Towards Reverse Engineering The Brain: Modeling Abstractions and Simulation Frameworks

Jayram Moorkanikara Nageswaran[1], Micah Richert[2], Nikil Dutt[1], Jeffrey L Krichmar[1, 2]
[1]Department of Computer Science, [2]Department of Cognitive Sciences
University of California - Irvine
Irvine, CA USA
jmoorkan@uci.edu , mrichert@uci.edu , dutt@uci.edu , jkrichma@uci.edu

Abstract—**Biological neural systems are well known for their robust and power-efficient operation in highly noisy environments. Biological circuits are made up of low-precision, unreliable and massively parallel neural elements with highly reconfigurable and plastic connections. Two of the most interesting properties of the neural systems are its self-organizing capabilities and its template architecture. Recent research in spiking neural networks has demonstrated interesting principles about learning and neural computation. Understanding and applying these principles to practical problems is only possible if large-scale spiking neural simulators can be constructed. Recent advances in low-cost multiprocessor architectures make it possible to build large-scale spiking network simulators. In this paper we review modeling abstractions for neural circuits and frameworks for modeling, simulating and analyzing spiking neural networks.**

Keywords-Spiking neural networks, GPU, vision, computational neuroscience, parallel processing, synapse

I. Introduction

Understanding the principles of brain processing by reverse-engineering neural circuits and computational modeling is one of the biggest challenges of the 21st century [1]. These computational models provide the promise of practical applications in many domains, including vision, navigation, and decision-making. Anatomically, the human brain is divided into three regions: the hindbrain, mind-brain and forebrain [2]. The forebrain is further divided into two stages: the thalamic region and cerebrum. Most concepts discussed in this paper deal with the cerebral cortex, the outermost region of the cerebrum that is responsible for many functions unique to humans, including memory, attention, thought, language, etc. Recent developments in multi-electrode recording and brain mapping techniques (e.g., fMRI) have generated impressive amounts of data on different aspects of brain circuitry. These developments have spawned many approaches to understanding multiple aspects of brain-circuitry (Figure 1).

At the circuit level, the human brain has an estimated 100 billion neural processing elements (neurons) and about 10^{15} synaptic connections [3]-[4]. Each neuron is a sophisticated analog processor that not only integrates information from other neurons but also exhibits complex internal dynamics. Neurons communicate with other neurons via all-or-none digital signals called action potentials or spikes. These spikes are propagated using long-range fibers called axons. The neurons can be either excitatory or inhibitory, that is they increase or decrease the spiking or firing capability of downstream neurons. The synaptic connections between neurons exhibit different kinds of adaptation (or *plasticity*) based on the firing pattern of the "sending" (pre-synaptic) neuron and the "receiving" (post-synaptic) neuron. Several excellent overviews of neural circuit models have been published from a computer engineering perspective [5]-[8].

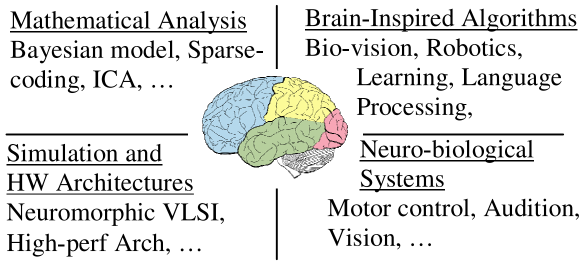
Mathematical Analysis
Bayesian model, Sparse-coding, ICA, …
Brain-Inspired Algorithms
Bio-vision, Robotics, Learning, Language Processing,
Simulation and HW Architectures
Neuromorphic VLSI, High-perf Arch, …
Neuro-biological Systems
Motor control, Audition, Vision, …

Figure 1: Brain-based research challenges

In this overview paper we focus on modeling abstractions, simulation frameworks and hardware (HW) architectures for modeling brain circuitry (Figure 1). Section II analyzes brain circuits from a VLSI perspective. Section III discusses different types of models used to study properties and principles of brain-circuits. Section IV focuses on spiking neural network (SNN) brain circuit models and analyzes their properties. Section V presents HW platforms for simulation of large-scale SNNs, with a focus on high-performance graphics processors. Section VI outlines a CAD-based software framework for development of large-scale SNN models, and Section VII concludes with future challenges and directions.

II. VLSI Perspectives on Brain circuits

While tremendous advances have been made in neuroscience, reverse engineering the brain is still a grand challenge with researchers proposing competing hypotheses to explain the nature of coding, energy efficiency, and speed of processing by brain circuits. But clear parallels can be drawn between brain circuits and VLSI circuits [5], and these can be exploited to enable the simulation of large-scale, biologically realistic brain circuits.

A. *Brain-network organization*

Studies on brain networks have revealed interesting details about macroscopic and microscopic structures within the brain [9][10]. At a macroscopic level the brain appears to consist of highly-clustered circuits performing similar functions that connect to other clusters via hubs in order to reduce the overall path length. These "small-world networks" are predominantly developed in the brain mainly to conserve wiring cost and also to enable an efficient way to transfer information between different brain regions [10].

Consistent with the "small-world network", the brain also exhibits a hierarchical organization for sensory processing. Sensory neural systems organize themselves with primitive features in the early stages to more and more complex features at the later stages of hierarchy [11]. Many interesting models have been developed to exploit this hierarchical organization for object recognition [12][13].

At a microscopic level, the columnar organization of cortical circuits offers a simple model to support general purpose processing [14]. A typical *cortical column* has hundreds to thousands of highly, and stereotypically interconnected neurons that functions as a stable unit, performing simple computations and competing with neighboring columns. This columnar abstraction of cortical processing has been applied as a general template for various pattern recognition tasks [15]. Further, for sensory systems, topologies are preserved such that neighboring regions of a sensory space (e.g., vision) are processed by neighboring cortical columns. These topological maps are also seen in the regions of the brain involved with other functions such as auditory processing and motor control.

B. Speed of Processing

Using rapid visual categorization experiments it was determined that humans can process natural images in 150ms [16]. Such speed of processing is mainly achieved through massive, parallel processing pathways for visual recognition [17]. One might wonder how a signal can be transmitted let alone processed so quickly using spikes in a noisy, clock-free environment. A recent hypothesis [16] suggests that vision processing can be modeled by a wave of spike propagation through a hierarchy of layers. Computation can be performed as soon as a sub-population of nearly coincident pre-synaptic spikes reaches a layer. This mode of computation is quite similar to asynchronous logic design, where a pulse representation is used to convey results of the computation.

C. Energy Efficiency

Nature has evolved an energy-efficient system that is 5-6 orders of magnitude more efficient than typical digital systems. A number of factors appear to contribute to the ultra-low power operation in neural systems such as sub-threshold spiking operations [5], sparse-energy efficient codes for signaling [9], proper balance of analog computation (for energy efficiency), and digital signaling (for signal restoration and reduced noise propagation) [18]. Many interesting ultra-low power circuits have been developed that use these neural processing principles [18].

D. Coding Strategies

Studies have shown the presence of a variety of coding techniques [19] such as: (1) Firing rate coding where the frequency of spiking is used to encode the information, (2) Temporal coding, where the exact timing of the spike encodes the information, (3) Phase coding, where the phase relationship between the spikes and the background oscillation from large groups of neurons encodes the information. These coding techniques appear to be constrained by energy, signal-to-noise ratio, processing speed, and other factors [9]. Whereas currently most VLSI architectures are based on centralized clocking with nano-second precise signaling, brain circuits utilize asynchronous signaling by a sparse population of neurons to encode the information [19]. Another interesting theory called *Polychronization* [20] has been proposed that utilizes the axonal connection delays within the brain networks to increase the memory capacity to store and retrieve patterns.

E. Self-organization

A very interesting property of the brain is its ability to organize itself based upon the statistics of the environment. This self-organization has two phases. One period happens during the early stages of life and is called the developmental stage, where the coarse grain wiring is defined by various genetic growth and environmental factors [21]. The second period, which happens during and after the development phase, fine-tunes the network according to the sensory information and experiential learning [22]-[24]. Exactly how these two periods contribute to brain-circuit development is an active area of research within the neuroscience community [21]- [24].

III. MODELING ABSTRACTION

Because of the brain's sheer complexity, designers of brain models need to choose an appropriate level of analysis, termed "modeling abstraction" (Figure 1) for their study. Each level of abstraction serves a different purpose and incorporates a subset of features of the brain. As we move higher up the abstraction ladder the models simplify the complex biological processes as to more computationally efficient approximations; thus the higher up, the more removed from the biology substrate. Higher abstractions have strong explanatory power and tend to be easier to apply to practical applications. However, lower-level abstractions incorporate key biological features that may not be included in a higher-order model. We extend the modeling abstractions introduced in the previous work [3] with more types and provide suitable examples for each.

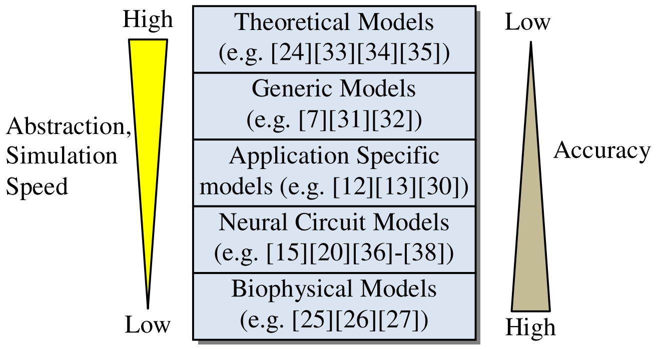
High
Abstraction, Simulation Speed
Low
Theoretical Models (e.g. [24][33][34][35])
Generic Models (e.g. [7][31][32])
Application Specific models (e.g. [12][13][30])
Neural Circuit Models (e.g. [15][20][36]-[38])
Biophysical Models (e.g. [25][26][27])
Low
Accuracy
High

Figure 2: Levels of Modeling Abstraction.

A. Cellular and Biophysical Models

Cellular level models incorporate molecular, electrical and morphological properties of neurons, detailed compartmental models of axons and dendrites from anatomical observations, and various kinds of ion channels to simulate the synaptic connections (e.g., GENESIS [25], NEURON [26] tools). A major goal of these models is to study detailed ionic channels and their influence on neuronal firing behavior [27]. While these models are biologically accurate, they incur tremendous computational costs for simulation. Further, these models require a large parameter space with some parameters insufficiently constrained by biological experiments, thus necessitating costly parameter sweeps. Hence large-scale simulation of the brain is extremely challenging at this level.

B. Neural Circuit Models

Neural circuit models abstract away many molecular and cellular details and consider the brain as a massive circuit composed of four basic components: neurons for computation, synapses for learning and storage, axons for communication, and neuromodulatory systems to control the attention and learning [8][28][22]. This approach is very similar to circuit theory used in analog and digital design, where circuits are studied using basic electronic components. Models of this type (and to some extent cellular level models) are able to exhibit network dynamics such as a persistent firing, random asynchronous firing, synchronous firing, and chaotic activity [29]. Section IV presents details of spiking neural networks as an exemplar for these neural circuit models.

C. Application Specific Models

Application Specific Models further simplify details of the brain-circuit (e.g., using signal processing transformations or algorithmic approximations), for use in specific applications such as object recognition, audition, navigation, etc. These models do not explain all the principles or behaviors of a specific brain region, but are developed from the understanding of some aspects of brain circuit properties. Several application specific models [12][13] have been developed to explain the human visual hierarchy [4] and consist of layers of simple and complex cells having progressively larger receptive fields. For example, the simple cell responses can be calculated by a convolution operation and the complex cell response is calculated by finding the maximum response over a pool of simple cells [13]. Another application specific model recognizes line-based objects [30] and is based on the idea that simple and complex cells in the visual cortex, V1, respond to oriented lines [4]. Thus a hierarchy of line-based feature detectors was developed. Each feature detector responds to a spatial combination of line segments, with the first layer responding to line-triplets and higher layers responding to more complicated feature combinations [30].

D. Generic Models

Generic models use algorithms that are not restricted to one application, sensory domain or dataset; they exploit studies showing that brain-circuits have a template architecture, where similar circuits are used for processing and learning various sensory signals [7][31][32]. These generic models can also incorporate top-down feedback to incorporate predictive signaling. Unlike neural circuit level models, most generic models do not incorporate detailed neural mechanisms such as the notion of spikes, synchronization, or oscillations. Rather, these models consist of algorithmic steps such as clustering, temporal pooling, associative memory etc [7][31][32].

E. Theoretical Models

Theoretical models are essentially developed to understand the statistics of sensory data and to explain high-level mathematical principles that can elucidate the function of the cortex [24][33]. Theoretical models can also address cognitive aspects such as consciousness (Global Workspace [34]) and brain function (Neural Darwinism [35]).

Ideally a model that spans several levels of abstraction could provide a good tradeoff between accuracy and simulation complexity. In the next sections we discuss the design and framework for spiking-based computation, since spike-based neural circuit models exhibit an interesting hybrid-computation paradigm that can be realized in VLSI circuits.

IV. Spike-Based Computation

Spike-based neural networks (SNN) model, to varying degrees, properties of the neuron and synaptic state, but most importantly they use an explicit representation of spike timing [6][28]. A spike-based model is inherently event driven, where an incoming spike changes the state of the post-synaptic (downstream) neuron.

A model of a simple local spiking circuit is illustrated in Figure 3(a) and the dynamics of a typical neuron is shown in Figure 3(b). In Figure 3(a), the excitatory neuron group (Group B) receives input spikes from the spike generator group (Group A). Furthermore each excitatory neuron is connected to an inhibitory neuron (Group C), which in turn inhibits the neurons in the local neighborhood of Group B. This circuit performs a form of winner-take-all (WTA) computation. The inhibition strength determines if the network is a hard-WTA network (only the winning neuron in the group fires) or soft-WTA network (a small group of strongly excited neurons fires). In Figure 3(a), the inhibitory group also stabilizes the excitatory group activity. A highly active excitatory group causes more inhibition and hence self-regulates its overall activity. Figure 3(c) shows the synaptic input connections to the center neuron, and receiving inputs from a large number of other neurons. When a sufficient number of input spikes (i.e., pre-synaptic) arrive “together” (i.e., temporally close), the post-synaptic neuron generates a spike of its own. The neurons are continually learning and modifying the strength of their connections based on their inputs [36]. The synaptic strength (weight) of a connection is increased if input (i.e., pre-synaptic) spikes consistently come before the output (i.e., post-synaptic) spikes; in other words, the temporal order of spikes is consistent with the input causing the neuron to spike. If however, the order is anti-causal (i.e., input spike occurs after output), the synaptic weight is decreased. This learning rule is called STDP (spike-timing dependent plasticity) [22] and has many interesting properties that enable a network to learn from spatial-temporal spike patterns [36]-[38].

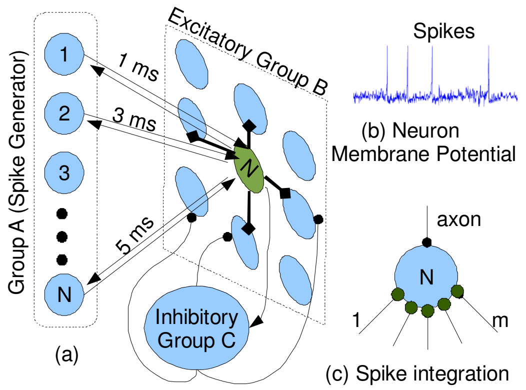

Figure 3: (a) Simple spiking neural network model. Axonal delay shown as X ms. (b) Membrane potential of the center neuron (c) synaptic connections for the center neuron.

Other neural circuits have been developed for unsupervised learning [37], and working memory [6]. More interesting dynamics and memory storage can be achieved if axonal delays are incorporated into the circuits [20]; interestingly -- unlike in VLSI circuits -- neural circuit delays can be viewed as a benefit, rather than a detriment to be minimized. It is also possible to create a generic recurrent microcircuit of spiking neurons that acts as a dynamical system that converts recent input data into a high-dimensional state [15]. This high-dimensional state can be easily classified or processed. Interesting applications are emerging using such a “reservoir” SNN [39].

V. HW ARCHITECTURES FOR SPIKE-BASED COMPUTATION

Large-scale SNN simulators are essential for simulating different aspects of human vision and other applications. For example, to model the behavior of the primary visual cortex, numerous neurons are required to process input images at different retinal (i.e., pixel) locations, spatial orientations, spatial frequencies, color, velocity, binocular disparity etc. These simulations can easily surpass a million neural elements even for low resolution input images, and while offering large amounts of parallelism, pose tremendous challenges for achieving fast simulation. Many dedicated hardware architectures and multiprocessor approaches have been used to increase the simulation speed of large-scale SNNs. Earlier works utilized supercomputers such as the IBM Blue Gene [27] or large-scale distributed clusters [40]. Techniques for distributed simulation of SNN are helpful for any kind of HW platform because it is difficult to fit the large-scale SNNs within one HW device. However, the cost and development time required pose fundamental limitations for these distributed approaches.

The advent of high-performance graphics architectures (e.g., NVIDIA GPUs) provides an interesting platform for large-scale SNN simulations [41]. Some fundamental benefits and limitations of such graphics architectures for simulating SNNs are:

1. Large fine-grained parallelism: Contemporary GPUs with hundreds of scalar processors can execute thousands of threads concurrently. Maximum performance can be achieved as long as a group of threads are executing the same instruction. However, when different threads within the same group require different instructions, *thread divergence* occurs, causing poor parallelism performance.
2. Large off-chip memory bandwidth: A typical GPU’s off-chip memory bus is based on a 512- (or 256-) bit wide DDR interface, resulting in a 5-fold increase in GPU memory bandwidth over the CPU. Through *memory coalescing*, this GPU memory bandwidth is exploited by clustering memory access patterns from different threads within a 128, 64, or 32-byte memory address space.
3. Special Function Units: Each streaming multiprocessor may have multiple special functional units (SFUs) allowing single instruction calculation of exponentials and other mathematical functions.

The mapping of SNNs on to GPUs is non-trivial due to the random memory access structure in SNNs and large memory requirements to store the connectivity and network state (synapses and neurons) information. We proposed various optimization techniques to overcome the above limitations and effectively map SNNs on to GPUs [41], including: (1) Exploiting both neuronal and synaptic parallelism to maximize thread level parallelism, (2) Efficient representation of large-scale SNNs that improves the off-chip memory coalescing, and (3) Minimizing thread divergence by delaying the execution of diverging conditions by buffering them and running them concurrently later. Using these optimization techniques, a SNN simulation with 100,000 neurons and 10 million synapses can be executed close to real-time [41]. Further, the GPU-SNN simulation was about 26 times faster than the CPU for 100K neurons with 50 million synapses.

Large-scale SNN simulations are memory dominated and hence overall speed-up is limited due to saturation of off-chip memory-bandwidth. Off-chip memory requirements can be reduced by providing large, persistent local shared memory to store the neuron state, and also by providing on-chip communication networks for direct spike transfer between multiprocessors. Such improvements have been incorporated into an application specific multiprocessor in the Spinnaker project [7] for large-scale SNN simulations. But to truly approach the power and area efficiency of brain circuits it is essential to directly model neuronal circuits using hybrid analog-digital architectures [5][18]. Wafer-scale integration [42] and memristor architectures [43] have been proposed to implement neuronal circuits with high-efficiency. These and other approaches are currently being investigated in the EU FACETS [44] and the DARPA SyNAPSE [45] projects.

VI. FRAMEWORK FOR SPIKE-BASED MODELING ON HIGH-PERFORMANCE ARCHITECTURES

Currently, SNNs have been used mostly to model various parts of the human brain, such as the visual cortex, hippocampus, or prefrontal cortex. But spike-based computations also hold promise as a generic technique that can be applied to processing other kinds of spatial-temporal information [15][38]. A CAD-based software framework would enable the modeling and development of SNNs for a wide range of practical applications, and their simulation on a wide range of hardware platforms.

Figure 4 shows a framework for SNN development and depicts four stages of the spike-based modeling framework: (A) Network Specification, (B) Network Optimization and Mapping, (C) SNN Simulation, and (D) Parameter tuning. We have released a downloadable open-source version of the GPU spiking simulator and related tools used for our research [49].

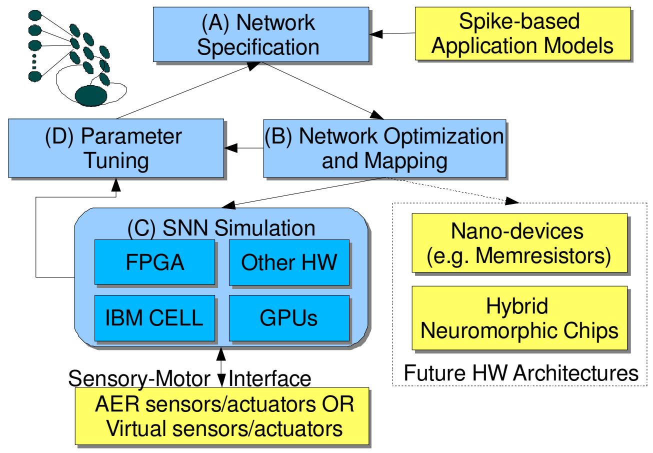
(A) Network Specification
Spike-based Application Models
(D) Parameter Tuning
(B) Network Optimization and Mapping
(C) SNN Simulation
FPGA
Other HW
IBM CELL
GPUs
Nano-devices (e.g. Memresistors)
Hybrid Neuromorphic Chips
Future HW Architectures
Sensory-Motor Interface
AER sensors/actuators OR Virtual sensors/actuators

Figure 4: Framework for Spike-based Neural Network Development

A. *Network Specification (Figure 4:A)*

Since most implementations of SNNs for vision and other applications require large populations of neural elements (~100K to 1 Million), it is infeasible to explicitly describe every neuron type and connections. Instead, the following high-level constructs can be used to describe SNNs [46]:

- Neuron population or Grouping: Denotes a population of neurons that share a specific topology and cell type. All neurons in one group have similar spiking dynamics and parameters (e.g., Group A and Group B in Figure 3a).
- Connection projection: Specifies the connection topology between two groups. Commonly used connection topologies are one-to-one, one-to-all, random, center-surround and Gaussian. More flexible user-defined topologies can also be supported.
- Synaptic properties: Specifies if the synaptic strength is fixed or plastic, the type of synaptic plasticity [22] (STDP), and other connection specific properties.
- Spike Generators: Special groups that accept data from external sources and generate Poisson spikes. Poisson spike generators allow for easy conversion of a real-valued signal to a spike train.
- Output effectors: Groups that convert spikes to real-values that are typically used for motor control and decision-making.

B. *Network optimization and mapping(Figure 4: B)*

In the network mapping stage, the SNN is optimized and mapped onto different simulators including HW based SNN simulators. Each optimization is specific to the underlying architecture on which the SNN will be simulated. For example, in GPU based simulators the out-going connections can be sorted based on axonal delays to improve the parallelism and memory bandwidth [41]. Similarly other network optimizations are essential for remapping the SNNs to run on distributed GPU clusters [40] or on future nano-architectures.

C. *SNN simulation (Figure 4: C)*

SNNs can be simulated using a discrete clock approach, an event-driven approach, or a mixture of the two [28]. In the pure clock-driven approach, the entire state of the SNN is updated every time step (1ms or smaller time-steps). The clock-driven approach generates spikes only at discrete time steps and hence can lead to some inaccuracies in spiking behavior and can change the spiking dynamics of simulated network [28]. In contrast, event-driven spiking simulators can generate spikes at any arbitrary time resolution, not just at specific timing steps. Most hardware simulators (e.g., Verilog/VHDL) work in event-driven mode. The main advantage of pure event-driven spiking simulators is better simulation performance and accuracy in simulating spiking dynamics. Simulation performance is high because state updates need to be computed only when a neuron spikes. The accuracy is higher than clock-driven because event-driven simulation calculates exact timings of spikes (avoiding temporal binning) [28]. Our existing GPU based spiking simulator [41] adopts a mixed-approach. The neurons are updated every time-step, and synaptic connections are updated in an event-driven fashion thus reducing the overall computation required. Many other HW platforms are being investigated for large-scale SNN simulations [8] [40]-[42].

D. *Parameter Tuning (Figure 4: D)*

In many applications, the selection of suitable parameter values for SNN simulations is extremely challenging because of the large number of parameters in a given model and complicated SNN dynamics. Sample SNN parameters that require tuning include the neuron dynamics, synaptic time constants, and synaptic strengths [47]. Most SNN models are hand tuned to avoid unstable states or locked synchronous oscillatory states (where *all* neurons fire synchronously). For some experiments the simulation model developer has a good understanding of the expected responses [38] and optimization techniques can be exploited [48]. GPUs and other parallel architectures are well suited for tuning and screening multiple networks concurrently to find a good set of candidate models. Based on the response from the simulator and SNN model, the parameter tuning stage (Figure 4:D) can adapt the model parameters and/or network connectivity to maximize some fitness function. In general the fitness function is highly non-linear and hence stochastic optimization techniques, such as simulated annealing or evolutionary strategies can be adopted to search the parameter landscape [48].

VII. DISCUSSION AND CONCLUSTION

Even though rapid progress is being made towards understanding details of brain circuits, many challenging problems remain to be solved. Some of them include unsupervised learning techniques in deep-hierarchical SNNs, multi-modal sensory integration and learning, techniques for stable operation of large-scale SNNs, and the role of feedback connection within brain-circuits. In this brief overview we explored various abstractions for modeling brain circuits and also frameworks for simulation of large-scale SNNs on high-performance architectures. Our ongoing research addresses the development of large-scale SNN based cortical vision models for various applications, and researching the role of neuromodulation in attention and learning tasks.

ACKNOWLEDGEMENTS

This work was supported in part by HRL subcontract 801888-BS under the DARPA SyNAPSE project. We also thank Professors Alex Nicolau and Alex Veidenbaum for discussions on architectural platforms and parallelization.

REFERENCES

[1] NAE Grand challenges for engineering -www.engineeringchallenges.org
[2] L.W. Swanson, Brain Architecture: Understanding the Basic Plan, Oxford University Press, USA, 2002
[3] P. Churchland and T.J. Sejnowski, The Computational Brain, 1994
[4] D.H. Hubel, Eye, brain, and vision, Scientific American Library, 1995
[5] C. Mead, "Neuromorphic electronic systems", Proceedings of the IEEE, vol. 78, 1990, pp. 1629-1636
[6] W. Maass and C.M. Bishop, Pulsed Neural Networks, MIT Press, 2001
[7] J. Hawkins and S. Blakeslee, On Intelligence, Times Books, 2004
[8] S. Furber and S. Temple, "Neural systems engineering," Journal of The Royal Society Interface, vol. 4, Apr. 2007, pp. 193-206
[9] S.B. Laughlin and T.J. Sejnowski, "Communication in Neuronal Networks," Science, vol. 301, Sep. 2003, pp. 1870-1874
[10] E. Bullmore and O. Sporns, "Complex brain networks: graph theoretical analysis of structural and functional systems", Nat Rev Neurosci, vol. 10, Mar. 2009, pp. 186-198.
[11] C. Weber and K. Obermayer, "Emergence of Modularity within One Sheet of Neurons: A Model Comparison", Emergent Neural Computational Architectures Based on Neuroscience, 2001, pp. 53-67
[12] K. Fukushima, "Recent Advances in the Neocognitron", Neural Information Processing, 2008, pp. 1041-1050
[13] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust Object Recognition with Cortex-Like Mechanisms," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, 2007, pp. 411
[14] V.B. Mountcastle, "The columnar organization of the neocortex," Brain, vol. 120, 1997, p. 701
[15] W. Maass, T. Natschläger, and H. Markram, "Computational models for generic cortical microcircuits", Computational neuroscience: A comprehensive approach, 2003
[16] R. VanRullen and S.J. Thorpe, "Surfing a spike wave down the ventral stream," Vision Research, vol. 42, Oct. 2002, pp. 2593-2615
[17] J.J. Nassi and E.M. Callaway, "Parallel processing strategies of the primate visual system", Nat Rev Neurosci, May. 2009, pp. 360-372
[18] R. Sarpeshkar, Ultra Low Power Bioelectronics, Cambridge Press, 2010
[19] M. Recce, "Encoding information in neuronal activity", Pulsed neural networks, MIT Press, 1999, pp. 111-131
[20] E.M. Izhikevich, "Polychronization: Computation with Spikes," Neural Comput., vol. 18, 2006, pp. 245-282
[21] Willshaw, D J, "Self-organisation in the nervous system," In Cognitive Systems: Information Processing Meets Brain Science. Eds: R G M Morris, L Tarassenko, M Kenward. 2005, Elsevier, pp5-33
[22] L.F. Abbott and S.B. Nelson, "Synaptic plasticity: taming the beast," Nature neuroscience, vol. 3, 2000, pp. 1178-1183
[23] G.E. Hinton, "Learning to represent visual input," Philosophical Transactions of the Royal Society B: Biological Sciences, Jan. 2010
[24] L. Wiskott and T.J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," Neural computation, 2002, pp. 715-770
[25] J.M. Bower and D. Beeman, "The Book of GENESIS", Springer, 1998
[26] M.L. Hines and N.T. Carnevale, "The NEURON simulation environment," Neural computation, vol. 9, 1997, pp. 1179-1209
[27] H. Markram, "The Blue Brain Project," Nature Rev Neurosci, Feb. 2006
[28] R. Brette, et. al, "Simulation of networks of spiking neurons: a review of tools and strategies," Journal of computational neuroscience, vol. 23, 2007, pp. 349-398
[29] T.P. Vogels, K. Rajan, and L. Abbott, "Neural Network Dynamics" Annual Review of Neuroscience, vol. 28, 2005, pp. 357-376
[30] J. M. Nageswaran, A. Felch, A. Chandrasekhar, N. Dutt, R. Granger, A. Nicolau, and A. Veidenbaum, "Brain Derived Vision Algorithm on High Performance Architectures," IJPP, vol. 37, 2009, pp. 345-369
[31] R. Granger, "Essential circuits of cognition: The brain's basic operations, architecture, and representations," In: AI at 50: The future of Artificial Intelligence, vol. 2006
[32] G.A. Carpenter and S. Grossberg, "Adaptive resonance theory," The handbook of brain theory and neural networks, vol. 2, 2003, pp. 87-90
[33] R.P.N. Rao, B.A. Olshausen, and M.S. Lewicki, Probabilistic Models of the Brain: Perception and Neural Function, The MIT Press, 2002
[34] B.J. Baars, "Global workspace theory of consciousness: toward a cognitive neuroscience of human experience," Progress in Brain Research, vol. 150, 2005, pp. 45-53
[35] G.M. Edelman, "Neural Darwinism: Selection and reentrant signaling in higher brain function," Neuron, vol. 10, Feb. 1993, pp. 115-125
[36] S. Song, K.D. Miller, and L.F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," nature neuroscience, vol. 3, 2000, pp. 919-926
[37] Micah Richert, J.M. Nageswaran, N. Dutt, J.L.Krichmar, "Biologically plausible homeostasis and STDP: Stability and learning patterned inputs in spiking neural networks," Intl. conf. on Cognitive NeuroScience-2010
[38] A. Gupta and L.N. Long, "Hebbian learning with winner take all for spiking neural networks," Proceedings of the International Joint Conference on Neural Networks, 2009
[39] M. Lukosevicius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," Computer Science Review, 2009
[40] A. Morrison, C. Mehring, T. Geisel, A. Aertsen, and M. Diesmann, "Advancing the Boundaries of High-Connectivity Network Simulation with Distributed Computing," Neural Computation, 2005, pp. 1776-1801
[41] J.M. Nageswaran, N. Dutt, J.L. Krichmar, A. Nicolau, and A.V. Veidenbaum, "A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors," Neural Networks, vol. 22, 2009, pp. 791-800
[42] J. Fieres, J. Schemmel, and K. Meier, "Realizing biological spiking network models in a configurable wafer-scale hardware system," , IEEE Intl. Joint Conference on Neural Networks, 2008
[43] S.H. Jo, T. Chang, I. Ebong, B.B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale Memristor Device as Synapse in Neuromorphic Systems," Nano Letters, vol. 10, 2010, pp. 1297-1301
[44] EU FACETS Project - http://facets.kip.uni-heidelberg.de
[45] DARPA SyNAPSE - www.darpa.mil/dso/solicitations/baa08-28.html
[46] A.P. Davison, et. al, "PyNN: a common interface for neuronal network simulators," Frontiers in Neuroinformatics, vol. 2, 2008
[47] Scholarpedia Article: Neuronal parameter optimization, www.scholarpedia.org/article/Neuronal_parameter_optimization
[48] W. Van Geit, E. De Schutter, and P. Achard, "Automated neuron model optimization techniques," Biological Cybernetics, Nov. 2008, p241-251
[49] UCI GPU-SNN release, http://www.ics.uci.edu/~jmoorkan/project/