Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 61 (2015) 314 – 321

Procedia Computer Science

Complex Adaptive Systems, Publication 5
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2015-San Jose, CA

Adaptation of Spike-Timing-Dependent Plasticity to Unsupervised Learning for Polychronous Wavefront Computing

Fred Highland[a]*, Corey B. Hart[b]

[a]*Lockheed Martin IS&GS, 9321 Corporate Blvd, Rockville, MD, 20850, USA*
[b]*Lockheed Martin IS&GS, 230 Mall Blvd, King of Prussia, PA, 19406, USA*

Abstract

Non-Von Neumann computational architectures have lately aroused significant interest as substrates for complex computation. One recent development in this domain is the Polychronous Wavefront Computing (PWC) computational model based on multiple wavefront dynamics. This model is an abstraction and simplification of the artificial neural network paradigm based on temporal and spatial patterns of activity in a pulse propagating media and their interaction with transponders. While this framework is capable of computing basic logical functions and exhibiting interesting dynamic behaviors, methods for unsupervised training of the framework have not been identified. The lack of input weights and the spatio-temporal nature of the PWC framework make direct application of weight adjusting learning methods (e.g., backpropagation) impractical. The paper will describe research into unsupervised learning for PWCs inspired by Spike-Timing-Dependent Plasticity (STDP) methods used with other types of polychronous models. The method is based on adding Leaky Integrate-and-Fire semantics to the PWC framework allowing analysis of activating wavefronts and determination of the optimal location for future stimulation. The transponder's location is then incrementally adjusted to improve its future response. The paper will discuss the learning approach and examine the results of applying the method over a series of stimulations to sample configurations.

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology

Keywords: neural networks; non-von Neumann computation; polychronous wavefront computing; spike-timing-dependent plasticity; unsupervised learning

* Corresponding author. Tel.: +0-301-471-4685
E-mail address: fred.highland@lmco.com

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology
doi:10.1016/j.procs.2015.09.146

1. Introduction

Non-Von Neumann computational architectures have lately aroused significant interest as substrates for complex computation[1,2]. These architectures offer the potential to develop low-power electronic neuromorphic computers that can scale to biological levels. One recent development in this domain is definition of Polychronous Wavefront Computing (PWC)[3] a computational model based on multiple wavefront dynamics. This model is an abstraction of the spiking neural network paradigm based on temporal and spatial patterns of activity in a pulse propagating media and their interaction with transponders. It simplifies the physical implementation of complex networks by eliminating direct connections between nodes. While this framework is capable of computing basic logical functions and exhibiting interesting dynamic behaviors, methods for programming or training the framework have not been identified and their definition remains a manual, or at best, a heuristic[4], process. The lack of input weights and the spatio-temporal nature of the PWC framework make direct application weight adjusting learning methods (e.g., backpropagation[5]) impractical. The mathematics of the geometric and temporal relationships of these configurations have been defined[6] but manual construction of large scale configurations remains a difficult task.

The work presented here describes research which combines PWC concepts with Leaky Integrate-and-Fire spiking neuron models[7,8] and Spike-Timing-Dependent Plasticity (STDP) learning methods[9] to provide a means of programming PWC configurations. It will begin with a brief overview of basic PWC concepts to provide background. It will then discuss modifications to the PWC framework to covert it from a digital model to an analog one that will allow the application of STDP learning concepts. It will then present a learning algorithm based on STDP concepts adapted to PWC. An alternate view of this model with then be discussed that provides a basis for a possible physical implementation and shows its relationship to conventional weight-based neuron models. It will then discuss experiments with the model applying it to simple configurations to investigate its characteristics. It will conclude with a summary of results and directions for future work.

2. Background

Traditional semiconductor-based computation based on Von Neumann architectures has begun to run up against basic limitations of the medium. While nearly everyone is familiar with Moore's law, the continuation of this truism is being increasingly questioned[10]. Given these constraints, there have been ongoing efforts to develop alternatives to digital computing. One of the more interesting developments has been the adoption of architectures inspired by neural network structures. It is possible to show that mimicking the spiking neural approach of the brain yields computational capabilities greater than an N-gate digital logic gate based system by a factor of at least log N[11]. This has led to a proliferation of neuromorphic processors[12] that recapitulate this brain-like structure to a greater or lesser degree. The most realistic of these architectures take inspiration from what is known as the third generation of neural networks - spiking neurons connected with both realistic time-delays and learnable synaptic weights.

Polychronous Wavefront Computing (PWC) was proposed by Izhikevich and Hoppensteadt[3] as a generalization of spiking neural networks. It replaces explicit connections between neurons by wavefront pulses generated and received by transponders. When two or more wavefronts intersect a transponder a pulse is emitted by that transponder. PWC configurations have been show to exhibit interesting computational properties such as logic gates and recognition of time coded pulses[3]. They provide many of the properties of spiking neural networks in a simpler implementation. Their use of wavefront pulses for communication instead of explicit wired connections greatly simplifies the implementation neurocomputing systems.

For PWC to fulfill its promise of implementing practical neurocomputing systems, methods to define useful PWC configurations must be developed. Currently, programming PWC configurations by placing transponders at certain locations is still an art rather than a science[3]. Research has explored the mathematical properties and constraints of configuring PWC networks[6] but no research has been done on the application of automated configuration methods or learning techniques. Methods have been proposed that use numerical programming to incorporate the large number of constraints that will occur[4] but these have not been experimentally proven and the number of geometric constrains presented by large scale PWC configurations may make this approach computationally challenging. The ultimate objective of this work is to define methods of unsupervised learning localized to the PWC transponder that can recognize useful patterns of input data and scale to extreme levels of complexity.

3. Adaptation of STDP to PWC using a Leaky Integrate-and-Fire Neuron Model

STDP (Spike-Timing-Dependent Plasticity)[9] provides a method to update the synaptic weights of spiking neural network nodes (neurons) to improve their recognition of input patterns. It works by adjusting the weights associated with input nodes based on the timing of the pre-synaptic stimulus and the post-synaptic spike generated by the node. Weights for a stimulus preceding the post-synaptic spike are increased (potentiated) while weights for stimulus succeeding the post-synaptic spike are decreased (depressed). This method provides a learning mechanism that does not depend on external supervision of the learning process, is biologically plausible and has been shown to provide effective unsupervised learning in complex spiking neural networks.

The STDP method could provide a basis for learning in PWC configurations but it requires two properties that are not present in the PWC framework as originally proposed. First, it is based on modeling signals as continuous quantities. The proposed PWC framework defines stimulus as the intersection of multiple wavefronts with a transponder which is an integer quantity. Second, STDP is dependent on the modification of synaptic weights that determine the strength of the relationships between nodes. The PWC model does not define weights between transponders and is not aware of the relationships between specific transponders.

In order to make the definition of signals continuous within the PWC framework, a Leaky Integrate-and-Fire Model[7,8] is incorporated. The basic PWC framework relies on the simultaneous intersection of two (or more) wavefronts with a receiving transponder. The model used here generalizes that approach by modeling the sensing of each wavefront by the transponder independently and adjusting its strength based on its arrival time as shown in Figure 1. The arrival time is used as the trigger for a synapse in the Leaky Integrate-and-Fire Model which integrates multiple synaptic pulses together after applying an exponential decay adjustment. The node fires if the result exceeds a threshold value. If the two wavefronts arrive at the same time, the effect is the same as the base PWC framework. However, if the wavefront intersection does not exactly intersect with the transponder, a reduced wavefront stimulus is produced. Through adjustment of the coefficients of the model, activation of the transponder can be achieved without the wavefront intersection being precisely at the location of the transponder.

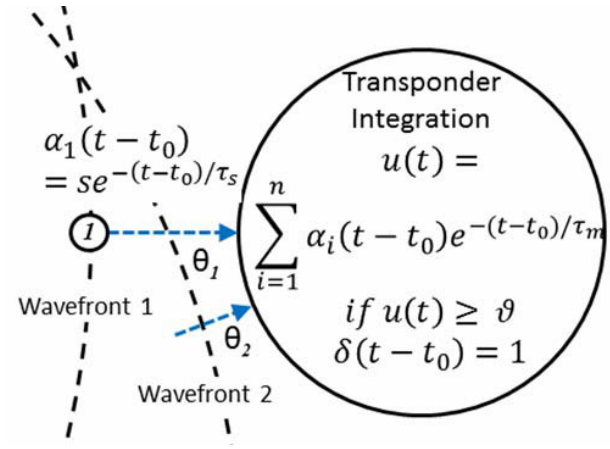

Figure 1 - Leaky Integrate-and-Fire Transponder Model.

As shown in Figure 1, sensor 1 detects Wavefront 1 (dashed arc) and sends a stimulus to the transponder integration function. The stimulus $\alpha_i(t-t_0)$ is exponentially time varying and decays based on the time since it was received (t_0) and τ_s, the synapse time constant. In the example, sensor 1 has just detected a wavefront, has a Δt of 0 and therefore produces a stimulus of 1. Wavefront 2 has already passed sensor 1 and produces a stimulus less than 1 based on exponential decay. Stimuli from multiple waves are integrated by the transponder by the summation function to produce the time dependent total stimulus *u(t)*. When the sum exceeds the membrane threshold (υ) a pulse ($\delta(t-t_0)$) is emitted by the transponder. The angle of the pulse waves, θ_1 and θ_2, is also detected for use in the learning algorithm below.

PWC transponders do not have weights on input signals nor, in the most general sense, do they even know where (which transponders) those signals came from. The activation of the transponder is based on the interaction of pulse waves from other transponders (usually two or more pulse waves) which is determined by the position of the transponders relative to each other and the timing of pulse wave arrival. To apply STDP-like mechanisms to PWC transponders, changes must be made to the transponder position and/or the activation threshold to improve the responsiveness under stimulation. This is addressed in the learning algorithm described below.

4. Learning Algorithm

The basic learning approach is as follows.

For each analysis cycle:

- If sum of pulse waves activate the transponder (i.e., the sum exceeds the membrane threshold (υ)), adjust the location of the transponder toward the pulse wave intersection by an amount inversely proportional to the estimated distance to the pulse wave intersection.
- If sum of pulse wave stimulus falls below the membrane threshold, adjust the transponder location away from the pulse wave intersection by an amount inversely proportional to the estimated distance to the wave intersection.

In order to represent a biologically plausible mechanism and provide an unsupervised learning method which does not rely on global knowledge about the configuration, the learning algorithm relies on only locally available information. The process is represented in Figure 2. The direction of the waves (θ_1, θ_2) are determined by sensing how the wave intersects the transponder at x_0,y_0. In practice, it can be computed from the location of the originating transponder but local sensors could accomplish the same effect. The distance of the waves (L_1, L_2) can be derived from the time after passing the transponder sensors location. The wave directions can be used to find the slope of the lines perpendicular to the direction of wave travel that intersect x_1,y_1 and x_2,y_2 which can then be used to estimate the wavefront intersection x_i,y_i. Note that x_i,y_i is an overestimate of the intersection because lines are used in the formulation while the actual location is the intersection of two circles. This is reasonable because the wavefronts are nearly linear at low values of L and the incremental nature of the learning approach can compensate for the estimate.

The position of the transponder is then adjusted toward or away from the intersection point following a modified version of the STDP method[9]. The adjustment incorporates a non-linear adjustment based on the timing of the pulse to the activation of the transponder (which is proportional to the distances L_i). The adjustment is based on:

$$+A_{+}e^{-(t-t_0)/\tau_{+}}, \text{if waves activtate transponder}$$

$$-A_{-}e^{-(t-t_0)/\tau_{-}}, \text{if waves do not activtate transponder}$$

Where:

A_+ - max potentiation modification (potentiation limit)
A_- - max depression modification (depression limit)
t - time of transponder activation
t_0 - time of pulse receipt at the transponder
τ_+ - potentiation time constant
τ_- - depression time constant

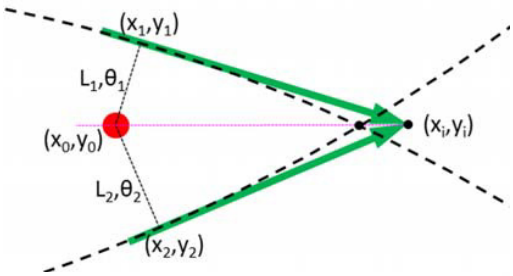

Figure 2 - Wavefront Intersection Location Estimation Approach.

Note that this differs from the STDP approach in that the sign of the depression times are not reversed because the time delta is always positive in this model. In addition, the modification values are used as a scaling factor for changes in position rather than an increment to the current position. This is necessary to prevent excessive transponder movement from overshooting the intersection estimate which would prevent convergence.

The method presented here adjusts the position of the transponder based on two wavefronts. To be useful in solving complex problems, transponders will need process multiple wavefronts. This can be addressed by extending the algorithm to compute the centroid of all pairs of stimulating wavefront intersections biased by their stimulus levels and using that point as the new movement target.

The approached defined above implements a basic analogue to STDP for PWC. As discussed below, the model is sensitive to various parameters that can make learning both too aggressive and requires wavefront intersections to be very close to the transponder. To address these issues, some extensions to the model are under investigation. To limit the effects of aggressive learning in complex configurations, the learning step size is limited to a feasible

region proportional to the standard deviation of previous positions. To allow learning to occur with wavefronts that are further from the transponder, "close" wavefronts that are slightly outside the firing threshold ($u(t) < \upsilon$) may be allowed to adjust the transponder location without generating and output pulse. The utility of these approaches are still being researched.

5. Physical Implementation

The method described above provides a means to reposition PWC transponders to improve their performance based on repeated stimulations. This approach is effective in simulations but problematic as a physical implementation because a dynamically movable transponder design will be significantly more complex to build.

However, the effect of moving physical transponders is to change the timing of the stimulus the transponder receives resulting in a more optimal firing pattern. If a time delay is introduced into the transponder model, the same effect can be implemented without the requirement for movement. This would mean that transponders would not have to be moved to learn as long as they were sufficiently close to an optimal point.

The transponder block diagram in Figure 3 shows a conceptual physical implementation of this approach. It consists of a central transponder "soma" circuit surrounded by a ring of "dendrite" sensors. The number of sensors does not have to be large, only enough to get a reasonable resolution of the signal direction (on the order of 8-12) and could be relatively close to the soma to minimize the footprint of each transponder. The dendrite sensors would sense a pulse and transmit a signal to the soma circuit. The soma circuit would determine the signal direction, implement the dendrite signal decay, time delay and summation of signals as well as determining the firing conditions. The time delay would be dynamically adjusted to simulate transponder movement as required by the learning algorithm to optimize future firings. The transponder design would be generic (all transponders are the same) and there would be no physical connections between the transponders - only the pulse signals are used for communication. Transponder location adjustments would be based on the sine and cosine of the predicted wavefront intersection angles and distances to simulate movement of the transponder but there is no reason dendrite locations couldn't move independently simulating irregular shapes of dendrite receivers and introducing more complex behaviors.

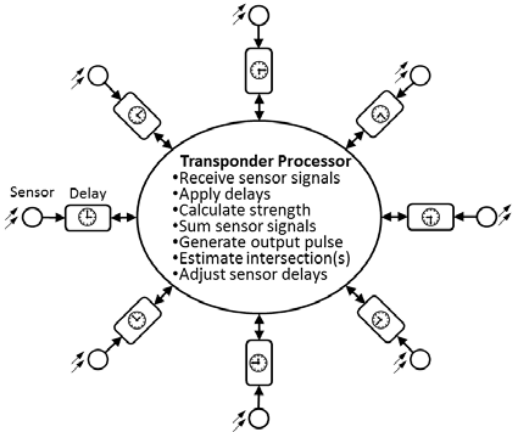

Figure 3 - Transponder Implementation Architecture.

The suggested implementation not only provides a practical approach to the physical instantiation of learning transponders, it also provides an alternate view of the method. The time delays associated with each sensor represent a form of synaptic weight for directional signal detection. This view suggests the STDP PWC learning approach, although conceptually based on movement of the transponders, can be mapped to weight adjustment methods used is other artificial neural networks (e.g., backpropagation or traditional STDP) and potentially allows future modifications of those methods to be applied to PWC configurations.

6. Experimental Results

A simulator for the combined PWC/Leaky Integrate-and-Fire Model and the modified STDP learning method was developed using NetLogo[13]. It utilizes an agent-based simulation approach to implement transponder agents that interact in a time-step simulation. The simulation supports loading predefined transponder configurations and the specification of all of the parameters discussed above to allow experimentation with model settings and exploration of behavioral characteristics. A visualization of the transponder configuration provides animation of wavefront propagation and transponder movement to observe learning behavior.

A number of training cases have been used in experiments including a simple "If A and B then C" case, a "Dual If A and B then C" case requiring learning two outputs simultaneously and a "Complex Dual If A then B" case that contains additional unused outputs to assess the behavior of the method in more complex situations. In addition, experiments have been conducted with more complex "Base 10 Addition Table" cases involving 20 input and 19 output nodes and a partial "Base 10 Multiplication Table" example involving 15 input, 28 hidden and 2 output nodes.

During tests, the learning algorithm moves the location of the output transponders to best match wavefronts generated by the test conditions. Tests have shown that repeated training scenarios result in a reduction of the distances to the wave intersections, an increase in the activation levels and a decrease in the position and activation level standard deviation. Depending on parameter settings, the learning process is able to adjust the output transponder position by approximately 4 distance units in 30 or less training cycles. More complex cases such as Base 10 Addition and Multiplication Tables are also being experimented with. The test cases have also show that transponders that "accidentally" activate are moved away from activation by the depression logic.

The behavior of the learning method and the model in general is highly dependent on the settings of the control parameters and the initial configuration of the transponders. The learning rate needs to be kept low to moderate by controlling the parameters A_+ (*potentiation-limit*) and τ_+ (*t-potentiation*). The τ_+ parameter used in the exponent of the equation is particularly sensitive. Similarly, the depression parameters A_- (*depression-limit*) and τ_- (*t-depression*) need to be used conservatively to prevent wavefronts from complex test cases undoing training.

In addition, since every transponder in the PWC model can see every other transponder, it is very sensitive wavefront interference from other subgroups in the configuration. Adjustment of the τ_s (*t-synapse*) parameter determines the time of decay of stimulations. For dense complex networks, this parameter needs to be smaller to prevent unwanted interactions in closely spaced transponders.

Stability and reproducibility of learned solutions is a key concern for any learning algorithm. In general consistent learning should occur over a reasonably large range of an algorithm's parameter space. For example, the size of the adjustment increment to the trajectory of the learning transponder (its potentiation limit) should not affect the ability of the algorithm to learn input patterns. To determine the extent to which the algorithm's convergence was sensitive to various initial parameters, the effect of changing A_+ (*potentiation-limit*) was explored on both the ability of the learning node to move to a position where coincident pulse inputs can activate the output node and the variability of the final position of this node. For a range of potentiation values (0.1 - 1.0) the trajectory of the learning transponder was recorded in a simple coincidence detection scenario, while also recording the response of the trained node uncoupled from any drive. The trajectories are plotted along with the corresponding activation levels in Figure 4. (Note, max(activation)=2.0 for a two wavefront case).

In all cases learning is initially rapid and converges to an x position. For lower values of the potentiation-limit (0.1 and 0.2) the convergence occurs more slowly to a change in position of approximately 4.5. At values from 0.4-0.94 it converges rapidly to a stable point at approximately 4.5. For values between 0.8 and 0.94 (the optimal setting indicated by a dashed line) convergence occurs in 5 training cycles. Above 0.94, however, the x position converges to larger values. This shows the effect of trying to learn too aggressively, overshooting the actual intersection and settling at another point of intersection, as there are a continuous set of these along the parabola of intersection for pairs of wavefronts[3]. As shown in Figure 2, the intersection prediction is an overestimate that needs to be compensated for in the A_+ setting. However, even at these settings, convergence eventually occurs. Convergence of the activation levels occurs with all setting as seen by the plot on the right. Regardless of the transponders final position, the node in question remains capable of performing a successful coincidence detection of firing from nodes A and B and improving its sensitivity to that condition.

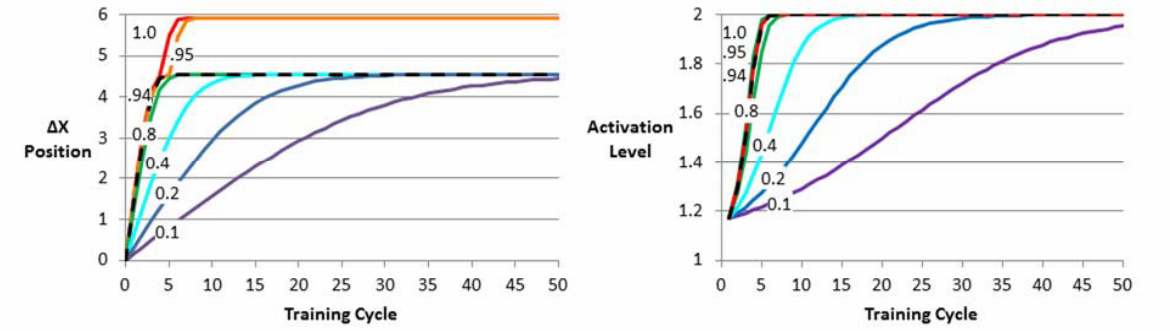

Figure 4 - Effect of potentiation limit on algorithm convergence.

7. Conclusions and Future Work

The work done to date shows that the method of relocating transponders to optimize their position with respect to stimulating waves is effective at implementing unsupervised learning in Polychronous Wavefront Computing configurations. The method requires that the basic PWC framework be modified to incorporate Leaky Integrate-Then-Fire mechanisms to allow the application of continuous adjustments and implement a STDP learning approach. This approach can both potentiate and depress transponder activation. Initial experiments with the model show that it is robust if model parameters are appropriate for the configurations and that learning is rapid due to the relative accuracy of the position estimates used.

Experimentation has just begun with this approach and additional research is planned to further explore its characteristics, refine the algorithms to improve their performance and explore more advanced uses of this method. Additional topics planned to be explored include:

- Additional experiments with different learning model parameter settings to further understand their effects on learning rates and quality
- Longer term tests involving hundreds and possibly thousands of learning cycles to assess the behavior over long term training
- Application to more complex PWC transponder configurations to explore issues with scaling of the approach
- Investigation into adjustments of the sensitivity by changing the membrane threshold (τ_m) as part of the learning approach
- Investigation of simpler intersection estimation algorithms that are less computationally complex and more biologically plausible than the line intersection estimates currently used.
- Experiments with multiple wavefront stimuli (>2 wavefronts)
- Creation of and experimentation with Polychronous Neuronal Groups (PNGs)[14]
- Implementation and testing of negative reinforcement mechanisms such as inhibitory transponders/pulses

The ultimate objective of this work is to provide a PWC model that can simplify the hardware implementation of neuromorhphic systems and is robust enough to support deep learning capabilities.

References

1. Yang, L., Zhang, L., and Ji, R. On-chip optical matrix-vector multiplier for parallel computation, *SPIE*; 2013. 10.1117/2.1201306.004932
2. DARPA. Systems of Neuromorphic Adaptive Plastic Scalable Electronics (SyNAPSE); 2014. http://www.darpa.mil/Our_Work/DSO/Programs/Systems_of_Neuromorphic_Adaptive_Plastic_Scalable_Electronics_(SYNAPSE).aspx
3. Izhikevich, E.M. and Hoppensteadt, F. Polychronous Wavefront Computations, *International Journal of Bifurcation and Chaos*; 2009. Vol. 19, No. 5:1733-1739. http://www.izhikevich.org/publications/polychronous_wavefront_computations.htm
4. Hart, C. Towards a Compiler for a Polychronous Wavefront Computer: Programming by Optimization, *Complex Adaptive Systems*, Procedia Computer Science; 2014. Vol. 36. http://www.sciencedirect.com/science/article/pii/S1877050914012599/pdf?md5=596c5892f02355715dbb15295b59b9d9&pid=1-s2.0-S1877050914012599-main.pdf
5. Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J. Learning representations by back-propagating errors. *Nature*; 1986. 323 (6088):533-536
6. Thomas, J.I. A Mathematical Treatise on Polychronous Wavefront Computation and its Application into Modeling Neurosensory Systems; 2014. viXra:1408.0104. http://vixra.org/pdf/1408.0104v1.pdf
7. Gerstner, W. and Kistler, W. *Spiking Neuron Models.* Single Neurons, Populations, Plasticity. Cambridge University Press; 2002.
8. Stein, R. B. Some Models of Neuronal Variability. *Biophysical Journal*, Volume 7, Issue 1; 1967. pp 37-68.
9. Song, S., Miller, K. D., & Abbott, L. F. Competitive Hebbian Learning Through Spike-timing-dependent Synaptic Plasticity. *Nature* Neurosci., 2000. 3:919-926. http://neurotheory.columbia.edu/~larry/SongNatNeuro00.pdf
10. Sutherland, B. 2013. http://www.economist.com/news/21589080-golden-rule-microchips-appears-be-coming-end-no-moore
11. Maass. W. Networks of Spiking Neurons: The Third Generation of Neural Network Models. *Neural Networks*; 1997. Vol. 10, No. 9. http://www.igi.tugraz.at/psfiles/85a.pdf
12. Monroe, D. Neuromorphic Computing Gets Ready for the (Really) Big Time, *Communications of the ACM*; 2014. Vol. 57 No. 6, Pages 13-15 10.1145/2601069. http://cacm.acm.org/magazines/2014/6/175183-neuromorphic-computing-gets-ready-for-the-really-big-time/fulltext
13. Wilensky, U. NetLogo. 1999. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. http://ccl.northwestern.edu/netlogo/.
14. Izhikevich, E.M. Polychronization: Computation with Spikes, *Neural Computation*; 2006. 18:245-282. http://www.izhikevich.org/publications/spnet.pdf