Proceedings of the 2007 IEEE Symposium on Computational
Intelligence in Bioinformatics and Computational Biology (CIBCB 2007)

Building Smart Machines by Utilizing Spiking Neural Networks
- Current Perspectives -

Heike Sichtig
Department of Bioengineering, Binghamton University
Binghamton, NY 13902
Email: hsichtig@binghamton.edu

***Abstract*— In this paper we survey the current state of the art in spiking neural networks research and outline our approach to building smart machines. A thorough understanding of the history, open questions, and limitations of these networks can help the research community to gain a better grip on this new technology and to bridge the missing gaps. It is necessary to look at various aspects of spiking neural networks, such as the different modeling approaches, encoding schemes, simulators and learning techniques in order to efficiently make use of these networks. One paramount characteristic of spiking neural networks is the precise timing of inputs and outputs. As a dynamic system itself, it naturally lends itself to solving problems in the continuous domain such as time series analysis. This will be the focal point of our efforts to develop a smart machine utilizing spiking neural networks.**

I. Introduction

Science progresses best when observations force us to alter our preconceptions -Vera Rubin

How to build smart machines? The field of Artificial Intelligence has been investigating for a long time the possibility of building a truly intelligent machine that would replicate human behavior. There have been countless attempts to make a machine ”see” or ”hear”, focusing on the intrinsic vocabulary of a specific field.

For example, researchers have been devising neural networks in combination with genetic algorithms to build intelligent automated customer representatives [1]. Calling the hotline to get a phone number from an automated service can be very frustrating. It seems very unlikely that these systems improve in the near future without radically changing our preconceptions about understanding and engineering human behavior as diverse capabilities. Very few people consider the brain to process signals the same way for touch, hearing or seeing, rather they are comfortable with the notion that we have build-in modules for every task [2]–[4].

In his book, on Intelligence, Jeff Hawkins, founder of Palm Computing, Handspring and the Redwood Neuroscience Institute, describes a new theory of the brain, the *memory-prediction framework* [5]. In his framework he desribes a new approach to understanding the intricate workings of the human brain through a common algorithm that is performed by all the cortical regions. This idea surfaced first in a paper published in 1978 by Vernon Mountcastle, who points out that the neocortex is remarkably uniform in appearance and structure, having the same layers, cell types and connections throughout. He postulates that the only reason that one region of the cortex looks slightly different from another is because of what it is connected to and not because its basic function is different [6]. Hence, all the regions might perform the same basic operations. This single idea unites many diverse and wondrous capabilities of the human mind.

For example, Hawkins describes the existence of a so-called special visual area that seems to be specifically tailored to representing letters and digits. But the age-old question is whether we are born with the language area or not. Written language is a far too recent invention for our genes to have evolved a specific mechanism; hence, the cortex is still dividing itself into task-specific functional areas long into childhood [5]. This would suggest that the brain regions develop specialized functions depending on the information input.

Naturally, our brain processes continuous data and current technologies such as Markov models and traditional neural networks claim to process dynamic data but truly they are designed for static and discrete environments. Rabiners tutorial on Hidden Markov Models emphasized that these were the first techniques that really attempted to capture and model the temporal structure of dynamic data [7]. Moreover, traditional neural networks are most successfully applied to real-valued data problems and give promising results when processing static data.

In contrast to the previous models, the emergence of spiking neural networks as dynamic systems themselves provides an excellent tool for problems in the continuous domain. For example, video surveillance would greatly benefit from applying such models because of its strict requirements of precise timing and continuous data processing.

II. CURRENT STATE OF THE ART

Spiking neural networks emerged within the past decade and were made popular during a two-day workshop in August 1997 entitled Pulsed Neural Networks, resulting in a book with the same title edited by Wolfgang Maas and Christopher M. Bishop [8]. This recent development makes it possible to represent a system with time varying data, e.g. time series,

1-4244-0710-9/07/$20.00 ©2007 IEEE

more accurately and does not discard important characteristics of temporal data. In fact, an event can only cause an effect in the future not the past. A truly intelligent computational tool will focus on the propagation of the biological network through time.

The essential feature of a spiking neural network is that it explicitly takes the timing of inputs into account. The precise spike timing, the input and output representation as series of spikes, using the Delta function or more complex shapes, and the ability to continuously process information as a dynamic system itself makes SNNs stand out from previous techniques, such as the Hidden Markov Models or traditional neural networks.

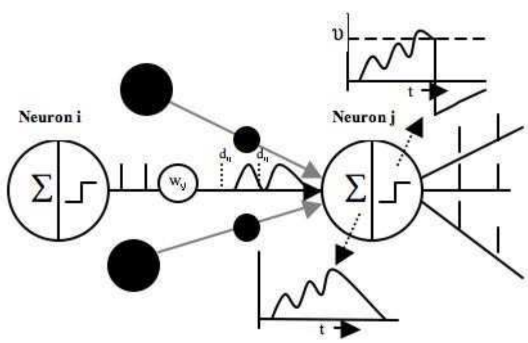

Fig. 1. Input/Output of Spiking Neuron j. Vertical bars represent firing time.

An interesting feature of spiking neural networks is the potential for having an unusually big memory capacity due to polychronization, a process of generating reproducible time-locked but not synchronous spiking patterns with millisecond precision. These patterns represent memory, and their number often exceeds the number of neurons, or even synapses, in the network.

In 2004, Olaf Booij emphasized in his Masters thesis that there is no practical way to teach the SNN model to process temporal patterns [1]. In fact, there is no specific formula of how to correctly model a spiking neural network to this date. There are several techniques for modeling spiking neural networks, the one we refer to here is called the General Spike Response model and was originally developed by Wulfram Gerstner [1], [8], [9]. Models of spiking neural networks can be broadly classified by their level of abstraction. The most abstract models do not describe the state of a neuron in terms of molecules but rather by a real number, called its membrane potential. The less detailed models make it easier to build a network and to figure out how to make them learn something.

The General Spike Response Model (SRM) and the Threshold and Fire Model are the most popular. Equation (1) is the mathematical formulation of the SRM model, where the total membrane potential of neuron j, u_j(t), is the sum of all the post-synaptic potentials (PSPs) caused by pre-synaptic firings of neuron i and the refractory effect of a negative reset potential. Once neuron j generates a spike at time t_j the membrane potential is reset by the spike after-potential, described by $\eta[t - t_j^{(f)}]$. Hence, the total membrane potential of neuron j is composed of the η-function modeling it's own refractoriness, the ϵ-function describing the effect of the pre-synaptic spikes of neuron i, the weights w_{ji} and the delays of the connections d_{ji}. Let F_i represent the spike train from pre-synaptic neuron i, F_j the spike train from the current neuron j and Γ_j the spikes of pre-synaptic neuron i affecting the potential of neuron j.

$$u_j[t] = \sum_{t_j^{(f)} \epsilon F_j} \eta[t-t_j^{(f)}] + \sum_{i \epsilon \Gamma_j} \sum_{t_i^{(g)} \epsilon F_i} w_{ji} \times \epsilon[t-t_i^{(g)} - d_{ji}] \quad (1)$$

The ϵ-function and η-function are not fixed in the Spike Response Model, but there are certain limitations outlined by Gerstner [1], [8], [9]. In fact, the ϵ-function must have a short rising part followed by a long decaying part, describing the effect of the pre-synaptic spike to the potential of the post-synaptic neuron. Moreover, the η-function requires that η(t) = 0 for t$\leq$0.

In 2003, Izhikevich introduced another simple model for spiking neurons that he derived by applying bifurcation methods to Hodgkin-Huxley-type neuronal models [10]. According to the author his model combines computational efficiency of integrate-and-fire and resonate-and-fire models and biological plausibility and versatility of Hodgkin-Huxley type models. There are four dimensionless parameters in the model that can be tweaked to produce desired spiking behaviors such as regular spiking, intrinsically bursting, chattering, low-threshold spiking or fast spiking. In 2005, Izhikevich used his simple model to finish simulating a large-scale model containing 10^{11} neurons and 10^{15} synapses [11].

In contrast to the more general models, the least abstract models try to simulate the neuron very accurately and take all the different biochemical processes into account, limiting the computations to few neurons due to its high complexity. Conductance-based models, like the Hodgkin-Huxley model and compartmental models fall into this category.

A. *Coding of Information*

There are many ways the input and output to a spiking neural network can be encoded. It depends on the type of task the SNN should solve. First, the time-to-first-spike coding includes one spike per numerical value, meaning that the firing time is proportional to the numerical value. If the problem contains more then one event, phase coding can be applied which encodes a stream of numerical values. In other words, the stream is sequentially encoded using one numerical value in one spike-time during a time period, then combining these periods one after the other. Both coding schemes have limited capabilities due to the restriction of only one spike at a time or period.

A popular method is thresholding where numerical values are reduced to a stream of bits, seen as a spike train. Although this coding scheme might result in large loss of information by reducing every value to a bit, the dimension reduction could be an advantage, resulting in compact spike trains. The reduction of neurons and spikes clearly stands out when having to deal with large data streams.

Lastly, data from one variable can also be scattered over more than one neuron called population coding. Here a numerical value gets transformed into a couple of spike times, each belonging to a different neuron, using receptive fields. This coding scheme is also commonly referred to as the receptive field method. Here every neuron overlaps the numerical value with a Gaussian kernel (Equation 2) having a specific mean and variance.

$$K_g[z_{ij}] = \frac{1}{\sqrt{2\pi}} \times \exp[\frac{-z_{ij}^2}{2}] \tag{2}$$

$$z_{ij} = \frac{1}{h}[v_j - x_i] \tag{3}$$

K_g is the Gaussian Kernel, z_{ij} is the distance of observation x_i from point v_j (Equation 3) and h is the bandwidth of the sliding window. The height of the Gaussian kernel of the numerical value determines the firing time of the neuron.

B. SNN Simulators

There is a large number of neural network simulators available on the internet but a need-tailored SNN simulator is very hard to find. This is in part due to the very recent development of spiking neural networks and its enormous number of modeling choices. We primarily focused on two simulators. One of the simulators is Amygdala 0.4 which has improved speed and capabilities for abstract modeling of spiking neural networks to represent a more plausible biological network, making it a valuable research tool for experimenting with large numbers of neurons. The other simulator is Genesis that is not only capable of modeling spiking neural networks but any kind of biological network by simulating neural systems ranging from sub cellular components and biochemical processes to rather complex single neurons and beyond.

Our reasoning for choosing these two simulators is that they can be used for cross-validation. In a mailing list posting in July 2005, Ruediger Koch, one of the developers of Amygdala, states that the software can be verified by comparing the same neuron model that is simulated with Genesis to Amygdala. Both simulators are available freely under the GNU General Public License. The Amygdala simulator is written in C++ and available for Linux and Windows (cygwin). Alternatively, the Genesis simulator is written in C and available for Linux, Windows (cygwin) and Mac OS/X.

We focus on using Amygdala as our simulator of choice because of its speed and simplicity. The creators of Amygdala claim that about 500,000 input spikes per second can be processed on an Athlon 2000+ when using the basic neuron with static synapses.

Amygdala uses exclusively the time-to-first-spike as the protocol between neurons and the network [12]. Furthermore, Newtons method (see Program Listing P1 below) is used for calculating the occurrence of a spike in a neuron.

P1: Next Spike Calculation (basicneuron.h)

1) *Determine the membrane potential at the current state (calculation time input spike time) by summing the state of each input spike up to the calculation time.*
2) *Find derivative of the function for the calculation time.*
3) *Calculate intercept with the threshold.*
4) *Set new calculation time to time of intercept.*
5) *Repeat until:*
   a) *Two successive iterations result in no change in calcTime. (Does Converge)*
   b) *The derivative of the function becomes negative. Go to the next current state. (Does not Converge)*

The simulator is based on integrate and fire model neurons, where the user is able to create Basic and Alpha neuron models with static or dynamic synapses. The Basic neuron is a simple integrate-and-fire neuron model based on the Gerstner model [8]. Alternatively, the Alpha neuron is based on the area W neurons described in the Hopfield-Brody Mus Silicium papers [13], [14]. Furthermore, the user can create models using static or dynamic synapses. The dynamic synapses are modeled based on the description by Maas and Markram [15], [16].

However, there are several limitations with Amygdala. First, there are no different coding schemes other than the time-to-first-spike. This could be a rather big problem when large amounts of information need to be processed. Another limitation is that neurons have to wait one simulation time step before they fire. Furthermore, Amygdala uses an approximation for calculating the post-synaptic potential by using only the synaptic time constant.

Other simulators include Spike, SpikeSNNS and SimSPINN [17]–[19].

C. Applications

Spiking neural networks can be applied to any problems traditional sigmoidal neural networks solve, especially where time-varying data is involved. There are four general goals within the traditional neural network domain for application areas [20].

- Auto-association, where the network is fed with a training-set of patterns, which it is supposed to store by tuning its free parameters. When presented with a new pattern it should reproduce the most similar training-pattern.
- Pattern-association, where the network is fed with a training-set of pairs of patterns and the network should learn the generalized mapping between the input- and output-patterns. When presented with a new input-pattern it should produce an output-pattern that is consistent with this mapping.
- Classification, which can be seen as a sub-task of the pattern-association; the required output-pattern consists of the predefined category the input-pattern belongs to.
- Clustering, in which training-patterns are not known a priori, but the network should discover certain salient features with which it is able to divide the data in different classes.

Not only can spiking neural networks accomplish these tasks but is also capable of dealing with time dependent data.

In a recent review by Bohte, Joost and Kok they propose an application of Spiking neural networks to liquid state machines (LSMs) [21]. This application is based on the work of Markram et al. [22] that a randomly connected network of spiking neurons effectively implements a complex temporal filter through the intricacies of reverberating activity and synaptic dynamics. Given a temporally extended input, like speech, the collective activity of the network can be described as a trajectory through a high-dimensional state space, and this trajectory should be identifiably specific for the input at hand. A simple read-out decoder should then be sufficient to classify the temporal pattern. Hence, spiking neural networks can be a very useful tool for classification tasks on temporal data, such as speech recognition and time-series prediction.

Moreover, Booij reported in his Masters thesis that spiking neural networks have been successfully applied to classification problems involving dynamic data, where the system predicts a value of a certain dynamic set of variables given its history [1], [23]. Indeed, a research group has recently developed an unsupervised learning rule that clusters temporal patterns by learning to discriminate two different audio samples [1], [24].

Other areas where spiking neural networks could make real progress due to its dynamic nature are face recognition, vision, and robot control. Additional areas of promise for the application of spiking neural networks are scene segmentation in film and the detection of criminal behavior with surveillance cameras.

D. Community

Despite the recent development of spiking neural networks there are many research groups already utilizing them in their research. Yet we found it difficult to find a centralized US-based portal dedicated to spiking neural networks.

One research group is at the University of Pennsylvania, lead by Dezhe Z. Jin, who was a student of Sebastian Seung from MITs Department of Brain and Cognitive Sciences. According to their mission statement, their major goal is to apply theoretical analysis of the biophysical properties of neural networks. Currently they are working on research in the following four areas: motor control learning in basal ganglia, song generation and recognition in songbirds, olfaction in mammals and insects and finally, orientation selectivity and feature maps in the primary visual cortex [25].

Olaf Booij from Amsterdam University has written his Masters thesis on spiking neural networks doing temporal pattern classification in lip reading, where his spiking neural network learns to recognize spoken words out of video fragments with stunning results. Moreover, he gives an excellent introduction to spiking neural networks making it a useful reference and starting point for researchers that are interested in using spiking neural networks in their own research. [1].

In addition, the Goodman Brain Computation Lab at the University of Nevada has done intriguing research using spiking neural networks by attempting to create realistic brain models with enough innate knowledge to pass a simple fitness test using their in-house SNN simulator, called NCS-NeoCortical Simulator and a python program, called Brainlab, applying a genetic algorithm for parameter tuning [26].

There are many other research groups and individuals that are involved with spiking neural networks that would benefit from a centralized resource. There is a high probability that the spiking neural network community will grow larger within the next years and develop not only efficient but useful applications that will most likely change the way we see the world today. For example, a more sophisticated video surveillance application could possibly be capable of stoping criminals in real time in the future. This does not have to be as futuristic as it sounds but can be reality by employing collaboration between researchers through communication.

E. Limitations

Despite the many applications of spiking neural networks, one must be aware of our rather limited knowledge of the intricate workings of biological neurons and how they process and encode information. Additionally, there is limited empirical data and computational theory about computing time series in biological and artificial pulsed neural nets.

III. CURRENT RESEARCH BUILDING A SMART MACHINE

Our interest in spiking neural networks revolves mainly around constructing an intelligent machine that will act as a classifier for time series data. The desired characteristics of such a machine are precise timing, hierarchical information flow in both directions and plasticity emerging its own machinistic intelligence.

There are many open questions within the field of spiking neural networks due to its recent development that need to be addressed. First, it is rather unclear how one can efficiently incorporate dynamic learning into spiking neural networks. Then there are many modeling techniques that are proposed but there is no sufficient computational theory or efficient topology linked to it. Hence, our research focuses on three specific areas:

- Pick a suitable neuron model and simulator.
- Utilize genetic algorithms to tune free parameters of the chosen model.
- Utilize genetic algorithms to further optimize network topology.

Currently most available SNN simulators are based on specific models and research problems or are defined too broadly. Adapting such a simulator for our research appeared to be very time consuming, because of the intensive amounts of source code that needed to be understood and rewritten. After reviewing simulators such as Genesis, Amygdala, SimSPINN, SpikeSNNS and Spike, we opted for using Amygdala as the simulator of choice because it has all the computational properties needed for our model, including the Gerstner Model with Markram synapses and Hebbian learning [8], [9], [15], [16],

[22], [27]. This technology is well documented in books and papers and the computational properties look very promising. Eventually we might write our own next-event SNN simulator. Initially we will keep our models small and the topologies fixed to gain more insights into the intricate workings of the rather complex dynamics of spiking neurons.

Furthermore, we will utilize genetic algorithms for fine tuning desired model parameters attempting to replicate the behavior of Lloyd Watt's tonic buster example from 1994 [28] and the more complex Jin model from 2004 [29]. The Watts model has 2 neurons that are connected and tuned in such a way that it produces adapting bursting behavior. In contrast, the Jin model consists of a synfire chain of four excitatory neurons and two globally inhibitory interneurons of different types providing fast feedback and delayed feed-forward inhibition. The Jin network recognizes a specific spatiotemporal spike sequence when the last excitatory neuron of the synfire chain spikes, but only if the sequence was present in the input spike stream. Initially, we will use a simple fitness function for parameter tuning of the network by taking the difference between each output pulse train and the target. After we achieve satisfactory parameter optimization, genetic algorithms will be applied to investigate the possibility for optimizing the network topology itself.

Currently, we are working on validating and generalizing input methods for Amygdala. At the same time we are in the process of writing our own simulator. This way we can have complete control over the properties of the model, synapses and learning techniques.

IV. FUTURE WORK

Once the simulator is giving satisfactory results for implementing spiking neural networks using the Gerstner Model with Markram synapses and Hebbian learning, we will write a more sophisticated Genetic Algorithm for SNN Parameter Optimization. The goal of the refined Genetic Algorithm is to learn the appropriate output to a set of dynamic inputs and apply it to time series analysis. Time permitting, we will look at other network topologies to analyze their efficiency and structure.

ACKNOWLEDGMENT

I thank J. David Schaffer of Philips Research and Craig B. Laramee of Binghamton University for their assistance in helping me to formulate my approach and their comments on drafts of this paper.

REFERENCES

[1] O. Booij, "Temporal pattern classification using spiking neural networks," Master's thesis, Intelligent Systems Laboratory (ISLA), University of Amsterdam, August 2004, available from http://www.science.uva.nl/˜obooij/publications/Booij04SNNThesis.

[2] J. A. Fodor, *Modularity of Mind: An Essay on Faculty Psychology*. Cambridge, MA, USA: MIT Press, December 1983, vol. 60, no. 4, ch. Language, pp. 976–979.

[3] S. Pinker, *How the Mind Works*. Norton, 1997.

[4] W. R. Uttal, *The New Phrenology: The Limits of Localizing Cognitive Processes in the Brain*. Cambridge, MA, USA: MIT Press, 2001.

[5] J. Hawkins and S. Blakeslee, *On Intelligence*. Times Books, 2004.

[6] V. Mountcastle, *An organizing principle for cerebral function: The unit module and the distributed system*. Cambridge, MA, USA: MIT Press, 1978, ch. The Mindful Brain, pp. pp. 17–49, in G.M. Edelman and V. B. Mountcastle.

[7] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Readings in Speech Recognition.*, pp. 267–296, 1990.

[8] W. Maas and C. M. Bishop, *Pulsed Neural Networks*. Cambridge, MA, USA: MIT Press, March 2001.

[9] W. K. W. Gerstner, *Spiking Neural Networks*. Cambridge University Press, August 2002.

[10] E. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on Neural Networks*, vol. 14, pp. 1569–1572, 2003.

[11] ——, "Simulation of large-scale brain models," Online, 2007, last Accessed on January 14, 2007.

[12] M. Groover and R. Koch, "Amygdala-a spiking neural network library," SourceForge," User Manual, April 2002, version 0.2.

[13] J. Hopfield and C. Brody, "What is a moment? cortical sensory integration over a brief interval," *PNAS*, vol. 97, no. 25, pp. 13 919–13 924, 2000.

[14] ——, "What is a moment? transient synchrony as a collective mechanism for spatiotemporal integration," *PNAS*, vol. 98, no. 3, pp. 1282–1287, 2001.

[15] W. Maas and H. Markram, "Synapses as dynamic memory buffers," *Neural Networks*, vol. 15, pp. 155–161, 2002.

[16] H. Markram, Y. Wang, and M. Tsodyks, "Differential signaling via the same axon of neocortical pyramidal neurons," *Neurobiology*, vol. 95, pp. 5323–5328, April 1998.

[17] L. Watts, "Spike and neuralog," Online, 2007, last Accessed on January 14, 2007.

[18] I. Marian and R. Reilly, "Spikesnns - a simulator for spike processing networks," in *In Proceedings of 12th DAAAM International Symposium on Intelligent Manufacturing und Automation: Focus on Precision Engineering*, B. Katalinic, Ed. DAAAM International, October 2001.

[19] M. Walker, H. Wang, S. Kartamihardjo, and U. Roth, "Simspinn – a simulator for spike-processing neural networks," in *In Proceedings of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics*, A. Sydow, Ed. Jena, Germany: Wissenschaft und Technik Verlag, October 1997.

[20] C. P. R. Group, *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA, USA: MIT Press, 1986.

[21] S. M. Bohte and J. N. Kok, "Applications of spiking neural networks," *Information Processing Letters*, vol. 95, no. 6, pp. 519–520, September 2005.

[22] W. Maass, T. Natschlger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[23] J. Sohn, B. Zhang, and B. Kaang, "Temporal pattern recognition using a spiking neural network with delays," in *In Proceedings of International Joint Conference on Neural Networks*, vol. 4. IJCNN99, 1999, pp. 4.2590–2593.

[24] J. Storck, F. Jaekl, and G. Deco, "Temporal clustering with spiking neurons and dynamic synapses: Towards technological applications," *Neural Networks*, vol. 14, no. 3, pp. 275–285, 2001.

[25] D. Jin, "Publications," Online, 2007, last Accessed on January 14, 2007.

[26] R. Drewes, "Modeling the brain with ncs and brainlab," *Linux Journal*, June 2005.

[27] W. Gerstner and W. Kistler, "Mathematical formulations of hebbian learning," *Biological Cybernetics: Hebb in Perspective*, 2002.

[28] L. Watts, "Event-driven simulation of networks of spiking neurons," *Advances in Neural Information Processing Systems*, vol. 6, p. 927934, 1994.

[29] D. Z. Jin, "Spiking neural network for recognizing spatiotemporal sequences of spikes," *Physical Review E*, vol. 69, 2004.