SAND2016-12503J

A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications

Conrad D. James[a,*], James B. Aimone[a], Nadine E. Miner[a], Craig M. Vineyard[a], Frederick H. Rothganger[a], Kristofor D. Carlson[a], Samuel A. Mulder[a], Timothy J. Draelos[a], Aleksandra Faust[b], Matthew J. Marinella[a], John H. Naegle[a], Steven J. Plimpton[a]

Affiliations:
[a] Sandia National Laboratories, Albuquerque, NM USA
b Author is now at Google X, Palo Alto, CA USA

***Correspondence:**
Dr. Conrad D. James
Sandia National Laboratories
P.O. Box 5800, Mailstop 1425
Albuquerque, NM 87185 USA
cdjame@sandia.gov

Keywords: neuromorphic computing, algorithms, artificial neural networks, data-driven computing, machine learning, pattern recognition

Abstract

Biological neural networks continue to inspire new developments in algorithms and microelectronic hardware to solve challenging data processing and classification problems. Here, we survey the history of neural-inspired and neuromorphic computing in order to examine the complex and intertwined trajectories of the mathematical theory and hardware developed in this field. Early research focused on adapting existing hardware to emulate the pattern recognition capabilities of living organisms. Contributions from psychologists, mathematicians, engineers, neuroscientists, and others were crucial to maturing the field from narrowly-tailored demonstrations to more generalizable systems capable of addressing difficult problem classes such as object detection and speech recognition. Algorithms that leverage fundamental principles found in neuroscience such as hierarchical structure, temporal integration, and robustness to error have been developed, and some of these approaches are achieving world-leading performance on particular data classification tasks. In addition, novel microelectronic hardware is being developed to perform logic and to serve as memory in neuromorphic computing systems with optimized system integration and improved energy efficiency. Key to such advancements was the incorporation of new discoveries in neuroscience research, the transition away from strict structural replication and towards the functional replication of neural systems, and the use of mathematical theory frameworks to guide algorithm and hardware developments.

Introduction

The mammalian brain has been the subject of scientific inquiry for decades, largely due its unique computational capabilities and its inherent ability to adapt and learn within a modest power budget (<50W). Many attempts to emulate the characteristics of biological neural networks have been made, especially in the microelectronics field where specialized brain-inspired hardware is being developed to fabricate "smart" systems (Kumar 2013). However, limitations in our understanding of how biological neural networks function have hindered the ability of engineered systems to solve challenging problems. Discovering the mechanisms of biological neural system functionality is crucial for the next generation of electronic hardware to meet the data science and "big data" demands of the 21st century. For instance, decades of research and billions of dollars have been invested in various forms of pattern recognition, and while substantial improvements have been made, synthetic electronic systems still cannot approach the abilities of human perception on particular problems (Gelly et al., 2012; Borji and Itti, 2014). This may be due in part to the primary focus on replicating the cortex for most neuromorphic and neural-inspired systems, whereas a more comprehensive approach that incorporates the modulatory role of other brain regions (striatum, etc.) might provide new breakthroughs.

A major challenge to harnessing the mammalian brain's computational capabilities is the lack of detailed understanding of its operating principles. Despite those limitations, neuroscience and psychology research have provided a strong foundation for the development of mathematical algorithms such as artificial neural networks (ANNs) and machine learning (Figure 1– INSERT FIGURE 1 HERE). Early work by psychologists led to theories on learning while the field of neuroscience has brought insight into how individual neurons may represent and process information via the development of tools such as the patch clamp technique (Neher et al., 1978). Other technologies such as *in vivo* electrodes have been crucial to neuroscience discoveries, including the activity of place cells and their impact on our understanding of how neural systems may use timing to encode information (O'Keefe, 1976; O'Keefe and Recce, 1993). Recently, neuroscientists have begun to appreciate the representational capacity of populations of neurons - a shift made possible by advances in large-scale recording technologies that permit simultaneous monitoring of thousands of neurons (Stevenson and Kording, 2011). Churchland et al.'s (2012) work with multi-electrode recordings highlighted the importance of considering dynamics in population coding, specifically the role of oscillatory-like neural activity for preparing and conducting physical activities such as arm movement. Other technology advances in techniques such as live brain imaging have improved the correlation of regional brain activity to particular computational tasks (Villringer and Chance, 1997; Price, 2012). On the other end of the scaling spectrum, advances in molecular-level investigations of neural circuitry have also shaped our understanding of the role played by different cell types in the brain (He et al., 2012; Hu et al., 2014). A major challenge for the neuroscience field is the difficulty in making the connection between neural activity and function across scales. High performance computing resources have been leveraged to use information theory to understand how individual cell-based phenomena such as adult neurogenesis can impact the overall computational capability of a large network (Vineyard et al., 2016). New initiatives at U.S. federal agencies are bridging this gap between the molecular biology of individual neurons and cognitive functions (Cepelewicz, 2016), and the Brain Research through Advancing Innovative Neurotechnologies (BRAIN) initiative is focused on developing new tools for such measurements (Insel et al., 2013). The European Human Brain Project

(HBP) is organized around the idea of improving our understanding of the brain, and also has neuromorphic computing as a major thread of research (Calimera et al., 2013). Considerations for the scaling of neuromorphic systems indicate the difficulty in emulating biological neural systems under the constraints of both mature and newly developed hardware (Hasler and Marr, 2013). With new technology to address these scientific questions, new theories of neural computation should be forthcoming and thus aid the development of neural-inspired algorithms and hardware systems to address existing challenges in data processing and analysis. The history of neuromorphic computing is complex (Boahen, 2005; Hammerstrom, 2010; Indiveri et al., 2011; Schmidhuber, 2015), and the purpose of this review is to highlight the important contributions made to the field by researchers who leveraged new discoveries in neuroscience, generated approaches aimed at functional replication of neural systems, and developed rigorous mathematical analyses of algorithms and hardware systems.

Historical development of data-driven computing

The early 20th century witnessed many advances in neuroscience and psychology, including developments in theories around learning, information representation, and neuroanatomy. Psychologists and neuroscientists at the time were among the earliest researchers to explore ideas in regard to viewing neurobiological organisms as templates for developing computational systems. Together, the fields of neuroscience and psychology led to the rise of data-driven computing methods in the form of ANNs and machine learning (Figure 1). Data-driven computing - in contrast to numerical computing which relies on the construction of closed-form equations and explicit programming – relies on the processing of example data to produce generalized models for analyzing new data and/or mapping data to new representations. This branch of computing uses data processing algorithms that mimic the anatomy of neural systems with layers of computing units (neurons) spanned by massive numbers of connections between computing units. For the purposes of our discussion here, we refer to the mimicry of neurobiological anatomy/morphology for computing as "neuromorphic computing" in contrast to methods such as machine learning which can be characterized as "neural-inspired computing" in that the algorithms are driven by high-level abstract concepts of human cognition such as decision-making and reinforcement-based learning. Within machine learning, two sub-branches emerged with statistical machine learning focusing on static problems and dynamic machine learning focusing on problems where the time domain needs to be included. With this suite of algorithmic developments, hardware systems were developed to simulate biological neural systems and to implement neuromorphic and neural-inspired systems for addressing particular application areas. We acknowledge that the distinction between the described branches of computing in Figure 1 as well as the attribution of different algorithms to particular branches can be debated; however, the objective of this survey is to examine large cross-cutting themes that span the algorithms and hardware implementations that have been developed in this field over the decades. This provides some degree of historical context to the technology development in neural-inspired and neuromorphic computing, and will help generate new ideas and directions for the field to pursue in the future.

Neural modeling and simulation

By providing insight into how neurobiological systems compute, neural modeling and simulation platforms hold great promise for supporting the development of neuromorphic and neural-inspired algorithms and hardware. Simulations of neural tissue have been conducted many years, starting with the small pattern-recognition learning network simulated by Farley and Clark (1954, 1955) using an IBM 704 digital computer. Even at the time of these early simulations, the limitations of using conventional off-the-shelf hardware were readily apparent, particularly in regard to scaling and density (10^{11} neurons and 10^{15} synaptic connections in ~1000 cm^3) as well as the separation of memory and processing. Simulations of biological neural systems have advanced in conjunction with the advances in microelectronics and computational hardware. The first large-scale brain simulation effort in Europe, the Blue Brain Project, was largely focused on supercomputer simulations with high performance computing resources (Markram, 2006). Subsequent work from this project demonstrated a detailed simulation of cortical circuitry by integrating multiple sources of experimental data (Markram et al., 2015). Additional groups have leveraged similar resources to simulate neural tissue, including a thalamus-cortex model to reconstruct functional magnetic resonance imaging signals (Izhikevich and Edelman, 2008), and a 10^9 neuron/10^{13} synapse cortical system with simulated EEG signals (Ananthanarayanan et al., 2009). The Semantic Pointer Architecture Unified Network (Spaun) was a large-scale (25 million neurons) computational model of multiple human brain regions capable of performing tasks such as image recognition and sequence recall (Eliasmith et al., 2012; Stewart and Eliasmith, 2014). This neural model leveraged the Neural Engineering Framework (NEF) approach wherein representations of information were mapped into the spatiotemporal domain with "spiking" neural networks and synaptic connections between neurons were used to approximate mathematical operations (Eliasmith and Anderson, 2003). Spiking neural networks (SNNs) are neural models that capture essential aspects of neural operation, such as spike dynamics, synaptic conductance, and plasticity while leaving out less central features such as axonal voltage propagation and spatial processing due to dendritic computations. These models represent a compromise between simulation run-time and biological fidelity which makes them well-suited for large-scale neural simulations and for the development of energy-efficient, fault-tolerant neuromorphic hardware devices. Due to the parallel nature of neural computation, a number of research groups have implemented parallel versions of SNN simulators for use on supercomputing clusters (Gewaltig et al. 2007), graphics processing units (GPUs) (Beyeler et al., 2015; Nowotny, 2010), and even specialized neuromorphic chips (Esser et al., 2013; Thomas et al., 2013). One example of a highly parallelized SNN simulator is CARLsim, an open source C/C++ based SNN simulator that allows for the execution of spiking neuron models with realistic spike dynamics on both generic x86 CPUs and standard off-the-shelf NVIDIA GPUs (Beyeler et al., 2015). The parallelized GPU implementation of CARLsim was written to optimize four key performance metrics: parallelism, thread divergence, memory bandwidth, and memory usage (Nageswaran et al., 2009). CARLsim uses a number of approaches to achieve high performance on GPUs such as using a hybrid neuron/synapse-parallelism scheme, performing data buffering to reduce thread divergence, and utilizing sparse representation techniques such as address event representation to reduce memory and bandwidth usage. CARLsim distinguishes itself from other simulation platforms by providing a number of important features together in a single software package. These features include platform compatibility (Linux, Mac OS X, and Windows), a test suite for code verification, rigorous code documentation, a MATLAB toolbox for visualization of neuronal and synaptic information, support for several spike-based synaptic plasticity mechanisms, and a network-level parameter tuning framework (Carlson et al., 2014).

Early neuromorphic algorithms and hardware systems

Biological neural systems have long served as an inspiration for developing new algorithms or engineering hardware systems to perform particular tasks. The earliest neuromorphic and neural-inspired systems replicated large-scale mechanisms observed in biological organisms such as reflex movements and maze-finding. And due to the limited knowledge of how neurobiological systems functioned, these systems were largely phenomenological. As the neuroscience field matured and more detailed knowledge of neural tissue functionality was discovered, researchers were able to improve the specificity and complexity of neural-inspired hardware. Many of the neural-inspired algorithms and hardware developed in the first half of the 20th century stemmed from research in both neuroscience and psychology (Figure 1). Psychologists H.D. Baernstein and C.L. Hull (1931) developed a model hardware system to replicate conditioned reflexes using a battery powered system made of push buttons (sensory organs), electrochemical cells (memory storage), thermoregulatory switches (synapses), and copper wire (nerves) (Dalakov 2016). A similar biomimicry hardware system developed several decades later was the homeostat (Ashby, 1960). Designed to emulate the homeostatic properties of biological organisms, this electromechanical system contained several control units with variables that were continually compared against target values. Input into the system that caused changes in the variables triggered internal feedback that restored the variables back towards their targets and thus stabilized the system. In addition to biological functions such as reflexes, researchers also developed maze-solving neuromorphic hardware (Ross, 1933; Bradner Jr, 1937). These systems largely relied on classical conditioning via trial-and-error exploration, with failed paths being retained and avoided on subsequent trials. Later, maze-navigating systems such as the Theseus magnetic mouse developed by Claude L. Shannon (1951) leveraged existing hardware such as telephone relay circuits and mechanical motors to enable the trial-and-error navigation of user-defined mazes.

A significant disadvantage for many of these early neuromorphic systems is that they lacked formalized algorithmic guidance and relied largely on empirically-observed phenomena. As such, large-scale behaviors (e.g. reflexes and maze-finding) could be modeled phenomenologically with trial-and-error, but only under strictly defined conditions meaning the systems lacked the adaptive properties exhibited by biological organisms. As the fields of neuroscience and psychology advanced, more detailed and algorithm-directed demonstrations of biological functions in neuromorphic hardware were developed. One of the earliest examples of the development of a theoretical framework for neural-inspired algorithms occurred in 1943 when Warren E. McCulloch, a neurophysiologist, worked with Walter H. Pitts, a self-trained logician, to develop the McCulloch-Pitts neuron model (1943). This model was the first step for ANN research by incorporating several neuroscience principles, including neuron spiking, limited temporal summation of inputs, and inhibitory and excitatory connections within networks. Also discussed by McCulloch and Pitts was the phenomenon of learning, which at the time they felt could "require the possibility of permanent alterations in the structure of nets" via changes in the excitation threshold of neurons (McCulloch and Pitts, 1943). While the McCulloch-Pitts neuron was an important development, a mechanism for learning was not fully explored until work pioneered by the psychologist Donald O. Hebb (1949). Hebb's rule of connected cells firing in concert to "induce lasting cellular changes" postulated a basic mechanism for synaptic plasticity that was later demonstrated *in vitro* in biological neurons (Dan and Poo, 2004). This Hebbian learning principle along with the mathematics of McCulloch-Pitts neurons were part of the inspiration behind Marvin Minsky's Stochastic Neural Analog Reinforcement Calculator (SNARC), a vacuum-tube based hardware system capable of simulating "rat-in-a-maze" type problems (Minsky, 1952). The machine's "synapses" were initiated with random values, but the weight probabilities changed over the course of the system operation based on the correctness of each path choice selected while navigating the maze.

The selection of maze-finding as an application for the earliest neuromorphic system was to be expected given that it represents one of the simplest classes of problems with well-defined and static constraints and boundaries. More challenging problems such as recognizing patterns within noisy data require more sophisticated algorithm and hardware development. The Perceptron, invented by Frank Rosenblatt (1958, 1960), was one of the first algorithms to be drawn from neuroscience ideas with regard to individual neurons and how they were perceived to process information. The concept behind the Perceptron was to use thresholding integrators (neurons) to act on a set of inputs with connections of variable strength (synapses). After training the Perceptron on labeled data, new unlabeled data input into the system is linearly separated into different classes. Initially simulated on an IBM computer, the Perceptron was eventually built in custom hardware known as the Mark I Perceptron, a 3-layer classifier that could learn visual patterns (Hay et al., 1960). The Mark I Perceptron was built using a 20x20 array of semiconductor photodiodes as the sense layer, an association layer with fixed weights connected to the sense layer, and a response layer with variable weights in the form of motor-adjusted potentiometers connected to the association layer (Tappert, 2011). This work represented a substantial shift away from traditional neural-mimicry and towards leveraging mathematical formulations to guide the assembly of specialized hardware. Later developments included multilayer perceptrons (Rosenblatt, 1962); however, concerns about the applicability of perceptrons to data that is not linearly separable led to reduced interest in Perceptron-based algorithms (Minsky and Papert, 1969). In this same timeframe, Bernard Widrow and Ted Hoff (1960a) developed the least-mean-squares algorithm, a simplified method to estimate gradients and minimize the error between an input and target vector during training procedures. The algorithm was implemented in a hardware system called ADALINE (Adaptive Linear Neuron) which relied on potentiometers and switches to demonstrate learning. Widrow later developed a three-terminal electrochemical resistor with memory device (termed a "memistor") to take the place of large potentiometers and to improve the resolution of changes in the synaptic weights (Widrow, 1960b). In addition to several hardware differences with the Perceptron, the ADALINE system sent weights directly between layers instead of thresholding weighted sums of inputs. Later developments by Winter and Widrow (1988) included a second iteration termed MADALINE which consisted of "many" ADALINE elements and was capable of handling classification problems in which the data was not linearly separable, which as mentioned earlier was a primary disadvantage of the Perceptron.

Advances in neuroscience inspire developments in neuromorphic algorithms and hardware

The algorithmic framework provided by McCulloch, Pitts, Hebb, Widrow, Rosenblatt and others laid a strong foundation for future decades of neural-inspired algorithms theory and hardware development driven by real-world applications. One of the first practical application drivers was pattern recognition, a term defined as “the extraction of the significant features from a background of irrelevant detail” by mathematician O.G. Selfridge (1955). Around this time, pattern recognition gained popularity amongst experimental psychologists and mathematicians (French, 1954; Dinneen, 1955; Fitts et al., 1956). In these examples, the focus was on understanding how visual patterns such as written characters and shapes within noisy backgrounds were detected. Whereas the work described earlier such as the Perceptron, the SNARC system, and other maze-navigating hardware were designed for pattern recognition applications, they were motivated by non-specific generalized concepts found in biological neural systems. The next generation of pattern recognition neuromorphic systems were more directly motivated by neuroscience research on specific neural systems such as the studies performed by neurophysiologists David Hubel and Torsten Wiesel (1959) on the V1 region of the mammalian visual cortex. Overall, Hubel and Wiesel’s studies supplemented earlier work that cast sensory regions that correspond to activity in a particular neuron (receptive fields) as “feature detectors” (Barlow, 1953). Although the concept of receptive fields had been around for some time, Hubel and Wiesel’s studies provided a new level of detail in regard to the selectivity of individual neurons to particular shapes and shape orientations. In addition, their work highlighted the importance of combined excitatory and inhibitory regions within fields to produce selectivity to particular stimuli, to improve contrast, and to aid in the perception of movement. The first algorithm designed to mimic visual perception using a hierarchical cascading network structure was the Cognitron and subsequently the Neocognitron developed by Kunihiko Fukushima (1975; 1988). Building off neuroscience work on individual neuron representations in the visual system, this learning algorithm was demonstrated to be resilient to noise, changes in positon, and geometrical distortion, which naturally led this approach to be used to detect 2D patterns in image data such as handwritten digits. The Neocognitron is an example of an unsupervised learning algorithm wherein the data is not labeled and classification accuracy is determined after the data is processed. On the other hand, supervised learning methods are used in cases where the data is labeled beforehand, and test data are evaluated in comparison to ground truth labeled data. A significant neural-inspired aspect of the Neocognitron design was the specialization of different “cells” within the network: receptor “cells” that receive the input data, “S-cells” which act as feature extractors from the raw data, “C-cells” which receive fixed connections from S-cells and allow for variations in stimuli to impact the network consistently, and “V-cells” which act as inhibitory cells to help confer relevance to extracted features. This specialization of processing components within the Neocognitron represented a major departure from previous neural-inspired algorithms which relied on large numbers of identical processors in massively parallelized structures to garner computational advantages. It also served as an example of the shift away from simple structural replication to a focus on the operational functionality of neural systems. Later, a digital VLSI hardware implementation of the Neocognitron algorithm was demonstrated on a character recognition problem with an improved and more noise-robust recognition rate (White and Elmasry, 1992). Although the Neocognitron contains both excitatory and inhibitory connections within its hierarchical network structure, the lack of recurrent connections limits its use on time-series data. Eventually, the blossoming electronics industry led to the development of very large scale integrated (VLSI) circuit hardware systems for emulating the retina portion of the visual system (Mead and Mahowald, 1988). In this system, complementary metal oxide semiconductor (CMOS) transistors were operated in the analog regime to create a 48x48 pixel artificial retina with biologically-relevant properties such as edge sensitivity and spatio-temporal filtering. The VLSI silicon retina developed by Delbruck (1993) used correlation-based computation to produce 2D "direction selective" outputs for detecting motion in video while consuming only 5 μW per pixel. The neuromorphic retina fabricated by Kameda and Yagi (2003) improved upon the design and imaging capabilities of such systems by mimicking both the sustained and transient responses of ganglion cells in the vertebrate retina. This provided the system with the capability to "perceive" both static and dynamic images whereas previous artificial retinas only replicated one of those functionalities. The system also incorporated compensating circuitry to reduce noise in captured image frames caused by voltage mismatches in subcomponents. Okuno et al. (2015) recently developed an emulator for replicating the imaging capabilities of a biological visual system. Using a VLSI silicon retina and additional hardware, a complex assortment of cell types such as amacrine cells and bipolar cells were incorporated into the emulator to generate graded potentials and perform visual system computations for detecting static and dynamic objects.

In addition to the visual system, the auditory system of biological organisms has also been a subject of interest for the neuromorphic computing community. Lyon & Mead (1988) developed an analog microelectronic cochlea by modeling the ear as a multi-stage frequency filter with active gain for rapid adaptation. The cochlea chip contained transconductance amplifiers used in subthreshold mode as active switching devices and in threshold mode as capacitors. An important demonstration from this system was the property of "scale invariance," a phenomenon that has been measured in biological cochleas wherein the output signal remains unchanged at different points throughout the cascaded structure of the system (Talmadge et al., 1998). However, the original silicon cochlea system was sensitive to many design parameters such as mismatches in transistor characteristics, and a new system designed to address these issues resulted in a larger and more complex circuit (Watts et al., 1992; Douglas et al., 1995). Although balancing power efficiency, functionality, and design complexity within these systems is difficult, Chicca et al. (2014) recently highlighted approaches to mitigate the circuit complexity of neuromorphic systems while maintaining computational functionality.

Resurgence in artificial neural network and neuromorphic computing research

As mentioned previously, the limitations of Perceptron and related algorithmic approaches led to a decline in the neural-inspired computing field for many years, but over time, researchers developed new neural-inspired and neuromorphic algorithms. J.J. Hopfield (1982, 1984) introduced a single-layer neural network for recognizing patterns that had distinct differences from earlier Perceptron-based networks. In contrast, to feed-forward Perceptron networks where all connections are directed from input neurons to output neurons, Hopfield Networks contain cyclic recurrent couplings that provide feedback from output neurons back to input neurons. This type of recurrent neural network (RNN) architecture is observed in biological neural systems such as the hippocampus, and Hopfield networks have been used for data clustering (Maetschke and Ragan, 2014) and data restoration (Paik and Katsaggelos, 1992). Fusi et al. (2000) developed a RNN in VLSI hardware containing excitatory and inhibitory neurons with memory storage in plastic synapses, and subsequently this technology was matured to demonstrate Hebbian-based learning with 56 plastic synapses on a 0.6 μm CMOS chip (Chicca et al., 2003). One of the main limitations of Hopfield-type networks is the limited storage capacity of memorized patterns, calculated by Amit et al. (1987) for a Hopfield network of N neurons to be 0.138N. However, the ability of Hopfield nets to store memories garnered interest for their use in associative memory applications where a memory bank is addressed via its contents. Atencia et al. (2007) implemented a Hopfield network on a Xilinx FPGA and demonstrated that the hardware was capable of representing parameters in a differential equation model at 24 bits of precision while saving significant computation/power compared to a floating point representation.

In addition to the development of new ANN algorithms that were more neural-inspired (e.g. Hopfield networks), another major breakthrough helped lead to a resurgence in neural network research with the rediscovery and use of the backpropagation technique (LeCun, 1985; Rumelhart et al., 1986, Werbos, 1990). Backpropagation is a principled way to formulate weight training as a gradient descent problem. Such approaches have been explored since the 1960s and allow for the error between a network's output values and the supervised ground truth to be propagated back through the entire network (Kelley, 1960; Bryson and Denham, 1962). This translates the error into a gradient distributed to each weight in the network via application of the chain rule, thus enabling the efficient use of multi-layered neural networks on pattern recognition problems. Backpropagation enabled the training of hidden layers in neural networks, thus beginning the progression toward modern multi-layered neural network techniques. Other error minimization techniques including the "feedforward-feedback" method described by Achler (2014) have also been developed to improve the ability of neural network algorithms to handle symbolic data. An example of a neuromorphic hardware system that used backpropagation was the system fabricated by Jackel et al. (1990) for handwritten digit classification in 0.9 μm CMOS, producing a chip with 32,000 reconfigurable synapses that could be evaluated in parallel at a rate of 3×10^{11} connections/s. The algorithm relied on hand-selected kernels to extract features and different techniques such as windowing and backpropagation for digit classification.

With the development of new algorithms, specialized hardware, and techniques for training neural networks, new types of problems other than static classification of objects became of interest. Dynamic problems such as tracking objects in video feeds and parsing speech have become the dominant focus of much of the research in the field. Atlas et al. (1988) implemented an early application of neural networks in the time domain in order to extract and classify phonemes from speech data. To apply neural networks to such time-varying data, the mathematics of the system were altered to have multiplication steps converted to convolutions and weights converted to transfer functions. Another type of neural networks that have been used in applications wherein the data varies in the spatial and time domains are Convolutional Neural Networks (CNNs) (LeCun et al., 1989; 1998; Serrano-Gotarredona et al, 2015). The NeuFlow system was developed for hierarchical visual data processing and relies on CNNs implemented on an FPGA board (Farabet et al., 2011). The system was used to label objects within outdoor street images at a rate of 12 frames/s and operating with a performance-power metric of approximately 14.7×10^9 operations/s/W (as compared to 0.04×10^9 operations/s/W using a CPU). A challenge with the NeuFlow system is the use of look-up tables which have limited accuracy for calculations but are useful for rapid reprogramming of the system when new functionality is required.

Continued interest in handling time-domain data eventually lead to new neural-inspired algorithms such as reservoir computing (Jaeger, 2001). In reservoir computing, the reservoir consists of a random recurrent network of neurons that perform nonlinear computations on input data that converts data into a set of complex states. The reservoir maps the input data from a low dimensional data space into a higher dimensional feature space where separability of the data is improved (Verstraeten et al., 2007). This approach is helpful in simulating complex nonlinear processes for which closed-form analytical models are not available. Two independently-developed examples of reservoir computing are echo state networks (Jaeger and Haas, 2004) and liquid state machines (Maass et al., 2002). Echo state networks are machine-learning-centric systems based on analog sigmoidal non-spiking neurons, whereas liquid state machines are more neurobiology-centric systems with leaky integrate-and-fire spiking neurons (Verstraeten et al., 2007). The reliance of liquid state machines on RNN architectures as “basic computational units” (Maass et al., 2002) indicates some degree of influence by the neuroscience concept of temporal coding (Figure 1). Reservoir computing approaches have been used in pattern classification, speech recognition, and control systems. Recently, specialized hardware has been developed to implement reservoir computing using opto-electronic systems to generate the reservoirs (Schürmann et al., 2004; Paquot et al., 2012; Vandoorne et al., 2014). In the system described by Vandoorne et al., the photonics-based reservoir is comprised of a set of optical components (e.g. waveguides) that fit within a 16 mm^2 chip that could perform digital operations such as Boolean logic and analog operations such as speech recognition. In addition, the flexible time-scale architecture and the use of coherent light increased the number of possible states that were represented in the reservoir, while the elimination of amplifiers from the system design prevented power consumption from occurring within the reservoir.

Modern developments in neuromorphic computing algorithms and hardware

Neuromorphic computing research eventually matured beyond sensory systems such as vision and hearing and into simulating and leveraging concepts from cognitive brain regions such as the cortex. This required a more substantive examination of the microarchitecture of neural tissue and of modern microelectronics in order to understand the differences in information processing between the systems. An important element of neuromorphic systems is the distinction between traditional von Neumann architectures used in modern computers (separated memory, computation, and control) and biological neural network architectures where these three components are integrated together. The energy efficiency observed in neural systems can be attributed to this component-level integration, but also to the massive parallelism and hierarchical structure of neural tissue. Non-von Neumann hardware has been developed to improve the energy efficiency of neuromorphic systems. Neftci et al. (2013) developed a system to simulate the visual tracking of objects. This work relied on a finite state machine approach to map a behavioral model of this task (including contextual cues) onto a spiking integrate-and-fire network. Another example of a non-von Neumann architecture is the Neurogrid, a specialized hardware platform developed at Stanford University to simulate large networks of biological neurons (Boahen, 2006; Benjamin et al., 2014). Inspired by the microarchitecture of the cerebral cortex, the Neurogrid was an analog system of transistors operated at a subthreshold state and configured into silicon-based neurons, axons, dendrites, and synapses to simulate neural systems in real time with dramatically reduced power consumption as compared to conventional digital hardware. Another effort, the European Union Human Brain Project (HBP), was also initiated with a focus on brain simulation and specialized hardware fabrication (Markram, 2012). One of the hardware development components of the project, named the SpiNNaker project, used a parallelized communications architecture for high-volume transmission of small data packets for fixed-point-based computations (Furber et al., 2014). The system was comprised of processing nodes, each of which contained 18 ARM968 processor cores with local and shared memory. An individual core was capable of simulating hundreds of neurons each with thousands of synaptic connections and this system has been used to characterize learning algorithms and to process sensor data in robotic systems. The strength of the SpiNNaker project is that the architecture provides a platform wherein proposed neural algorithms can be explored with parametric studies, thus enabling such neuromorphic hardware to be used to test and eventually influence our understanding of how biological networks function. Recently, the SpiNNaker hardware was coupled with a silicon retina to demonstrate a neuromorphic vision system that used high temporal precision graded potential and spike-based signaling and also contained circuitry for cortex-to-retina feedback (Kawasetsu et al., 2014). Another neuromorphic simulation effort connected to the HBP was the FACETS (Fast Analog Computing with Emergent Transient States) project led by Heidelberg University (Schemmel et al., 2010). This project focused on performing *in vitro* and *in vivo* studies in animal models to generate single cell and network data to improve computational neuroscience models and facilitate new neuromorphic chip designs (http://facets.kip.uni-heidelberg.de/). Hardware was implemented in 180nm CMOS VLSI technology, and the team developed the software language PyNN to simplify the user interface. As shown in the FACETS program, the standardization of the interface to neuromorphic systems and between computational neural models is crucial to promoting the use of neuromorphic hardware, algorithms, and models throughout the broader research community and to generating useful comparisons between different platforms. Additional neural model interchange standards and tools that provide capabilities such as file read-in and translation include NeuroML (Gleeson et al., 2010), Nengo (Bekolay et al., 2014), PyNCS (Stefanini et al., 2014), and N2A (Rothganger et al. 2014). A follow-up project to FACETS was the BrainScaleS program started in 2011 (https://brainscales.kip.uni-heidelberg.de). Subsequent to the FACETS program, the BrainScaleS effort focused on leveraging biological data that spanned multiple spatial and temporal scales from individual synapses to macroscopic networks of neurons in order to produce neural models and hardware with improved functionality. This program has also worked to develop novel algorithm ideas to address conventional numerical computing problems such as solving differential equations.

Industry has also developed an interest in non-von Neumann architectures for computing applications. The CM1K chip from CogniMem (Cognimem Technologies, Inc. 2013) was related to the IBM ZISC036 technology (Eide et al., 1994) and Intel Corporation's radial basis function (RBF) effort (Holler et al., 1992). The CM1K chip was a fully parallel chip with 1024 silicon neurons that used either a RBF or K-nearest neighbor non-linear classifier to learn patterns up to 256 bytes. This chip has been used in several pattern recognition applications such as target tracking in unmanned aerial vehicle videos (Yang et al., 2014) and network intrusion detection (Payer et al., 2014). A neural-inspired architecture called the Golden Gate chip was developed by IBM under the DARPA Systems of Neuromorphic Adaptive Plastic Scalable Electronics (SyNAPSE) program (Merolla et al., 2011).

This chip employed a non-von Neumann architecture with a clock-less digital design to couple computation and memory to achieve low operational power consumption (~45 pJ per spike). Fabricated in IBM's 45nm process, the chip consisted of 256 digital neurons and over 260,000 binary synapses and was demonstrated with a probabilistic restricted Boltzmann machine (RBM)-based neural network algorithm to process image data for digit recognition. An important finding from this effort was that the use of binary values for weights did not significantly reduce the system's digit classification performance. TrueNorth is the most recent version of this IBM chip architecture, and it consists of 4 Golden Gate core chips to yield 1 million neurons and over 250 million programmable synapses (Merolla et al., 2014). In this study, the TrueNorth chip was used to recognize disparate objects in video feeds in real-time, with a large reduction in power consumption over traditional hardware under ideal conditions (400×10^9 synaptic operations/watt for TrueNorth compared to 4.5×10^9 floating-point operations/watt for a supercomputer). The absence of on-chip learning in the TrueNorth platform is a limitation, however, a similar effort from the SyNAPSE program that included on-chip learning was the microelectronic neuron and synapse architecture developed by HRL Laboratories (Cruz-Albrecht et al., 2012). This system used a low-power architecture in 90 nm CMOS technology for a phenomenological representation of synaptic plasticity-based learning and demonstrated an energy/spike power budget of 0.4 pJ. One of the major debates within the neuromorphic computing community is the degree of biological fidelity that should be replicated in hardware given the tradeoffs between biological accuracy and application performance (Krichmar et al., 2015). On-chip learning in neuromorphic systems serves as a good example of the appropriate pursuit of biological replication in that data communication costs (in terms of energy) are reduced and data processing speeds are improved (theoretically). However, the specifics of how to incorporate neurobiological plasticity into hardware remains a subject of research given the increased system complexity required for on-chip learning and the difficulty in translating biological mechanisms into microelectronic components. Phenomenological models of plasticity have been developed including a model that used a combination of spike-timing and spike-rate-based learning mechanisms in VLSI hardware (Rahimi Azghadi et al., 2013). Mitra et al. (2009) demonstrated the use of a similar model on a pattern matching application. On the other side of the modeling spectrum, Rachmuth et al. (2011) developed a detailed biophysical model of spike-based plasticity in VLSI, emulating down to the level of ion channels and membrane receptors. Qiao et al. (2015) recently developed the Reconfigurable On-line Learning Spiking (ROLLS) neuromorphic architecture for biophysical emulations of neural systems and used the platform to classify objects from the Caltech 101 database. This system indicated that the design criteria for neural simulation-focused hardware does not preclude the use of such a system for practical applications.

A major theme in modern approaches towards neuromorphic computing is the development of hierarchical representations of data. The concept is to generate low-level features (such as phonemes in speech or edges in images) that can be combined and transformed mathematically to reconstruct more complex features such as phrases or objects of interest, respectively. The structural hierarchy observed in biological neural circuitry provides a degree of flexibility to these tissues in that information is processed sequentially by different populations of neurons, allowing increasingly complex features and other salient components of information to build-up and aggregate into comprehensive representations (Felleman and Van Essen, 1991). This structure also potentially allows for different combinations of information to be pooled and thus new representations of data can be constructed and anticipated. The previously discussed Neocognitron represents an algorithm that leverages hierarchy to pool low-level features of visual objects from separate fields of view into fully-assembled representations of objects that can then be classified. The Hierarchical Temporal Memory (HTM) algorithm was a learning model developed by Jeff Hawkins at Numenta Inc. which was intended to model the physical functionality of the neocortex using a uniform neural structure composed in layers (Hawkins, et al. 2010). HTM is at the core of Numenta' s Grok cyber analytics tool, and the algorithm is typically used for unsupervised learning with sparse cell activation and inhibitory connections to efficiently learn correlations and make temporal predictions based on incoming data. A major challenge to developing layered hierarchical algorithmic approaches is the difficulty in training such algorithms within a reasonable length of time relevant to the problem of interest. Deep Learning (DL) is a modern approach towards neural networks that enables the unsupervised learning of hierarchical representations of data using multi-layered architectures in contrast to shallow networks (Hinton and Salakhutdinov, 2006). When combined with the increased speed of modern computers, DL has achieved considerable success in addressing pattern recognition problems and has attracted wide-spread attention by outperforming alternative machine learning methods. Algorithms theory has been developed around deep neural networks (DNNs), including training optimization techniques for RBMs (Hinton 2012) and methods for displaying data representations throughout networks (Bengio, 2007; 2009). Supervised DNNs have won numerous recent international pattern recognition competitions, achieving the first visual pattern recognition results that surpass human performance in limited domains such as traffic sign recognition (Schmidhuber, 2015). In 2012, a deep CNN won the ImageNet competition (Krizhevsky et al., 2012) and since then, every entry now leverages CNNs to some degree. DL has been applied to a host of problems including object recognition in images and video, speech recognition, particle searches in collider data, and predictive analytics of protein-nucleic acid interactions (Jones, 2014; Baldi et al., 2014; Alipanahi et al., 2015). Recently, companies such as Samsung and Panasonic have sought to leverage DL for smartphone applications such as facial expression recognition (Song et al., 2014) and for classification of data in noisy environments (Gu and Rigazio, 2014).

As mentioned previously, the training of DNNs presents a significant hindrance for the use of such networks, especially for problem spaces that require large amounts of unlabeled data. Training of deep architectures is also difficult due to the increasingly small adjustments made to weights when applying the chain rule during backpropagation calculations (vanishing gradient problem) (Hochreiter et al., 2001). Faster computers and improvements in algorithm techniques have helped with these training challenges (Schmidhuber, 2015), and numerous efforts to assemble specialized hardware for training deep networks have been initiated, including a 16,000 CPU core system developed by Google, Inc. for use with video data (Le, 2013). In this work, the individual frames of the data were unlabeled and after 3 days of training on randomly-sampled frames from 10 million YouTube videos, the algorithm learned to recognize human faces and bodies in addition to cat faces. The Google system outperformed competitor systems that relied on manually-crafted features to process images from the standardized database ImageNet, achieving a 15.8% classification accuracy. Schroff et al. (2015) recently demonstrated a 30% reduction in facial recognition error rates using the Labeled Faces in the Wild and Youtube Faces datasets. The FaceNet system used a deep CNN trained using gradient descent with backpropagation to achieve high accuracy in facial recognition under the additional challenge of having images with changes in pose and illumination. Google DeepMind has focused on leveraging reinforcement learning and deep CNNs for complex tasks such as video game play (Mnih et al., 2015). Recently, this team used CNNs to generate feature representations of player positions in the board game Go, and relied on a traditional Monte Carlo tree search algorithm to select appropriate moves (Silver et al., 2016). DeepMind's AlphaGo program eventually defeated several champion human players at the game of Go in 2016, marking a significant achievement for data-driven computing algorithms.

Project Adam was a DL effort from Microsoft Research Corporation that used a cluster of 120 server machines to train and operate a 2×10^9 connection DNN for image classification (Chilimbi et al., 2014). The system was demonstrated on MNIST digit data (99.63% accuracy) and ImageNet picture data, the latter of which displayed an accuracy of 29.8%, an improvement of ~2x over the previous best from Google, Inc.'s multicore CPU-based deep learning system. The performance improvement is largely attributed to running the system with asynchronous batch processing of the weights, a process that injects noise into the training and assists the system in escaping local minima. Other laboratories have focused on incorporating GPUs into specialized hardware for DL applications. Coates et al. (2013) assembled a system with GPU servers and Infiniband interconnects to rapidly communicate gradient calculations for large network training. This system was capable of training a network with $\sim10^{10}$ connections in 3 days of processing time. Dean et al. (2012) showed that with a "distributed optimization" approach wherein the DNN training is performed in parallel across several model replicas, the combination of model parallelism and data parallelism in a CPU cluster can produce a significant performance advantage in classification accuracy (object and speech recognition) over GPU-based deep learning systems.

Another DL hardware effort was the Deep Speech system from Baidu Inc. (Hannun et al., 2014). This speech recognition system implemented a RNN on a multi-GPU hardware platform and displayed a record low word error rate on a standardized telephone speech dataset compared to other DNN/hidden Markov model-based methods. Branching off from the speech recognition work, Baidu Inc. recently described an image recognition system named Deep Image (Wu et al., 2015). The Minwa hybrid supercomputer developed for this effort was a combination of CPU and GPU cores with high-speed Infiniband connections for processing the ImageNet Large-Scale Visual Recognitions Challenge dataset. Crucial to improving the classification performance was a series of data pre-processing steps such as vignetting that were used to increase the amount of training data available for the algorithm.

Statistical and dynamical machine learning algorithms and hardware

In addition to algorithms such as the Perceptron that directly emerged from biophysical concepts in neuroscience, other techniques with less of a connection to neuroscience and more directly tied to psychology also developed (Figure 1). One example is statistical learning theory, an approach originating from the psychology field that used statistics to map behaviors onto complex stimuli (Estes and Suppes, 1959). Although the neural-inspired work by Hebb, Rosenthal, and others provided some degree of mathematical formalism, the use of statistical analyses in neuromorphic and neural-inspired algorithms was mostly lacking. Statistical learning theory was a sharp departure from convention given its reliance on statistics, and this formalism was eventually incorporated into concepts of learning network theory (Barron and Barron, 1988; Vapnik, 2000; Bousquet et al., 2004). Later, support vector machines (SVMs) were developed to use statistics to maximize the separation between data classes while minimizing classification error (Cortes and Vapnik, 1995). The strength of SVMs is the use of kernels to map data that in its raw form is not linearly separable into higher dimensions where the data is linearly separable. Once mapped, the margin between the classification decision boundaries and the training data is maximized in this feature-based solution space. As a result, a single unique solution is provided, and thus SVM algorithms are not susceptible to becoming trapped in local minima or producing different solutions based on initial conditions. Drawbacks to the use of SVMs include the training cost scalability (in general, a problem with n data points would require n^2 optimization steps) and the difficulty in parallelizing the algorithm for implementation onto hardware accelerators. SVMs have been used in many applications such as chemistry, bioinformatics, face detection, and character recognition (Bennett and Campbell, 2000; Ivanciuc, 2007). Hardware implementations of SVMs such as the Kerneltron have been developed for applications in object recognition in video data (Genov and Cauwenberghs, 2003). The Kerneltron was a VLSI chip capable of high-throughput parallel matrix-vector multiplication with a 100-10,000x improvement in performance-power efficiency as compared to a 32bit floating point digital signal processor. In this system, wavelet decomposition was used to extract feature vectors from training data and then a SVM was trained on these vectors to generate accurate classifications. The classification procedure relied on computing inner-products with matrix-vector multiplication, followed by a thresholding procedure to make final object classifications. Proposed applications for the 9 mm^2 Kerneltron chip included use in applications where power and weight are major concerns such as navigational systems. Other laboratories have demonstrated the capabilities of VLSI-based SVM systems for real-time simultaneous tracking of multiple objects within high-definition video data (Takagi et al., 2014). In this work, a modified histogram of oriented gradients algorithm was implemented in VLSI (65 nm CMOS), including an SVM module with dedicated SRAM for storing classification coefficients of detected objects.

Another algorithm in the statistical machine learning lineage is the decision tree. Decision trees largely emerged from concept learning theory, a psychology framework that relied on the use of induction and assignment of attributes to separate data into distinct classes (Bruner et al., 1956; Hunt et al., 1966). In Hunt et al.'s original formulation, the set of attributes needed to classify a set of data was assembled into a decision tree, and the cost of classification was assessed in regard to the cost of assigning value to attributes as well as the cost of misclassifying data. Later developments in induction-based decision trees include the ID3 algorithm, a method that focused on minimizing entropy (and thus maximizing information) during classification procedures (Quinlan, 1986). Decision trees have been used for data mining applications where a large number of related variables are used to classify data based on examples (Quinlan, 1990). The random forest implementation of decision trees incorporated the use of ensemble learning by randomly generating multiple decision trees in order to optimize data classification and reduce the likelihood of overfitting (Ho, 1998; Breiman, 2001; Banfield et al., 2007). Recently, several labs have focused on hardware acceleration of random forest algorithms using graphical processing units (GPUs) and CPUs (Osman, 2009; Van Essen et al., 2012; Liao et al., 2013), with Sharp et al. (2008) demonstrating a 100x speed-up (GPU compared to a CPU) of the evaluation of a decision tree forest designed to recognize objects.

While statistical machine learning approaches brought a degree of mathematical rigor to data-driven computing, these methods struggle to handle dynamical problems where the data and conditions are changing over the course of time. Recent work combined SVMs with game theory in order to accommodate dynamical distributions of data (Vineyard et al., 2015; 2015). However, another branch of algorithms referred to here as dynamical machine learning were developed specifically to handle these types of problems. The previously discussed SNARC system was influenced by the work of early psychologists and physiologists in the area of reinforcement as a method of learning, a temporal process in which an agent is rewarded (or not rewarded) for particular behaviors through a "cost" function that has to be optimized over the course of time (Pavlov and Gantt, 1928; Skinner, 1933). A differentiating aspect of reinforcement learning is the need to balance exploration (examining new solutions with potential for greater reward) with exploitation (using already known solutions with known rewards) to minimize the overall system cost function. Forms of reward-based learning in neurobiological systems have been modeled to examine the role of dopamine as a short-term (milliseconds to seconds) modulator of plasticity (Izhikevich, 2007) and experimentally measured to determine the impact of dopamine on longer-term (minutes to hours) memory encoding in the hippocampus (Du et al., 2016). In this sense, dopamine-reinforced learning can serve as a mechanism by which neurobiological networks can be trained to minimize "error" in network activity at a wide dynamic range of time-scales. Reinforcement learning as an algorithm has been used in numerous applications including pattern recognition, robotics control, and game theory (Minsky, 1961; Kaelbling et al., 1996; Kober and Peters, 2012). Another example of a dynamic algorithm is the Markov Decision Process (Bellman, 1957; Szepesvari, 2010). In this algorithm, sequential decision-making operates in a loop with an agent observing and planning actions to drive the system to the next "state" under the influence of a quantifiable reward (Sutton and Barto, 1998; Faust, 2014). A similar state-transition algorithm is a Bayesian network. Originally designed as a "model for humans' inferential reasoning" and used for static problems with conditional probabilistic state transitions (Pearl, 1986), the subsequent development of Hidden Markov Models (Baum and Petrie 1966, Rabiner 1989) and Dynamic Bayesian Networks (Murphy, 2002) brought these techniques into the time domain and enabled new applications in speech recognition and navigation. Hardware implementations of state-transition-based algorithms have been developed, including the automata processor from Micron Technology (Dlugosch et al., 2014). This work demonstrated a hardware system configured to process Perl Compatible Regular Expression (PCRE) syntax as well as XML-based language for network data applications. The design was implemented in DRAM process technology and consisted of several elements for symbol processing, a parallelized routing matrix for distributing signals, and components for counters and Boolean logic functions. The Micron Automata design compared favorably to nondeterministic finite automata implemented in field programmable gate array (FPGA) technology (Kaneta et al., 2011; Yang and Prasanna, 2012). Recently, the simulator for Micron's Automata Processor chip was used to demonstrate its potential use in part-of-speech tagging (Zhou et al., 2015).

Device technologies for neural-inspired and neuromorphic computing

The neuromorphic and neural-inspired hardware systems discussed thus far have relied on existing microelectronic device technology and have developed new designs to combine those devices into different architectures. Conventional devices can also be operated in different modes in order to achieve better neuromorphic and neural-inspired characteristics, e.g. CMOS devices operated in subthreshold mode. New designs for conventional CMOS hardware such as switched capacitor circuits have also been developed to avoid the use of electrical currents for computation, thus reducing the negative impact of leakage currents (Mayr et al., 2015). And to improve the ability to model synaptic learning rules, CMOS transistors have been modified with a floating gate design (Ramakrishnan et al., 2011).

Researchers have also investigated the design of fundamentally novel microsystem device technologies to achieve neuromorphic and neural-inspired computation with improved performance characteristics such as lower energy consumption, reduced areal footprint, and wider dynamic range (Kuzum et al., 2013). For example, the size of a static random access memory (SRAM) cell limits the amount of SRAM that can be placed on chip; thus, conventional microelectronic systems rely on energy-intensive off-chip memory storage which is a severe limitation for data-driven computing approaches that require significant training. In addition, an SRAM cell can only hold one bit of information. These limitations have led to the development of dense, non-volatile alternative memory technologies to serve as biologically-inspired microelectronic hardware synapses for low-power mobile computing applications (Wong and Salahuddin, 2015). Candidate technologies typically store device state with a property other than charge given the difficulty in maintaining charge absent a continuous supply voltage. Technologies capable of back-end processing for high-density 3D layering are also viewed as advantageous. Panasonic Inc. has undertaken investments in three-terminal lead-zirconium-titanate ferroelectric devices to construct electronic synapses (Kaneko et al., 2014). However, like SRAM and dynamic random access memory (DRAM), ferroelectric RAM is also a front-end device technology incompatible with 3D layering. Other technologies currently being investigated include resistance-based memory which relies on controlled switching between low and high conductance states. Different resistive switching materials technologies include metallic oxides (Strukov et al., 2008; Wei et al., 2008; Lee et al., 2011; Mickel et al., 2014; Prezioso et al., 2015), oxides with metallic carriers (Kozicki et al., 2004; Mai et al., 2015), and non-oxide semiconductors with metallic carriers (Jo et al., 2010). Advantages to using these resistive and memristive (when the resistance is a function of the historical current) technologies include that the conductance state of the device is retained without any sustaining current and the inherent noise in these devices can be leveraged for probabilistic computing (Al-Shedivat et al., 2015). Potential advantages to using resistive memory devices are the low write energy, high scalability with potential for 3D layering, and the analog-like state-transition behavior (Indiveri et al., 2013; Mandal et al., 2014; Saighi et al., 2015; Agarwal et al., 2016a; Agarwal et al., 2016b). Phase change memory (PCM) is a similar technology wherein the conductance of a semiconductor layer is reversibly switched with Joule heating between a low conductivity amorphous phase to a high conductivity crystalline phase (Raoux et al., 2008; Wong et al., 2010). Points of interest for PCM devices are the relatively high level of development of this technology by industry and the high retention times (Jackson et al., 2013; Shelby et al., 2015). Spin transfer torque magnetic random access memory (STT-RAM) devices rely on the use of an electrical current to change the polarization direction of a ferromagnet and the corresponding change in conductivity between parallel and anti-parallel spins in thin films (Kishi et al., 2008; Kent and Worledge, 2015). Information is stored magnetically, which provides superior long-term retention, and state changes are written and read electrically in these devices. Challenges with this technology include difficulty in scaling due to the use of nanoscale magnetic structures and the limited dynamic range between the on and off states.

Conclusions

Over the last century, researchers have recognized the distinct advantages that neuromorphic and neural-inspired algorithms and hardware can provide to address challenging, data-intensive classes of problems. The first wave of neural-inspired computing research sought to develop phenomenological model systems of how organisms perform certain complex tasks such as maze-navigation. Additional efforts that were more closely coupled to mathematical formulations of algorithms theory helped move the field past trial-and-error niche demonstrations and into more generalizable applications such as object and speech recognition. The theoretical limitations and practicality of neural-inspired approaches have always been a source of concern within the research community, and new developments in algorithms theory and improvements in hardware have provided new opportunities for addressing some of those concerns. The most recent wave of neural-inspired computing has produced a significant amount of math theory around algorithm development, addressing important practical issues such as training techniques, visualization of data representations, and learning strategies. In addition, hardware has been fabricated to instantiate algorithms with improved computational efficiency in speed and/or power consumption. Much of this work has been supported by the steady advances made by the microelectronics industry via Moore's law. Smaller and faster microprocessors and advanced architectures such as GPUs have driven the neuromorphic and neural-inspired computing field through previous computational hurdles and have also led to a proliferation of data at unmanageable volumes. Still, neuromorphic systems face challenges in regard to incorporating learning circuitry with adaptable timescales capable of rapid low-power updating of synaptic weights (Hasler and Marr, 2013). The reputed end of Moore's law presents an opportunity for researchers to leverage modern advances in neuroscience to spur the next wave of algorithm and hardware advancements. For instance, modern neuroscience research is using new technologies such as optogenetics to improve our understanding of how the brain processes, transforms, and calculates information (Boyden et al., 2005; Deisseroth, 2015). Developments in this technology have enabled closed-loop experiments where an initial probing of a set of neurons can then be modified based on recorded responses (Sohal et al., 2009). This is a crucial advance necessary to improve the specificity of connections between neurons and to improve our understanding of the signaling dynamics within networks of neurons. Another issue that needs to be resolved includes identifying the time-evolving neural circuits ("chronnectome") involved in complex sensory, motor, and cognitive activities (Churchland et al. 2012; Calhoun et al., 2014), and then performing such population-level measurements with single cell resolution (Packer et al., 2015).

In order to maintain progress in this field, the research community must navigate several difficult questions in regard to the next generation of neuromorphic and neural-inspired algorithms and hardware systems:

1. How connected should the development of neuromorphic hardware be to the neuroscience field? This question was raised earlier in regard to the level of mimicry of neural tissue that should be pursued. To highlight the biological complexity of neural tissue, Figure 2 describes

a variety of plasticity mechanisms that impact learning, memory, and other forms of computation in neurobiological systems (INSERT FIGURE 2 HERE). The range over which these phenomena operate in time and throughout neural tissue is large, from Spike-Timing-Dependent-Plasticity (STDP) which occurs rapidly at individual sub-micron synapses (Feldman 2009) to slower processes such as the regional restructuring of neural tissue at the scale of millions of cells that take place on the time-scale of months (Zatorre et al., 2012). In addition, we previously discussed the role of chemical neuromodulators such as dopamine on reward-based learning. Clearly, neurobiological systems have an array of tools by which complex computational activities can be performed. One implication of this considerable diversity in plasticity mechanisms is that it suggests neuromorphic hardware designers should be deliberate in how neural plasticity is abstracted into hardware systems. The combination of broad and narrow spatio-temporal scales used by brain to process information is more powerful than any one mechanism in isolation, and this partly explains the performance challenges observed when neural-inspired systems focus on only a single unsupervised learning process such as STDP. Another issue raised by Figure 2 is the significant difference between learning in biological systems and neural-inspired algorithms. The various forms of plasticity in biological systems are demonstrably robust and better capable of handling unstructured and noisy data compared to relatively fragile artificial neural network algorithms. It could be argued that this robustness means that strong statistical assumptions such as independent and identically-distributed (iid) requirements (Achler, 2014) are not as necessary for biological systems. Thus, to meet the challenge of dynamic and noisy real-world problems, neural-inspired algorithms and hardware need to develop this level of flexibility. The specifics of the problem at hand are obviously influential in that neuromorphic algorithms and hardware designed for applications should be driven to the optimum point where functionality is achieved while minimizing size, weight, power, etc. Application-focused neuromorphic hardware should focus on replicating function (e.g. coincidence detection) instead of replicating biology (e.g. the binding kinetics of molecules involved in biological coincidence detection). The more difficult challenge is to determine the degree to which hardware used to model and simulate neural systems as a research tool be driven to biological fidelity. Traditional high-performance computing (HPC) resources have been used for large-scale computational models (e.g. the neurogenesis model in Aimone et al. 2009) that have then inspired *in vivo* neuroscience experiments (multi-electrode field recordings described in Rangel et al., 2014). The neuromorphic hardware described earlier in this manuscript for use in neural system modeling have been useful tools, yet we are unaware of any cases where these systems have performed simulations not capable of being performed on traditional HPC hardware and subsequently being used to guide novel *in vivo* or *in vitro* neuroscience research. We expect more differentiating neural simulations to be performed on neuromorphic hardware as the systems become more widely distributed.

2. What level of neural-inspiration should be pursued for algorithms? Neural inspiration can range from very abstract concepts to highly specific mechanisms. Cognitive architectures are abstract approaches that have been used to develop models for high-level phenomena such as episodic memory (Nuxoll and Laird, 2007) and cognitive self-knowledge (Sun et al., 2006). But because

these models are abstracted from experimental neuroscience observations, it is unclear how they should be altered or improved in situations where their function differs from the biological system. On the other hand, experimental neuroscience can be used to measure neural phenomena at the molecular, cellular, and network level, but such data is difficult to translate to higher-level cognitive activities and to incorporate within algorithms. For example, traditional machine learning methods such as Markov models and neural-inspired methods such as DL and CNNs have been successful in speech recognition and image recognition applications. But besides the hierarchical structure and the input integration and thresholding functionality, there are few neuroscience principles embedded within ANN-based algorithms. For instance, DL algorithms require extensive training with large volumes of data whereas biological neural systems don't have such stringent requirements for complex representations to be learned. Lake et al. (2015) recently demonstrated Bayesian Program Learning (BPL) wherein data is represented with probabilistic generative models. With this framework, complex concepts are partitioned into subpart "primitives" that can be sampled and recombined in different ways to create highly complex representations. On a one-shot classification task (learning from only one example data-point), BPL showed a superior error rate (3.3%) compared to humans (4.5%) and deep convolutional nets (13.5%). Approaches such as these which seek to replicate biological network functionality such as one-shot learning hold great promise for the future of neural-inspired algorithms. To realize this potential, formal mathematical theories by which to translate such functionality into new algorithms are needed. The progression of retina-inspired neuromorphic hardware from the phenomenological and generalized concepts of the Neocognitron (e.g. "S" and "C" cells) to the biologically-accurate concepts of Okuno et al.'s (2015) VLSI retina-based emulator (e.g. photoreceptors and ganglion cells) shows how new scientific developments should encourage technology to not only mature in complexity but to also improve application-driven functionality. Finally, as previously discussed in regard to Figure 2, neurobiological systems rely on a diverse suite of mechanisms to process information. Algorithms have historically been applied in isolation with the selection of algorithms being based upon the nature and complexity of the problem. Thus, Perceptrons have been used for problems with limited spatial and temporal complexity, while DL has seen prevalent use for problems with significant spatial complexity such as image recognition. In the time-domain, neither of these techniques can be used in isolation, and thus algorithms such as RNNs and Reservoir Computing have been used for complex time-domain problems such as speech recognition. Only recently have multiple algorithms been combined to address the spatio-temporal complexity of challenging problems such as the game of Go (Silver et al., 2016) and image captioning (Karpathy and Fei-Fei, 2014). Future algorithmic development should continue along this path of integrated solutions that are capable of handling a wide variety of datasets.

3. Should the community focus on developing specialized hardware or adapting commercial-off-the-shelf (COTS) hardware? The community is split between these two options and impressive systems in both realms have been demonstrated. While specialized hardware typically requires higher cost and results in less generalizability, we believe this approach presents the most promising path forward given the improved ability to tailor such systems for specific

application needs. This will also require the incorporation of standardized interfaces to improve ease-of-use and the technical maturation of such technologies to eliminate performance problems. Specialized hardware such as the Neurogrid system, SpiNNaker, and TrueNorth hold promise not only as research tools, but as solutions for commercial applications. As the connections between such hardware platforms and algorithms strengthen (e.g. convolutional neural networks on SpiNNaker in Serrano-Gotarredona et al., 2015), the positive impact of specialized hardware on the research community will increase.

4. How will the practical limitations of existing microelectronics technologies be handled in order to build next generation neuromorphic and neural-inspired hardware? A major challenge for neuromorphic and neural-inspired hardware is the limited fan-in/fan-out connectivity and its negative impact on system performance. Biological neural systems have massive parallelism (upwards of 10,000 connections on individual neurons), thus new architectures and microelectronic devices capable of such connectivity may or may not need to be developed (see question #1 above). If this level of parallelism is to be pursued, then in addition to improving connectivity technologies in hardware, this issue can also be address algorithmically. For instance, an algorithm that requires thousands of interconnects may possibly be transformed into a lower connectivity version for hardware implementation, with perhaps a trade-off in sparsity or network size. This would require a more thorough understanding of biological neural circuit behavior, however, such hardware-guided algorithm development may be essential for implementing algorithms extracted from three-dimensional biological neural systems and projected onto two-dimensional semiconductor platforms.
5. Will conventional CMOS microelectronics be supplanted by novel devices for use in neuromorphic systems? The operating principles of conventional CMOS devices are well understood and strategies have been implemented to adapt these devices for neuromorphic applications. However, translating biological systems consisting of ion channels and membrane receptors into transistors and other microelectronic components is difficult and at times can be forced. Novel devices with properties that more readily comport to neurobiological functions should continue to be pursued in order to improve the functionality of hardware implementations. As an example, resistive memory devices are more similar to biological synapses than other microelectronic devices given their operational reliance on changes in conductance. The two-terminal architecture of resistive memory devices also lends itself to the high density networks necessary for difficult pattern recognition applications such as object classification in video feeds. However, these devices obviously lack some of the characteristics of biological synapses such as gain and other modulatory features that make biological systems computationally powerful. Future work in novel devices needs to balance the pursuit of biological computation features with the biological fidelity concern discussed previously in question #1 above. Finally, new devices should also be developed in regard to their ability to perform particular mathematical functions more rapidly and/or more efficiently. A considerable amount of neural network hardware is focused on the multiply-and-accumulate calculations needed for matrix operations. Hardware researchers need to continue to collaborate with math theory and algorithm researchers to identify additional mathematical functions that may be

useful for neural network-based hardware systems, and then develop new microsystem devices capable of those calculations with fewer or less energy-intensive steps.

Several of the challenges enumerated here involve the use of neuroscience research, thus strong collaborations between neuroscientists, hardware designers, and math theoreticians will help to facilitate the cross-disciplinary dialogue to identify and decipher important computational functionality in biological systems. The challenge will be to leverage such advances into the development of new algorithms and to implement hardware-based solutions where necessary and practical.

Acknowledgements

The authors gratefully acknowledge financial support from Sandia National Laboratories' Laboratory Directed Research and Development Program, and specifically the Hardware Acceleration of Adaptive Neural Algorithms (HAANA) Grand Challenge Project. Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under Contract No. DE-AC04-94AL85000.

References

1. Agarwal, S., Quach, T., Parekh, O., Hsia, A.H., DeBenedictis E.P., James, C.D., Marinella, M.J., Aimone, J.B. (2016a). Energy scaling advantages of memristor crossbar based computation and its application to sparse coding. Frontiers in Neuroscience, 9, 484-. doi: 10.3389/fnins.2015.00484.
2. Agarwal, S., Plimpton, S. J., Hughart, D. R., Hsia, A.H., Richter, I., Cox, J. A., James, C.D., Marinella, M.J. (2016b). Resistive memory device requirements for a neural algorithm accelerator. International Joint Conference on Neural Networks (IJCNN), 929-938. doi: 10.1109/IJCNN.2016.7727298.
3. Aimone, J. B., Wiles, J. and Gage, F. H. (2009). Computational influence of adult neurogenesis on memory encoding. *Neuron* 61, 187–202. doi:10.1016/j.neuron.2008.11.026
4. Aimone, J. B., Deng, W. and Gage, F. H. (2010) Adult neurogenesis: integrating theories and separating function. Trends in Cognitive Neuroscience 14, 325-337. doi: 10.1016/j.tics.2010.04.003.
5. Alipanahi, B., Delong, A., Weirauch, M.T., Frey, B.J. (2015) Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology* 33, 831-838. doi:10.1038/nbt.3300
6. Al-Shedivat, M., Naous, R., Cauwenberghs, G., and Salama, K.N. (2015). Memristors Empower Spiking Neurons With Stochasticity. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 5**,** 242-253. doi: 10.1109/Jetcas.2015.2435512
7. Amit, D.J., Gutfreund, H., and Sompolinsky, H. (1987). Statistical mechanics of neural networks near saturation. *Annals of Physics* 173**,** 30-67. doi: 10.1016/0003-4916(87)90092-3
8. Ananthanarayanan, R., Esser, S.K., Simon, H.D., and Modha, D.S. (2009). The cat is out of the bag: cortical simulations with 109 neurons, 1013 synapses. In *IEEE Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, 1-12. doi: 10.1145/1654059.1654124
9. Ashby, W.R. (1960). *Design for a Brain.* Springer Science & Business Media
10. Atencia, M., Boumeridja, H., Joya, G., Garcia-Lagos, F., and Sandoval, F. (2007). FPGA implementation of a systems identification module based upon Hopfield networks. *Neurocomputing* 70**,** 2828-2835. doi: 10.1016/j.neucom.2006.06.012

11. Atlas, L., Homma, T., and Marks, R. (1988). An artificial neural network for spatio-temporal bipolar patterns: Application to phoneme classification. *Proceedings Neural Information Processing Systems (NIPS)*, 31-40.
12. Baldi, P., Sadowski, P., and Whiteson, D. (2014) Searching for exotic particles in high-energy physics with deep learning. *Nature Communications* 5, 4308. doi:10.1038/ncomms5308
13. Baernstein, H., and Hull, C.L. (1931). A mechanical model of the conditioned reflex. *The Journal of General Psychology* 5**,** 99-106. doi: 10.1080/00221309.1931.9918381
14. Banfield, R.E., Hall, L.O., Bowyer, K.W., and Kegelmeyer, W.P. (2007). A comparison of decision tree ensemble creation techniques. *IEEE Trans Pattern Anal Mach Intell* 29**,** 173-180. doi: 10.1109/TPAMI.2007.2
15. Barlow, H.B. (1953). Summation and inhibition in the frog's retina. *J Physiol* 119**,** 69-88. doi: 10.1113/jphysiol.1953.sp004829
16. Barron, A.R., and Barron, R.L. (1988). Statistical learning networks: a unifying view. In *Symposium on the Interface: Statistics and Computing Science,* Reston, Virginia.
17. Baum, L. E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics* 37: 1554–1563. doi:10.1214/aoms/1177699147
18. Bekolay, T., Bergstra, J., Hunsberger, E., DeWolf, T., Stewart, T.C., Rasmussen, D., Choo, X., Voelker, A.R., and Eliasmith, C. (2014). Nengo: a Python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics* 7, 48-61. doi: 10.3389/fninf.2013.00048
19. Bellman, R. (1957). A Markovian decision process. DTIC Document No. P-1066. Rand Corporation, Sant Monica, CA.
20. Bengio, Y., and LeCun, Y. (2007). Scaling learning algorithms towards AI. In *Large-scale kernel machines*. Bottou, L., Chapelle, O., DeCoste, D., and Weston, J., Eds., MIT Press.
21. Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends in Machine Learning* 2**,** 1-127. doi: 10.1561/2200000006
22. Benjamin, B., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran, A.R., Bussat, J.M., et al. (2014). Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations. *Proceedings of the IEEE* 102**,** 699-716. doi: 10.1109/Jproc.2014.2313565
23. Bennett, K.P., and Campbell, C. (2000). Support vector machines: hype or hallelujah? *ACM SIGKDD Explorations Newsletter* 2**,** 1-13. doi: 10.1145/380995.380999
24. Beyeler, M., Carlson, K. D., Chou, T. S., Dutt, N., & Krichmar, J. L. (2015). CARLsim 3: A user-friendly and highly optimized library for the creation of neurobiologically detailed spiking neural networks. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN'15) (Killarney), 1–8.
25. Boahen, K. (2005). Neuromorphic Microchips. *Sci Am* 292**,** 56-63. doi: 10.1038/scientificamerican0505-56
26. Boahen, K. (2006). Neurogrid: emulating a million neurons in the cortex. In *Conf. Proc. IEEE Eng. Med. Biol. Soc*, 6702. doi: 10.1109/IEMBS.2006.260925
27. Borji, A., and Itti, L. (2014). Human vs. computer in scene and object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 113-120. doi: 10.1109/CVPR.2014.22
28. Bousquet, O., Boucheron, S., and Lugosi, G. (2004). Introduction to statistical learning theory. In *Advanced Lectures on Machine Learning*. Berlin, Hidelberg, Springer, 169-207.
29. Boyden, E.S., Zhang, F., Bamberg, E, Nagel, G., Deisseroth, K. (2005). Millisecond-timescale, genetically targeted optical control of neural activity. *Nature Neuroscience* 8, 1263–1268. doi:10.1038/nn1525
30. Bradner Jr, H. (1937). A New Mechanical "Learner". *The Journal of General Psychology* 17**,** 414-419. doi: 10.1080/00221309.1937.9918012
31. Breiman, L. (2001). Random forests. *Machine Learning* 45**,** 5-32. doi: 10.1023/A:1010933404324

32. Bruner, J.S., Goodnow, J.J., and George, A. (1956). Austin. 1956. A study of thinking. *New York, John Wiley & Sons. Bruner.*
33. Bryson, A.E., and Denham, W.F. (1962). A steepest-ascent method for solving optimum programming problems. *Journal of Applied Mechanics* 29**,** 247-257. doi: 10.1115/1.3640537
34. Calhoun, V.D., Miller, R., Pearlson, G., and Adal. (2014). Connectivity networks as the next frontier in fMRI data discovery. *Neuron* 84, 262-274. doi:10.1016/j.neuron.2014.10.015
35. Calimera, A., Macil, E., and Poncino, M. (2013). The Human Brain Project and neuromorphic computing. *Functional Neurology* 28, 191-196. doi: 10.11138/FNeur/2013.28.3.191
36. Carlson, K.D., Nageswaran, J.M., Dutt, N., and Krichmar, J.L. (2014). An efficient automated parameter tuning framework for spiking neural networks. Frontiers in Neuroscience 8, 10. doi: 10.3389/fnins.2014.00010
37. Cepelewicz, J. (2016). The U.S. government launches a $100-million “Apollo Project of the Brain.” Scientific American. http://www.scientificamerican.com/article/the-u-s-government-launches-a-100-million-apollo-project-of-the-brain/
38. Chicca, E., Badoni, D., Dante, V., D’Andreagiovanni, M., Salina, G., Carota, L., Fusi, S., and Del Guidice, P. (2003) A VLSI recurrent network of integrate-and-fire neurons connected by plastic synapses with long-term memory. *IEEE Transactions on Neural Networks* 14, 1297-1307. doi: 10.1109/TNN.2003.816367
39. Chicca, E., Stefanini, F., Cartolozzi, C., and Indiveri, G. (2014). Neuromorphic electronic circuits for building autonomous cognitive systems. *Proceedings of the IEEE* 102, 1367-1388. doi: 10.1109/JPROC.2014.2313954
40. Chilimbi, T., Suzue, Y., Apacible, J., and Kalyanaraman, K. (2014). Project adam: Building an efficient and scalable deep learning training system. In *11th USENIX Symposium on Operating Systems Design and Implementation*, 571-582.
41. Churchland, M.M., Cunningham, J.P., Kaufman, M.T., Foster, J.D., Nuyujukian, P., Ryu, S.I., Shenoy, K.V. (2012). Neuronal population dynamics during reaching. *Nature* 487, 51-56. doi: 10.1038/nature11129
42. Clark, W., and Farley, B. (1955). Generalization of pattern recognition in a self-organizing system. In *Proceedings of the March 1-3, 1955, Western Joint Computer Conference*: *ACM*, 86-91. doi: 10.1145/1455292.1455309
43. Coates, A., Huval, B., Wang, T., Wu, D., Catanzaro, B., and Andrew, N. (2013). Deep learning with COTS HPC systems. In *Proceedings of the 30th International Conference on Machine Learning*, 1337-1345. doi: 10.1.1.308.9984
44. Cognimem Technologies, Inc. CM1K hardware User’s Manual. (2013). http://www.cognimem.com/_docs/Technical-Manuals/TM_CM1K_Hardware_Manual.pdf
45. Cortes, C., and Vapnik, V. (1995). Support-Vector Networks. *Machine Learning* 20**,** 273-297. doi: 10.1023/A:1022627411411
46. Cruz-Albrecht, J.M., Yung, M.W., and Srinivasa, N. (2012). Energy-efficient neuron, synapse and STDP integrated circuits. *IEEE Transactions on Biomedical Circuits and Systems* 6**,** 246-256. doi: 10.1109/TBCAS.2011.2174152
47. Dalakov, G. (2016). The Robot Rat of Thomas Ross. http://history-computer.com/Dreamers/Ross.html
48. Dan, Y., and Poo, M.-m. (2004). Spike timing-dependent plasticity of neural circuits. *Neuron* 44**,** 23-30. doi: 10.1016/j.neuron.2004.09.007
49. Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Senior, A., Tucker, P., Yang, K., and Le, Q.V. (2012). Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, 1223-1231. doi: 10.1.1.258.5430
50. Deisseroth, K. (2015). Optogenetics: 10 years of microbial opsins in neuroscience. *Nature Neuroscience* 18, 1213-1225. doi: 10.1038/nn.4091

51. Delbruck, T. (1993). Silicon retina with correlation-based velocity-tuned pixels. *IEEE Transactions on Neural Networks* 4, 529-541. doi: 10.1109/72.217194
52. Dinneen, G. (1955). Programming pattern recognition. In *Proceedings of the Western Joint Computer Conference*: ACM, 94-100. doi: 10.1145/1455292.1455311
53. Dlugosch, P., Brown, D., Glendenning, P., Leventhal, M., and Noyes, H. (2014). An efficient and scalable semiconductor architecture for parallel automata processing. *IEEE Transactions on Parallel and Distributed Systems* 25**,** 3088-3098. doi: 10.1109/Tpds.2014.8
54. Douglas, R., M. Mahowald, and C. Mead, Neuromorphic analogue VLSI. *Annual Review of Neuroscience*, 1995. 18: p. 255-281. doi: 10.1146/annurev.ne.18.030195.001351
55. Du, H., Deng, W., Aimone, J.B., Ge, M., Parylak, S., Walcvh, K., et al. (2016). Dopaminergic inputs in the dentate gyrus direct the choice of memory encoding. Proceedings of the National Academy of Sciences 113, E5501-E5510, doi: 10.1073/pnas.1606951113
56. Eide, A., Lindblad, T., Lindsey, C., Minerskjold, M., Sekhniaidze, G., and Szkely, G. (1994). An implementation of the zero instruction set computer (ZISC036) on a PC/ISA-bus card. Presented at *WNN/FNN Washington DC.* doi: 10.1016/0168-9002(95)00074-7
57. Eliasmith, C. and Anderson, C.H. (2003). Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems. Cambridge, MIT Press. ISBN: 978-0262550604
58. Eliasmith, C., Stewart, T.C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., and Rasmussen, D. (2012). A large-scale model of the functioning brain. *Science* 338**,** 1202-1205. doi: 10.1126/science.1225266
59. Esser, S. K., Andreopoulus, A., Appuswamy, R., Datta, P., Barch, D., Amir, A., et al. (2013). Cognitive computing systems: algorithms and applications for networks of neurosynaptic cores. In Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN'13). doi: 10.1109/IJCNN.2013.6706746
60. Estes, W.K., and Suppes, P. (1959). *Foundations of statistical learning theory. II. The stimulus sampling model.* Stanford University, Applied Mathematics and Statistics Laboratory, Behavioral Sciences Division. doi: 10.1.1.398.2539
61. Farabet, C., Martini, B., Corda, B., Akselrod, P., Culurciello, E., and LeCun, Y. (2011). Neuflow: A runtime reconfigurable dataflow processor for vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 109-116. doi: 10.1109/CVPRW.2011.5981829
62. Farley, B., and Clark, W. (1954). Simulation of self-organizing systems by digital computer. *Information Theory, Transactions of the IRE Professional Group on* 4**,** 76-84. doi: 10.1109/TIT.1954.1057468
63. Faust, A. (2014). *Reinforcement learning and planning for preference balancing tasks.* Doctoral thesis, University of New Mexico.
64. Feldman, D.E. (2009). Synaptic mechanisms for plasticity in neocortex. Annual Reviews in Neuroscience, 32:33-55. doi:10.1146/annurev.neuro.051508.135516.
65. Felleman, D. J., Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex*, 1: 1-47. doi: 10.1093/cercor/1.1.1
66. Fitts, P.M., Weinstein, M., Rappaport, M., Anderson, N., and Leonard, J.A. (1956). Stimulus correlates of visual pattern recognition: a probability approach. *Journal of Experimental Psychology* 51**,** 1. doi: 10.1037/h0044302
67. French, R.S. (1954). Pattern recognition in the presence of visual noise. *Journal of Experimental Psychology* 47**,** 27. doi: 10.1037/h0058298
68. Fukushima, K. (1975). Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics* 20**,** 121-136. doi: 10.1007/BF00342633
69. Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks* 1**,** 119-130. doi: 10.1016/0893-6080(88)90014-7

70. Furber, S.B., Galluppi, F., Temple, S., and Plana, L.A. (2014). The SpiNNaker Project. *Proceedings of the IEEE* 102**,** 652-665. doi: 10.1109/Jproc.2014.2304638
71. Fusi, S., Del Guidice, P., and Amit, D.J. (2000). Neurophysiology of a VLI spiking neural network: LANN21. In *International Joint Conference on Neural Networks*, 121-126. doi: 10.1109/IJCNN.2000.861291
72. Gelly, S., Kocsis, L., Schoenauer, M., Sebag, M., Silver, D., Szepesvári, C., and Teytaud, O. (2012). The grand challenge of computer Go: Monte Carlo tree search and extensions. *Communications of the ACM* 55**,** 106-113. doi: 10.1145/2093548.2093574
73. Genov, R., and Cauwenberghs, G. (2003). Kerneltron: support vector" machine" in silicon. *IEEE Transactions on Neural Networks,* 14**,** 1426-1434. doi: 10.1109/Tnn.2003.816345
74. Gewaltig, M.-O. and Diesmann, M. (2007). NEST (NEural Simulation Tool). Scholarpedia, vol. 2, no. 4, p. 1430.
75. Gleeson, P., Crook, S., Cannon, R.C., Hines, M.L., Billings, G.O., Farinella, M., et al. (2010). NeuroML: a language for describing data driven models of neurons and networks with a high degree of biological detail. *PLoS Compotational Biology* 6, e1000815. doi:10.1371/journal.pcbi.1000815
76. Gu, S., and Rigazio, L. (2014). Towards deep neural network architectures robust to adversarial examples. *arXiv:1412.5068*
77. Hammerstrom, D. (2010). A Survey of Bio-Inspired and Other Alternative Architectures. In *Nanotechnology*. Editor Waser, R., Wiley-Series. doi: 10.1002/9783527628155.nanotech045
78. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., and Coates, A. (2014). DeepSpeech: Scaling up end-to-end speech recognition. *arXiv:1412.5567*
79. Hasler, J. and Marr, B. (2013). Finding a roadmap to achieve large neuromorphic hardware systems. *Frontiers in Neuroscience* 7, 118-. doi: 10.3389/fnins.2013.00118
80. Hawkins, J., Ahmad, S., Dubinsky, D. (2010). Hierarchical temporal memory including HTM cortical learning alogrithms. Technical Report, Numenta, Inc., Palo Alto. http://numenta.com/assets/pdf/whitepapers/hierarchical-temporal-memory-cortical-learning-algorithm-0.2.1-en.pdf
81. Hay, J.C., Lynch, B.E., and Smith, D.R. (1960). Mark I perceptron operators' manual. No. VG-1196-G-5. Cornell Aeronautical Lab Inc, Buffalo, NY
82. He, M., Liu, Y., Wang, X., Zhang, M.Q., Hannon, G.J., and Huang, Z.J. (2012). Cell-type-based analysis of microRNA profiles in the mouse brain. *Neuron* 73**,** 35-48. doi: 10.1016/j.neuron.2011.11.010
83. Hebb, D.O. (1949). *The organization of behavior: A neuropsychological approach.* John Wiley & Sons.
84. Hinton, G.E., and Salakhutdinov, R.R. (2006). Reducing the dimensionality of data with neural networks. *Science* 313**,** 504-507. doi: 10.1126/science.1127647
85. Hinton, G.E. (2012). A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade.* Springer Berlin Heidelberg. 599-619. doi: 10.1007/978-3-642-35289-8_32
86. Ho, T.K. (1998). The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 20**,** 832-844. doi: 10.1109/34.709601
87. Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press. Eds. Kremer, S.C. and Kolen, J.F. doi: 10.1.1.24.7321
88. Holler, M., Park, C., Diamond, J., Santoni, U., Tam, S., Glier, M., et al. (1992). A high performance adaptive classifier using radial basis functions. In *Submitted to Government Microcircuit Applications Conference*, 1-4.

89. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* 79**,** 2554-2558. doi: 10.1073/pnas.79.8.2554.
90. Hopfield, J.J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences* 81**,** 3088-3092. doi: 10.1073/pnas.81.10.3088.
91. Hu, H., Gan, J., and Jonas, P. (2014). Fast-spiking, parvalbumin+ GABAergic interneurons: From cellular design to microcircuit function. *Science* 345**,** 1255263. doi: 10.1126/science.1255263
92. Hubel, D.H., and Wiesel, T.N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology* 148**,** 574. doi: 10.1113/jphysiol.1959.sp006308
93. Hunt, E.B., Marin, J., and Stone, P.J. (1966). Experiments in induction. New York, Academic Press.
94. Indiveri, G., Linares-Barranco, B., Hamilton, T.J., van Schaik, A., Etienne-Cummings, R., Delbruck, T., Liu, S.C., Dudek, P., Hafliger, P., Renaud, S., Schemmel, J., Cauwenberghs, G., Arthur, J., Hynna, K., Folowosele, F., Saighi, S., Serrano-Gotarredona, T., Wijekoon, J., Wang, Y., and Boahen, K. (2011). Neuromorphic silicon neuron circuits. *Front Neurosci* 5**,** 73. doi: 10.3389/fnins.2011.00073
95. Indiveri, G., Legenstein, R., Deligeorgis, G., and Prodromakis, T. (2013). Integration of nanoscale memristor synapses in neuromorphic computing architectures. *Nanotechnology* 24**,** 384010. doi: 10.1088/0957-4484/24/38/384010
96. Insel, T.R., Landis, S.C., and Collins, F.S. (2013). Research priorities. The NIH BRAIN Initiative. *Science* 340**,** 687-688. doi: 10.1126/science.1239276
97. Ivanciuc, O. (2007). Applications of support vector machines in chemistry. *Reviews in Computational Chemistry* 23**,** 291. doi: 10.1002/9780470116449.ch6
98. Izhikevich, E.M. (2007). Solving the distal reward problem through linkage of STDP and dopamine signaling. Cerebral Cortex 17, 2443-2452. doi: 10.1093/cercor/bhl152
99. Izhikevich, E.M., and Edelman, G.M. (2008). Large-scale model of mammalian thalamocortical systems. *Proc Natl Acad Sci U S A* 105**,** 3593-3598. doi: 10.1073/pnas.0712231105
100. Jackel, L., Boser, B., Denker, J., Graf, H., Le Cun, Y., Guyon, I., Henderson, D., Howard, R., Hubbard, W., and Solla, S. (1990). Hardware requirements for neural-net optical character recognition. In *International Joint Conference on Neural Networks*, 855-861. doi: 10.1109/IJCNN.1990.137801
101. Jackson, B.L., Rajendran, B., Corrado, G.S., Breitwisch, M., Burr, G.W., Cheek, R., et al. (2013). Nanoscale Electronic Synapses Using Phase Change Devices. *ACM Journal on Emerging Technologies in Computing Systems* 9**,** 12. doi: 10.1145/2463585.2463588
102. Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148**,** 34.
103. Jaeger, H., and Haas, H. (2004). Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* 304**,** 78-80. doi: 10.1126/science.1091277
104. Jo, S.H., Chang, T., Ebong, I., Bhadviya, B.B., Mazumder, P., and Lu, W. (2010). Nanoscale Memristor Device as Synapse in Neuromorphic Systems. *Nano Letters* 10**,** 1297-1301. doi: 10.1021/nl904092h
105. Jones, N. (2014). The learning machines. *Nature* 505, 146-148. doi:10.1038/505146a
106. Kaelbling, L.P., Littman, M.L., and Moore, A.W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4**,** 237-285. doi: 10.1.1.134.2462
107. Kaneko, Y., Nishitani, Y., and Ueda, M. (2014). Ferroelectric Artificial Synapses for Recognition of a Multishaded Image. *IEEE Transactions on Electron Devices* 61**,** 2827-2833. doi: 10.1109/Ted.2014.2331707

108. Kaneta, Y., Yoshizawa, S., Minato, S., and Arimura, H. (2011). High-Speed String and Regular Expression Matching on FPGA. In *Asia-Pacific Signal Information Processing Association Annu. Summit Conf.,* Xi'an, China.

109. Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3128-3137.

110. Kawasetsu, T., Ishida, R., Sanada, T., Okuno, H. (2014). A hardware system for emulating the early vision utilizing a silicon retina and SpiNNaker chips. *Proceedings of the 2014 IEEE Biomedical Circuits and Systems Conference,* 552-555. doi: 10.1109/BioCAS.2014.6981785

111. Kelley, H.J. (1960). Gradient Theory of Optimal Flight Paths. *ARS Journal* 30**,** 947-954. doi: 10.2514/8.5282

112. Kent, A.D., and Worledge, D.C. (2015). A new spin on magnetic memories. *Nat Nanotechnol* 10**,** 187-191. doi: 10.1038/nnano.2015.24

113. Kishi, T., Yoda, H., Kai, T., Nagase, T., Kitagawa, E., Yoshikawa, M., et al. (2008). Lower-current and fast switching of a perpendicular TMR for high speed and high density spin-transfer-torque MRAM. In *IEEE International Electron Devices Meeting*, 1-4. doi: 10.1109/IEDM.2008.4796680

114. Kober, J., and Peters, J. (2012). Reinforcement learning in robotics: A survey. In *Reinforcement Learning*, Springer, 579-610. doi: 10.1.1.366.5647

115. Kozicki, M.N., Gopalan, C., Balakrishnan, M., Park, M., and Mitkova, M. (2004). Nonvolatile memory based on solid electrolytes. In *Non-Volatile Memory Technology Symposium*, 10-17. doi: 10.1109/NVMT.2004.1380792

116. Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 1097-1105. doi: 10.1.1.299.205

117. Krichmar, J.L., Coussy, P., Dutt, N. (2015). Large-scale spiking neural networks using neuromorphic hardware compatible models. *ACM Journal on Emerging Technologies in Computing Elements* 11, doi: 10.1145/2629509

118. Kumar, S. (2013). Introducing Qualcomm Zeroth Processors: Brain-inspired computing. https://www.qualcomm.com/news/onq/2013/10/10/introducing-qualcomm-zeroth-processors-brain-inspired-computing.

119. Kuzum, D., Yu, S., and Wong, H.S. (2013). Synaptic electronics: materials, devices and applications. *Nanotechnology* 24**,** 382001. doi: 10.1088/0957-4484/24/38/382001.

120. Lake, B.M., Salakhutdinov, R., and Tenenbaum, J.B. Human-level concept learning through probabilistic program induction. *Science* 350, 1332-1338.

121. Le, Q.V. (2013). Building high-level features using large scale unsupervised learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 8595-8598. doi: 10.1.1.261.605

122. LeCun, Y. (1985). Une procédure d'apprentissage pour réseau a seuil asymmetrique (a Learning Scheme for Asymmetric Threshold Networks). In *Proceedings of Cognitiva*, 599-604.

123. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation* 1, 541–551. doi:10.1162/neco.1989.1.4.541

124. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324. doi:10.1109/5.726791

125. Lee, M.J., Lee, C.B., Lee, D., Lee, S.R., Chang, M., Hur, J.H., et al. (2011). A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta2O5-x/TaO2-x bilayer structures. *Nature Materials* 10**,** 625-630. doi: 10.1038/NMAT3070

126. Liao, Y., Rubinsteyn, A., Power, R., and Li, J. (2013). Learning Random Forests on the GPU.

127. Lyon, R.F., and Mead, C. (1988). An Analog Electronic Cochlea. *IEEE Transactions on Acoustics Speech and Signal Processing* 36**,** 1119-1134. doi: 10.1109/29.1639
128. Maass, W., Natschlager, T., and Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation* 14**,** 2531-2560. doi: 10.1162/089976602760407955
129. Maetschke, S.R., and Ragan, M.A. (2014). Characterizing cancer subtypes as attractors of Hopfield networks. *Bioinformatics* 30**,** 1273-1279. doi: 10.1093/bioinformatics/btt773
130. Mai, V.H., Moradpour, A., Senzier, P.A., Pasquier, C., Wang, K., Rozenberg, et al. (2015). Memristive and neuromorphic behavior in a LixCoO2 nanobattery. *Scientific Reports* 5. doi: Artn 7761 10.1038/Srep07761
131. Mandal, S., El-Amin, A., Alexander, K., Rajendran, B., and Jha, R. (2014). Novel synaptic memory device for neuromorphic computing. *Sci Rep* 4**,** 5333. doi: 10.1038/srep05333
132. Markram, H. (2006). The blue brain project. *Nat Rev Neurosci* 7**,** 153-160. doi: 10.1038/nrn1848
133. Markram, H. (2012). The human brain project. *Sci Am* 306**,** 50-55. doi:10.1038/scientificamerican0612-50
134. Markram, H., Muller, E., Ramaswamy, S., Reimann, M.W., Abdellah, M., Sanchez, C.A., et al. (2015). Reconstruction and simulation of neocrotical microcircuitry. *Cell* 163, 456-492. doi: 10.1016/j.cell.2015.09.029
135. Mayr, C., Partzsch, J., Noack, M., Hänzsche, S., Scholze, S., Höppner, S., Ellguth, G., and Schüffny, R. (2015). A biological-realtime neuromorphic system in 28 nm CMOS using low-leakage switched capacitor circuits. *IEEE Transactions on Biomedical Circuits and Systems*. doi: 10.1109/TBCAS.2014.2379294
136. McCulloch, W.S., and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics* 5**,** 115-133. doi: 10.1007/BF02478259
137. Mead, C.A., and Mahowald, M.A. (1988). A Silicon Model of Early Visual Processing. *Neural Networks* 1**,** 91-97. doi: 10.1016/0893-6080(88)90024-X
138. Merolla, P., Arthur, J., Akopyan, F., Imam, N., Manohar, R., and Modha, D.S. (2011). A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm. In *IEEE Custom Integrated Circuits Conference*, 1-4. doi: 10.1109/CICC.2011.6055294
139. Merolla, P.A., Arthur, J.V., Alvarez-Icaza, R., Cassidy, A.S., Sawada, J., Akopyan, F., et al. (2014). Artificial brains. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345**,** 668-673. doi: 10.1126/science.1254642
140. Mickel, P.R., Lohn, A.J., James, C.D., and Marinella, M.J. (2014). Isothermal switching and detailed filament evolution in memristive systems. *Adv Mater* 26**,** 4486-4490. doi: 10.1002/adma.201306182
141. Minsky, M.L. (1952). A neural-analogue calculator based upon a probability model of reinforcement. In *Harvard University Pychological Laboratories Internal Report.* Cambridge, Massachusetts.
142. Minsky, M. (1961). Steps toward Artificial Intelligence. *Proceedings of the Institute of Radio Engineers* 49**,** 8-&. doi: 10.1109/Jrproc.1961.287775
143. Minsky, M., and Papert, S. (1969). Perceptron: an introduction to computational geometry. *The MIT Press, Cambridge, expanded edition* 19**,** 88. doi: 10.1126/science.165.3895.780
144. Mitra, S., Fusi, S., and Indiveri, G. (2009). Real-Time Classification of Complex Patterns Using Spike-Based Learning in Neuromorphic VLSI. *IEEE Transactions on Biomedical Circuits and Systems* 3**,** 32-42. doi: 10.1109/Tbcas.2008.2005781
145. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518**,** 529-533. doi: 10.1038/nature14236

146. Montague, P. R., Hyman, S. E. and Cohen, J. D. (2004). Computational roles for dopamine in behavioral control. Nature 431, 760-767. doi:10.1038/nature03015

147. Murphy, K. P. (2002). Dynamic bayesian networks: representation, inference and learning. Doctoral dissertation, University of California, Berkeley.

148. Nageswaran, J.M., Dutt, N., Krichmar, J.L., Nicolau, A. and Veidenbaum, A.V. (2009). A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors. *Neural Networks*, 22(5), 791–800. doi:10.1016/j.neunet.2009.06.028

149. Neftci, E., Binas, J., Rutishauser, U., Chicca, E., Indiveri, G., and Douglas, R.D. (2013). Synthesizing cognition in neuromorphic electronic systems. *Proceedings of the National Academy of Sciences*, 110, E3468-E3476. doi: 10.1073/pnas.1212083110

150. Neher, E., Sakmann, B., Steinbach, J.H. (1978). The extracellular patch clamp: a method for resolving currents through individual open channels in biological membranes. *Pflügers Archiv* 375, 219–28. doi:10.1007/BF00584247

151. Nowotny, T. (2010). Parallel implementation of a spiking neuronal network model of unsupervised olfactory learning on NVidia CUDA. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN‘10), 1–8. doi: 10.1109/IJCNN.2010.5596358

152. O'Keefe, J. (1976). Place units in the hippocampus of the freely moving rat. *Experimental neurology* 51, 78–109. doi: 10.1016/0014-4886(76)90055-8

153. O'Keefe, J., Recce, M. L. (1993). Phase relationship between hippocampal place units and the EEG theta rhythm. Hippocampus 3, 317–30. doi:10.1002/hipo.450030307

154. Okuno, H., Hasegawa, J., Sanada, T., Yagi, T. (2015). Real-time emulator for reproducing graded potentials in vertebrate retina. IEEE Transactions on Biomedical Circuits and Systems 9, 284-295. doi: 10.1109/TBCAS.2014.2327103

155. Osman, H.E. (2009). Hardware-Based Solutions Utilizing Random Forests for Object Recognition. In *Advances in Neuro-Information Processing*. Springer, 760-767. doi: 10.1007/978-3-642-03040-6_93

156. Packer, A.M., Russell, L.E., Dalgleish, H.W.P., Häusser, M. (2015). Simultaneous all-optical manipulation and recording of neural circuit activity with cellular resolution in vivo. *Nature Methods* 12, 140-146. doi:10.1038/nmeth.3217

157. Paik, J.K., and Katsaggelos, A.K. (1992). Image restoration using a modified Hopfield network. *IEEE Transactions on Image Processing* 1**,** 49-63. doi: 10.1109/83.128030

158. Paquot, Y., Duport, F., Smerieri, A., Dambre, J., Schrauwen, B., Haelterman, M., et al. (2012). Optoelectronic reservoir computing. *Sci Rep* 2**,** 287. doi: 10.1038/srep00287

159. Pavlov, I.P., and Gantt, W. (1928). *Lectures on conditioned reflexes: Twenty-five years of objective study of the higher nervous activity (behaviour) of animals.* Liverwright Publishing, New York, NY.

160. Payer, G., McCormick, C., and Harang, R. (2014). Applying hardware-based machine learning to signature-based network intrusion detection. In *SPIE Sensing Technology+ Applications*: International Society for Optics and Photonics, 91190C-91190C-91116. doi: 10.1117/12.2052548

161. Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29, 241-288. doi:10.1016/0004-3702(86)90072-X

162. Prezioso, M., Merrikh-Bayat, F., Hoskins, B.D., Adam, G.C., Likharev, K.K., and Strukov, D.B. (2015). Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* 521**,** 61-64. doi: 10.1038/nature14441

163. Price, C.J. (2012). A review and synthesis of the first 20years of PET and fMRI studies of heard speech, spoken language and reading. *Neuroimage* 62**,** 816-847. doi: 10.1016/j.neuroimage.2012.04.062

164. Qiao, N., Mostafa, H., Corradi, F., Osswald, M., Stefanini, F., Sumislawska, D., and Indiveri, G. (2015). A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses. Frontiers in Neuroscience 9, 141. doi: 10.3389/fnins.2015.00141

165. Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning* 1**,** 81-106. doi: 10.1023/A:1022643204877

166. Quinlan, J.R. (1990). Learning Logical Definitions from Relations. *Machine Learning* 5**,** 239-266. doi: 10.1007/Bf00117105

167. Rabiner, L.R. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE* 77: 257–286. doi:10.1109/5.18626

168. Rachmuth, G., Shouval, H.Z., Bear, M.F., and Poon, C. (2011). A biophysically-based neuromorphic model of spike rate- and timing-dependent plasticity. *Proceedings of the National Academy of Science* 108, E1266-E1274. doi: 10.1073/pnas.1106161108

169. Rahimi Azghadi, M., Al-Sarawi, S., Abbott, D., and Iannella, N. (2013). A neuromorphic VLSI design for spike timing and rate based synaptic plasticity. *Neural Networks* 45**,** 70-82. doi: 10.1016/j.neunet.2013.03.003

170. Ramakrishnan, S., Hasler, P.E., and Gordon, C. (2011). Floating gate synapses with spike-time-dependent plasticity. IEEE Transactions on Biomedical Circuits and Systems 5, 244-252.

171. Rangel, L.M., Alexander, A.S., Aimone, J.B., Wiles, J., Gage, F.H., Chiba, A.A. and Quinn, L.K. (2014). Temporally selective contextual encoding in the dentate gyrus of the hippocampus. Nature Communications 5, 3181. doi:10.1038/ncomms4181

172. Raoux, S., Burr, G.W., Breitwisch, M.J., Rettner, C.T., Chen, Y.C., Shelby, R.M., et al. (2008). Phase-change random access memory: A scalable technology. *IBM Journal of Research and Development* 52**,** 465-479. doi: 10.1147/rd.524.0465

173. Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev* 65**,** 386-408

174. Rosenblatt, F. (1960). Perceptron Simulation Experiments. *Proceedings of the Institute of Radio Engineers* 48**,** 301-309. doi: 10.1109/Jrproc.1960.287598

175. Rosenblatt, F. (1962). Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Washington, Spartan Books

176. Ross, T. (1933). Machines that think. *Health* 243**,** 248.

177. Rothganger, F., Warrender, C.E., Trumbo, D., and Aimone, J.B. (2014) *Frontiers in Neural Circuits* 8, 1-12. doi: 10.3389/fncir.2014.00001

178. Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature* 323, 533-536. doi:10.1038/323533a0

179. Saighi, S., Mayr, C.G., Serrano-Gotarredona, T. Schmidt, H., lecerf, G., Tomas, J. et al. (2015). Plasticity in memristive devices for spiking neural networks. *Frontiers in Neuroscience* 9, 1-16. doi: 10.3389/fnins.2015.00051

180. Schemmel, J., Bruderle, D., Grubl, A., Hock, M., Meier, K., and Millner, S. (2010). A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, 1947-1950. doi: 10.1109/ISCAS.2010.5536970

181. Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural Networks* 61**,** 85-117. doi: 10.1016/j.neunet.2014.09.003

182. Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *arXiv:1503.03832.*

183. Schürmann, F., Meier, K., and Schemmel, J. (2004). Edge of chaos computation in mixed-mode VLSI- "a hard liquid." In *Advances in Neural Information Processing Systems* 17, Cambridge, MA, MIT Press. 1201-1208.

184. Selfridge, O.G. (1955). Pattern recognition and modern computers. In *Proceedings of the Western Joint Computer Conference*, 91-93. doi:10.1109/AFIPS.1955.20

185. Serrano-Gotarredona, T., Linares-Barranco, B., Galluppi, F., Plana, L., Furber, S. (2015). ConvNets Experiments on SpiNNaker. *IEEE International Symposium on Circuits and Systems*, 2405-2408. doi: 10.1109/ISCAS.2015.7169169

186. Shannon, C.E. (1951). Presentation of a maze-solving machine. In *8th Conf. of the Josiah Macy Jr. Found.(Cybernetics)*, 173-180.

187. Sharp, T. (2008). Implementing decision trees and forests on a GPU. In *Computer Vision-ECCV*. Springer, 595-608. doi: 10.1007/978-3-540-88693-8_44

188. Shelby, R.M., Burr, G.W., Boybat, I., and di Nolfo, C̄. (2015). Non-volatile memory as hardware synapse in neuromorphic computing: A first look at reliability issues. In *IEEE International Reliability Physics Symposium*, 6A. 1.1-6A. 1.6. doi: 10.1109/IRPS.2015.7112755

189. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, LL., van den Driessche et al. (2016) Mastering the game of Go with deep neurla networks and tree search. *Nature* 529, 484-492. doi:10.1038/nature16961

190. Skinner, B. (1933). The rate of establishment of a discrimination. *The Journal of General Psychology* 9**,** 302-350. doi: 10.1080/00221309.1933.9920939

191. Sohal, V.S., Zhang, F., Yizhar, O., and Deisseroth, K. (2009). Parvalbumin neurons and gamma rhythms enhance cortical circuit performance. *Nature* 459, 698-702. doi:10.1038/nature07991

192. Song, I., Kim, H.-J., and Jeon, P.B. (2014). Deep learning for real-time robust facial expression recognition on a smartphone. In *International Conference on Consumer Electronics*, 564-567. doi: 10.1109/ICCE.2014.6776135

193. Stefanini, F., Neftci, E.O., Sheik, S., and Indiveri, G. (2014). PyNCS: a microkernel for high-level definition and configuration of neuromorphic electronic systems. Frontiers in *Neuroinformatics* 8, 73-77. doi: 10.3389/fninf.2014.00073

194. Stevenson, I.H., and Kording, K.P. (2011). How advances in neural recording affect data analysis. *Nature Neuroscience* 14**,** 139-142. doi: 10.1038/nn.2731

195. Stewart, T.C., and Eliasmith, C. (2014). Large-Scale Synthesis of Functional Spiking Neural Circuits. *Proceedings of the IEEE* 102**,** 881-898. doi: 10.1109/Jproc.2014.2306061

196. Strukov, D.B., Snider, G.S., Stewart, D.R., and Williams, R.S. (2008). The missing memristor found. *Nature* 453**,** 80-83. doi: 10.1038/nature06932

197. Sun, R., Zhang, X., and Mathews, R. (2006). Modeling meta-cognition in a cognitive architecture. Cognitive Systems Research 7, 327-338. doi: 10.1016/j.cogsys.2005.09.001

198. Sutton, R.S., and Barto, A.G. (1998). *Reinforcement learning: An introduction.* MIT press, Cambridge. doi: 10.1109/TNN.1998.712192

199. Szepesvari, C. (2010). Algorithms for Reinforcement Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning.* Morgan and Claypool Publishers. doi: 10.2200/S00268ED1V01Y201005AIM009

200. Takagi, K., Tanaka, K., Izumi, S., Kawaguchi, H., and Yoshimoto, M. (2014). A Real-time Scalable Object Detection System using Low-power HOG Accelerator VLSI. *Journal of Signal Processing Systems for Signal Image and Video Technology* 76**,** 261-274. doi: 10.1007/s11265-014-0870-7

201. Talmadge, C. L., Tubis, A., Long, G.R. and Piskorski, P. (1998). Modeling otoacoustic emission and hearing threshold fine structures. J. Acoust. Soc. Am. 104, 1517-1543. doi: 10.1121/1.424364

202. Tappert, C. C. (2011). Rosenblatt's contributions. http://csis.pace.edu/~ctappert/srd2011/rosenblatt-contributions.htm

203. Thomas, P. Grübl, A. Jeltsch, S., Müller, E., Müller, P., Petrovici, M. A., Schmuker, M., Brüderle, D., Schemmel, J., and Meier, K. (2013). Six networks on a universal neuromorphic computing substrate. *Frontiers in Neuroscience* 7, 11. doi: 10.3389/fnins.2013.00011

204. Van Essen, B., Macaraeg, C., Gokhale, M., and Prenger, R. (2012). Accelerating a random forest classifier: Multi-core, GP-GPU, or FPGA? In *IEEE 20th Annual International Symposium on Field-Programmable Custom Computing Machines,* 232-239. doi: 10.1109/FCCM.2012.47

205. Vandoorne, K., Mechet, P., Van Vaerenbergh, T., Fiers, M., Morthier, G., Verstraeten, D., et al. (2014). Experimental demonstration of reservoir computing on a silicon photonics chip. *Nat Commun* 5**,** 3541. doi: 10.1038/ncomms4541

206. Vapnik, V. (2000). *The nature of statistical learning theory.* Springer-Verlag, New York, doi: 10.1007/978-1-4757-3264-1

207. Verstraeten, D., Schrauwen, B., D'Haene, M., and Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural Networks* 20**,** 391-403. doi: 10.1016/j.neunet.2007.04.003

208. Villringer, A., and Chance, B. (1997). Non-invasive optical spectroscopy and imaging of human brain function. *Trends Neurosci* 20**,** 435-442. doi:10.1016/S0166-2236(97)01132-6

209. Vineyard, C.M., Verzi, S.J., James, C.D., Aimone, J.B., and Heileman, G.L. (2015) Repeated play of the SVM game as a means of adaptive classification. In *International Joint Conference on Neural Networks*, 1-8. doi: 10.1109/IJCNN.2015.7280729

210. Vineyard, C.M., Verzi, S.J., James, C.D., Aimone, J.B. (2016). Quantifying neural information content: a case study of the impact of hippocampal adult neurogenesis. International Joint Conference on Neural Networks (IJCNN), 5181-5188. doi: 10.1109/IJCNN.2016.7727884.

211. Vineyard, C.M., Verzi, S.J., James, C.D., Aimone, J.B., and Heileman, G.L. (2015) MapReduce SVM game. In *International Neural Network Society Conference on Big Data*, Procedia Computer Science 53, 298-307. doi: 10.1016/j.procs.2015.07.307

212. Watts, L., Kerns, D.A., Lyon, R.F., and Mead, C.A. (1992). Improved Implementation of the Silicon Cochlea. *IEEE Journal of Solid-State Circuits* 27**,** 692-700. doi: 10.1109/4.133156

213. Wei, Z., Kanzawa, Y., Arita, K., Katoh, Y., Kawai, K., Muraoka, et al. (2008). Highly reliable TaOx ReRAM and direct evidence of redox reaction mechanism. In *IEEE International Electron Devices Meeting*, 1-4. doi: 10.1109/IEDM.2008.4796676

214. Werbos, P.J. (1990). Backpropagation throught time: what it does and how to do it. *Proceedings of the IEEE* 78, 1550-1560. doi: 10.1109/5.58337

215. White, B.A., and Elmasry, M.I. (1992). The digi-neocognitron: a digital neocognitron neural network model for VLSI. *IEEE Trans Neural Netw* 3**,** 73-85. doi: 10.1109/72.105419

216. Widrow, B. (1960b). *Adaptive "adaline" Neuron Using Chemical "memistors".* Office of Naval Research Technical Report, Stanford University - Stanford Solid State Electronics Laboratory.

217. Widrow, B., and Hoff, M.E. (1960a). Adaptive switching circuits. *Institute of Radio Engineers WESCON Convention Record*, 4, 96-104.

218. Winter, R., and Widrow, B. (1988). Madaline Rule II: a training algorithm for neural networks. In *IEEE International Conference on Neural Networks*, 401-408. doi: 10.1109/ICNN.1988.23872

219. Wong, H.P., Raoux, S., Kim, S., Liang, J., Reifenberg, J.P., Rajendran, B., et al. (2010). Phase change memory. *Proceedings of the IEEE,* 98**,** 2201-2227. doi: 10.1109/JPROC.2010.2070050

220. Wong, H.S., and Salahuddin, S. (2015). Memory leads the way to better computing. *Nat Nanotechnol* 10**,** 191-194. doi: 10.1038/nnano.2015.29

221. Wu, R., Yan, S., Shan, Y., Dang, Q., and Sun, G. (2015). Deep image: Scaling up image recognition. *arXiv:1501.02876.*

222. Yang, Y.H.E., and Prasanna, V.K. (2012). High-Performance and Compact Architecture for Regular Expression Matching on FPGA. *IEEE Transactions on Computers* 61**,** 1013-1025. doi: 10.1109/Tc.2011.129

223. Yang, W., Jin, Z., Thiem, C., Wysocki, B., Shen, D., and Chen, G. (2014). Autonomous target tracking of UAVs based on low-power neural network hardware. In *SPIE Sensing Technology+ Applications*: International Society for Optics and Photonics, 91190P-91190P-91199. doi: 10.1117/12.2054049

224. Zatorre, R. J., Fields, R. D. and Johansen-Berg, H. (2012). Plasticity in gray and white: neuroimaging changes in brain structure during learning. Nature Neuroscience 15, 528-536. doi:10.1038/nn.3045.

225. Zhou, K., Fox, J.J., Wang, K., Brown, D.E., and Skadron, K. (2015). Brill Tagging on the Micron Automata Processor. In *IEEE International Conference on Semantic Computing*, 236-239. doi: 10.1109/ICOSC.2015.7050812

226. Zito, K. and Svoboda, K. (2002). Activity-dependent synaptogenesis in the adult Mammalian cortex. Neuron 35, 1015-1017. doi: 10.1016/S0896-6273(02)00903-0

Figure Captions

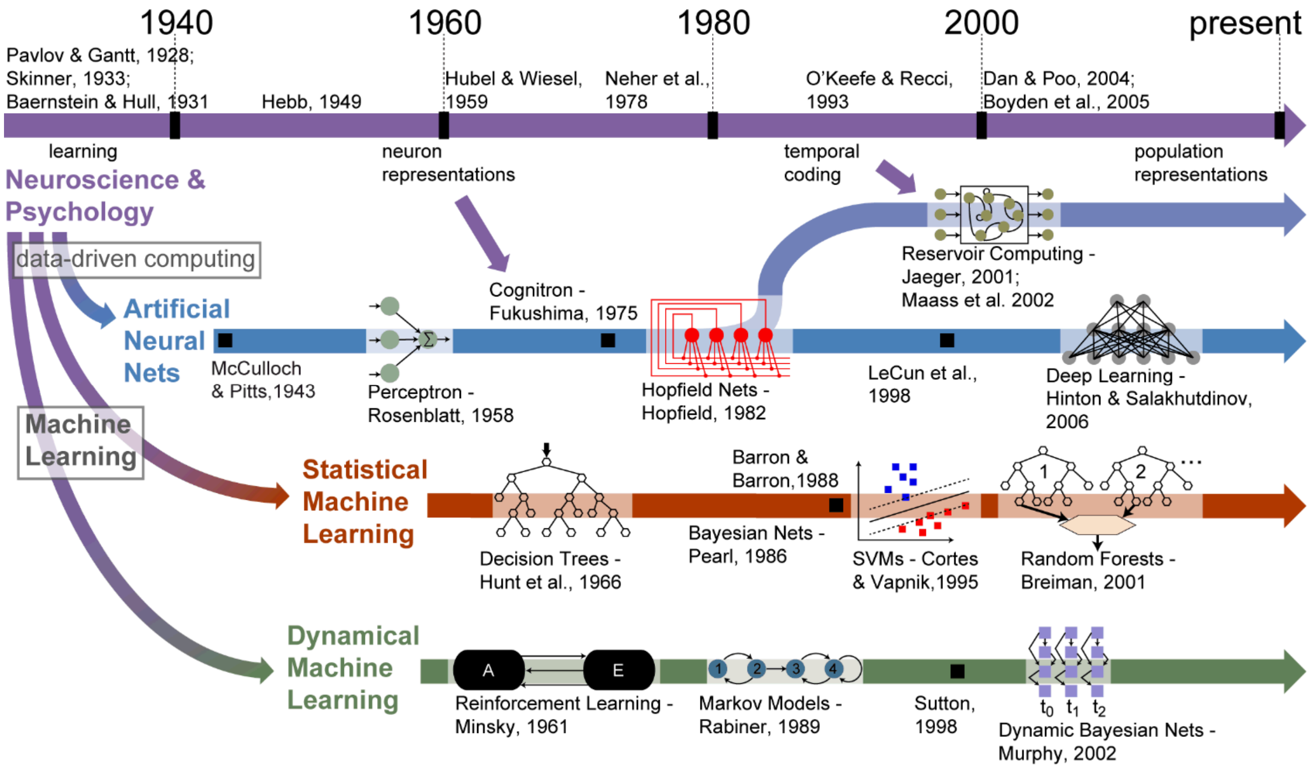

Figure 1 (color): Historical timeline of neuroscience and psychology and the influence of the fields on neuromorphic and neural-inspired algorithms and hardware research.

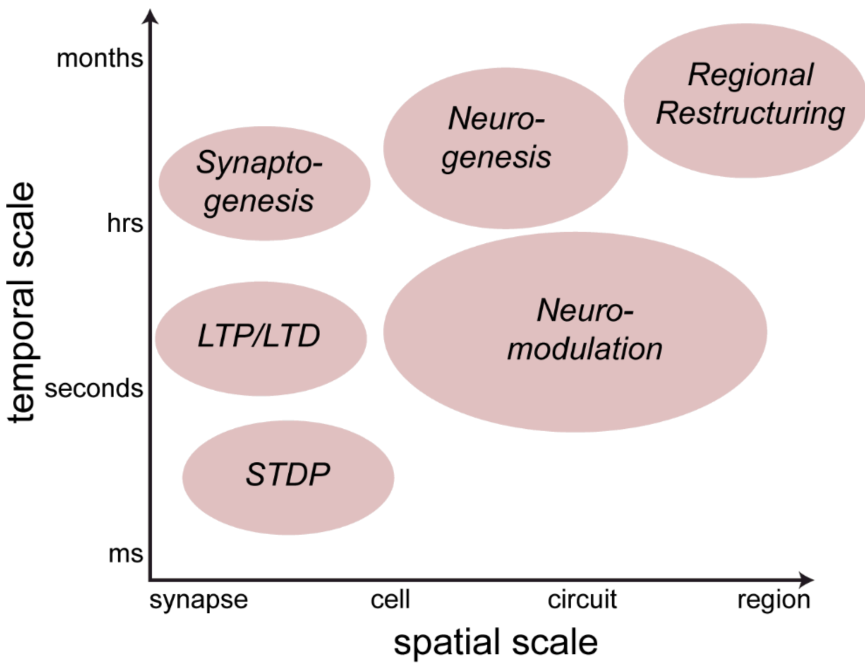

Figure 2 (color): Plasticity mechanisms that impact computation in neurobiological systems. Spike timing-dependent plasticity (STDP) occurs rapidly at the synapse-level while long-term potentiation and depression (LTP, LTD) take longer to occur (Feldman 2009). The production of new synapses and neurons (synaptogenesis and neurogenesis) and the regional restructuring of neurobiological tissue take place over hours to months (Zito and Svoboda, 2002; Aimone et al., 2010; Zatorre et al., 2012). Neuromodulators such as dopamine act over a wide range of spatial scales to impact phenomena including reinforcement learning and behavior (Du et al., 2016; Montague et al., 2004).