vision processing. The EV6x supports any convolutional neural network, including popular networks such as AlexNet, VGG16, GoogLeNet, Yolo, Faster R-CNN, SqueezeNet and ResNet. Designers can run CNN graphs originally trained for 32-bit floating point hardware on the EV6x's 12-bit CNN engine, significantly reducing the power and area of their designs while maintaining the same levels of detection accuracy. The engine delivers power efficiency of up to 2,000 GMACs/sec/W when implemented in 16-nm FinFET process technologies (worst-case conditions). The EV6x's CNN hardware also supports neural networks trained for 8-bit precision to take advantage of the lower memory bandwidth and power requirements of these graph types.

To simplify software application development, the EV6x processors are supported by a comprehensive suite of tools and software. The latest release of the DesignWare ARC® MetaWare EV Development Toolkit includes a CNN mapping tool that analyzes neural networks trained using popular frameworks like Caffe and Tensorflow, and automatically generates the executable for the programmable CNN engine. For maximum flexibility and future-proofing, the tool can also distribute computations between the vision CPU and CNN resources to support new and emerging neural network algorithms as well as customer-specific CNN layers. Combined with software development tools based on OpenVX™, OpenCV and OpenCL C embedded vision standards, the MetaWare EV Development Toolkit offers a full suite of tools needed to accelerate embedded software development.

8. Intelligent machines

Dr. Winfried Wilcke

While Deep Learning networks have made huge strides in image recognition, speech processing and similar pattern recognition tasks, this is only the beginning on the long path to creating truly intelligent machines, also called general or strong artificial intelligence.

Current deep learning networks are very superficially inspired by the brain in that they consist of layers of neurons connected with synapses of varying weights, but that's where the similarity ends. The fundamental operation of (most) artificial neural networks is based on supervised training, where the network receives some known and human labeled input ("this is a cat"), then compares the current output of the network with the desired output and tweaks the values of synapses until the difference (error) is minimized. Mathematically this corresponds to minimizing an error function in a very high dimensional space by tools like stochastic gradient descent. We can be certain that this is NOT at all how the brain works. A symptom is that today's neural network may need tens of thousands of cat images to learn to recognize cats, whereas a

child may need to be told only a few times that this is a cat. Moreover, humans learn continuously and new knowledge usually doesn't damage prior knowledge, whereas today's neural networks (except for reinforcement learning) need to have a clear separation between training phase and execution (or inference) phase and they are very brittle when trying to add new knowledge.

Machines will only become intelligent in the human sense if they develop 'common sense' and reasoning which is the ultimate challenge for general intelligence. A common sense statement like "Clouds pay no taxes" is obvious to us, but a machine needs to learn a huge amount of facts about the world to even have a concept of taxes, clouds and any relationship between them (none in this case). This requires an intelligent machine - like a child - to autonomously develop a detailed model of the world and the relations between the elements in this world.

It is likely that elements of such world models can form autonomously in the brain or an intelligent machine based on a specific mathematical concept (hierarchical sparse distributed representations), where sensory inputs form an invariant hierarchy of ever more complex model elements. However, this bottom up approach will take a long time to bring to fruition, so a hybrid approach where some preprocessing is done with conventional networks is currently being developed.

The potential of this approach has been demonstrated by building several two-legged robots which learned on their own - without explicit programming - how to walk without falling down. A new type of neural supercomputer (Escape 9000) is being built by IBM to accelerate the research into the algorithms underlying intelligent machines. We will shortly discuss a possible wafer-scale implantation of Escape 9000 called Shannon.

9. **Efficient machine learning inference acceleration in the cloud**
   Andy Walsh, Xilinx

This demo will allow users to get a look at Xilinx’s machine learning software stack using FPGA F1 instances on the Amazon EC2 Public cloud. The demo will highlight accelerated image classification using a modern neural network model such as ResNet50. It’s implemented through the open source frameworks, such as Caffe and accelerated with the Xilinx Deep Neural Network library to be fully optimized, delivering the highest compute efficiency for 8 bit inference.