Advancing Deep Learning With Neuromorphic Computing: Review and Survey of Results

Giovanni Michel

***Abstract*—Since deep neural networks depend on the interconnections of neurons in the brain's information processing since the beginning it has pioneered the advancement of perceptrons and multi layer perceptrons. With new more powerful AI and more computationally expensive data sets modern Moore's law architectures can only support so much compute. This leads to a new implementations of innovative methods for delivering more computation. This paper is a review on Neuromorphic computing architectures that are related to Neuroscience can be bench marked and compared to modern NN architectures. This paper will also look to introduce other advancements of spiking neurons such as Optogentics and how membrane potential for neurons can affect surrounding neurons and how classical Hebbian learning explains that pre- and postsynaptic activity and other modulators that control these spikes can be used. Especially with Neuroscience being extremely useful to the advancement of NN's we can show that the Nature has solutions to our modern day Moore's law problems. This paper will also look to introduce topics on the beginner level for NN's so that correlation between conventional NN's and SNN's can be extracted. To understand SNN's Locally Competitive Algorithm (LCA) will be bench marked and demonstrated to show the energy efficiency, the fully integrated memory-and-computing and how it learns in adaptive processes associated with learning [2]. The LCA is essentially a Hopfield network is a energy normalization problem of the symmetric weight matrix where these networks are essentially analog-valued rate neurons, in a LCA the input signal is projected onto a set of feature neurons that are unrolled with a recurrent connection similar to how Elman RNN's work. Essentially the LCA is [7] "a Hopfield-style neural network that efficiently solves sparse approximation problems (e.g., approximating a vector from a dictionary using just a few non-zero coefficients)." It's also good to mention that compared to biological neurons, Recurrent NN's are the most similar for these reasons to be specific: it has a similar general architecture, similar temporal dynamics, and learning through weight adjustments but in the Neuroscience point of view that would be the synaptic connections of the different neurons. This will be conducted with a Introduction that will introduce our work and the motive, followed by Related Works from Dr. Carver Mead at Cal Tech, Mike Davies at Intel's Neuromorphic computing lab and Methodologies which will cover the past research and our methods for the experiments. We will also cover other SNN's Neuromorphic implementations of conventional ANN's algorithms and applications like Nearest Neighbor Search, Graph Search, Event-Based Sensing and Perception, and Closed-Loop Control for Robotics. The benefits for Neuromorphic computing are competitive with those results of ASIC's and GPU's in energy efficient acceleration of DL architectures.**

I. Introduction

College Of Engineering and Computer Science at Florida Atlantic University , Los Alamos National Laboratory Email: Giovannimich2020@fau.edu

Submitted: December 11, 2022

CURRENTLY, This paper was written in the intent to give a simple introduction into Neuromorphic Computing and Spiking Neural Networks. Originally in Machine Learning and Artificial Intelligence Multi layer Perceptrons have paved the way for new technology. Even Quantum Computing has made breakthroughs in a multitude of different disciplines just like AI. In our work we will introduce you to Neuromorphic Computing and how it can be used to advance AI which leads to more breakthroughs in technology. The main difference is one uses labeled data to help predict outcomes, while the other does not. We will also introduce concepts from Reinforcement Learning such as eligibility trace and demonstrating that Spiking Neural Networks (SNN) can learn from rewards. Especially since Reinforcement Learning is built around theories that originated in Psychology from watching living organisms interacting with it's environment and those interactions are measured by the good and bad things the organism experiences. We can also assume that methods originating from biology and Neuroscience can lead to benefits. The motive for Deep Learning in Spiking Neural Networks arises from the multiply-and-accumulate (MAC) from traditional ANN's. Some research focused on creating ANN's that reduce the MAC problem but essentially ANN's accelerators and expensive GPU's are the current solution to the complex compute in ANN's. For example, RNN's have "memory" of past features or "data" that can be used to make future decisions, in SNN's the memory that is used to keep track on whether a spike train from a pre-synaptic neuron trigger a post-synaptic spice can be used as some form of modulation given a post-synaptic spike happen based off of the previous feature embedding pre-synaptic spikes.

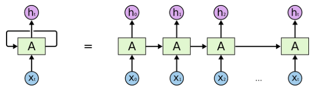


Fig. 1. Architecture for an RNN

II. Related Work

To begin with, Neuromorphic computing is an example of the principles from nature that introduced perceptrons and neurons which lead to the break-through of the Multi-layer Perceptron and then lead to challenging Moore's law and the von Neumann model. Neuromorphic computing was introduced when Dr. Carver Mead at Cal Tech wrote his book Analog VlSI and Neural Systems. In history dating back to 1950s, the Perceptron Mark 1 [1] was made with 16 analog neurons. Here he introduced biological concepts such as membranes in neurons, synapses, Transistor Physics and Transconductance Amplifier's. His research later led to new frameworks that can achieve the same performance when converting ANN's to SNN's. Other related works include [2] Urbanczik Senn when they developed a reinforcement learning learning rule using spike-time dependent plasticity (STDP) of population coding spiking neural networks. Another example of spiking neural networks where membrane potentials are in Optogenetics which was mentioned in [3] Pashai et al. (2014), where the action potential of neurons can be manipulated by different colors of light. In there work they mentioned how brain interfaces can be used to interact with the neurons that are interconnected and introduced the modulations of spiking neurons using Optogenetics which is a stimulation method that effects the activity of neurons using light. Sparse coding related research includes [6] where they introduced a sparse LCA that performed well when solving the LASSO problems.

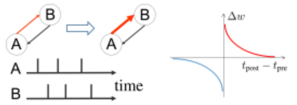
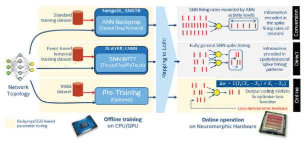


Fig. 2. Illustration of the STDP.

With RNN's being extremely popular for complex NLP and other information processing capabilities that are similar to the brain, [13] (Bellec, et al.,2020) developed a spiking RNN that tried to solve the solution to the learning problem of backpropagation in spiking neurons with there "e-prop" method. Bellec's work had great results. Pfister [14] (Pfister, Toyoizumi, Barber,Gerstner, 2006) and Gerstner used supervised learning to derive a synaptic update rule by gradient ascent and a temporal probabilistic distribution of the likelihood for post-synaptic neurons to fire at once or at several designated time duration's. There are only a few types of Neuromorphic chips in research such as Intel's Loihi FPGA Chip [2], SpiNNaker, and IBM's True North Neuromorphic chip. These chips are solving a diverse range of problems such as event-based data processing, constrained optimization, low precision and stochastic computation.


Network Topology
ANN Backprop
SNN BPTT
Pre-Training
Mapping to Loihi
Offline training on CPU/GPU
Online operation on Neuromorphic Hardware
Fig. 3. Deep learning methods in Neuromorphic computing for SNNs.

Other approaches to solving the Moore's law problem included Intel's AI Movidius stick but essentially the performance was not able to be as energy efficient with the increase in data and compute just like GPU's and other AI accelerators. Intel's Loihi chip turned a lot of heads in the Neuromorphic computing community when the low power consumption was bench marked with other traditional AI compute hardware such as NVIDIA GPU was tested against IBM's True North which outperformed all it's competitors with the lowest power consumption and still had the highest accuracy and lowest latency as the data increased.

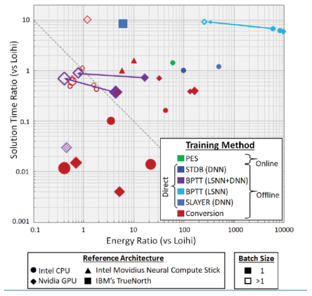


Fig. 4. Loihi energy consumption bench marked with IBM's True North Neuromorphic chip

III. Perceptrons and Activation Functions

Lets give some information on what a neural network is, basically it is artificial neuron called a perceptron which was developed in the 1950s and 1960s by Frank Rosenblatt, inspired by earlier work. To be honest the main neuron model used is called a sigmoid neuron. What the perceptron does it takes several inputs, x1,x2,. . . , and produces a output based on the features and the activation function:

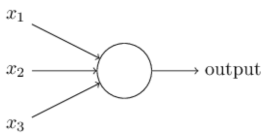


Fig. 5. Perceptron that has three inputs, x1, x2, x3

Where each neuron output is between 0 and 1, which is determined by the weighted inputs. This is a simple example of how perceptron can weigh up different kinds of factors in order to make decisions. Now when talking about a perceptron with multiple layers, we can train the weights and to learn classify different data by learning and then doing inference with what was learning. Each small change during learning in the weight will cause a small change in the output, sometimes it can change the output drastically which can lead to gradient vanishing or even gradient exploding. Sigmoid functions or sigmoid neurons are a type of perceptron, but are different because the output is smoother from the sigmoid is from 0 to 1 when compared to other activation functions such as Relu and tan. The main idea with perceptrons and Multi-layer Perceptrons is that small changes in their weights can lead to changes in their output.

IV. Understanding Spiking Neural Networks and Leaky Integrate and Fire Model

The above figure shows a Leaky-Integrate-and-Fire (LIF) neuron. A neuron receive a presynaptic spike in the form of a timestamp, if the neuron is not in the resting period the spikes are integrated and then it emits a postsynaptic spike whether the neurons membrane potential exceeded the voltage threshold for the neuron. The basis of the SNN is

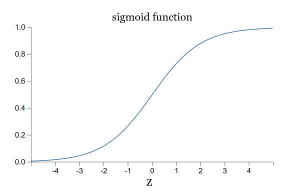


Fig. 6. sigmoid function shape visualized

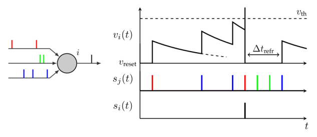


Fig. 7. A model of a LIF neuron

the Hebbian learning rule and that voltage spike trains from pre-synaptic and post-synaptic neurons will create a change in the synaptic activity which is from the time difference between the pre and post spikes. In the next sections we will cover how normalization of synaptic weights can lead to inference of Spiking Neural Networks. To achieve the same performance from standardized ANN's in SNN's for Neuromorphic computing let us first introduce the encoding methods for features in SNN's. The similar to the way a filter in CNN do local learning for the different features or even similar to when a data set is used as an input and is able to classify the different features using traditional FFNN's. For example, one-hot encoding is extremely well known in the ML community and is used in many different applications for solving problems that have large state or feature space representations an it becomes computationally inexpensive and inefficient to compute.

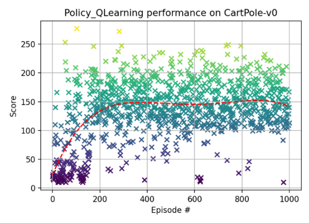


Fig. 8. Performance of One-Hot Encoded Q-Learning Cart-Pole Problem

The Swing-Up Cart-Pole is a complex problem in the field of Reinforcement Learning and Deep Learning because it involves two problems since the pole has to be swung up and then the second problem arises when the agent has to balance the pole. This is a type of problem that would benefit from some form of discretization of the state features which can be done using one-hot encoding or sparse coding. This concept is the similar to data compression where less data can still represent the same amount of information, this is helpful when the problem is extremely large and state space requires a lot of compute and memory. Another example would also be known as "bucketing" where you can categorize a feature into different buckets, in Deep Q-Networks they discretize the continuous dimensions to a number of buckets where essentially the goal is to have fewer buckets and keep the state space as small as possible. In our work we had the opportunity to implement one-hot encoded Q-Learning where the continuous state space is represented as sparse one-hot encoded input. In figure 5 it shows the performance of the one-hot encoded features can produce the same competitive performance to when no discretization, this arises the question on how encoding mechanisms can have benefits for ML and other AI algorithms. With knowledge of different discretization methods, the next section will cover more encoding concepts. The code for the above One-Hot implementation is available upon request. And for the remaining of the paper any code or visualizations will be cited if appropriate or can follow up with a request.

V. Feature Encoding and Basic Concepts: Hebbian and Modulated Plasticity

A. *Hebbian Theory and STDP*

The concept of Hebbian theory was introduced by [8](Hebb ,1949) in his hypothesis about the process of how synaptic strengths in the human brain "change in response to experience" [9] (Hertz, 1991). For mathematical derivations we can simply label the states for a neuron as +1 (fire) and -1 (no fire). The sgn(x) is illustrated as 1 if x greater than or equal to zero and -1 if x less than zero.

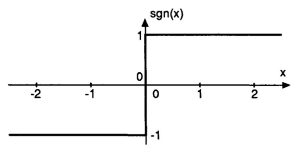


Fig. 9. The function Sgn(x).

The Hebbian theory essentially is an arbitrary function that uses sgn(x) to represent the activity of a specific neuron. This is mathematically convenient since the neurons can are represented by +1 or -1. This is important to point out because in the Hopfield model which was an important contribution to SNN's which led to the development of the LIF model was that the Hebb rule can be viewed as normalization problem or even an "energy function" in neural network theory. This energy function can be represented H and where the first term

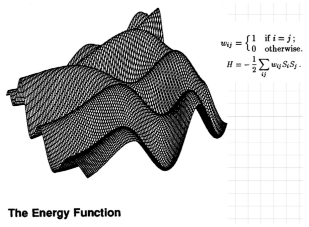
$$W_(ij)$$

can be represented as:

$$w_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} \xi_i^{\mu} \xi_j^{\mu} .$$

Fig. 10. The Hebb Rule.

Where the patterns for neuron u is summed for all p's and that each pattern u is still represented by -1(no fire) or +1(fire). This energy function has a certain property that makes it so valuable because [9]"it always decreases (or remains constant) as the system evolves according to it's dynamic rule", which means the memorized patterns are at some local minima at that energy surface. The energy function is also discussed in fields of magnetic system, that a state function will always decrease with continued dynamical evolution or will be normalized to find a stable optimum state.



$$w_{ij} = \begin{cases} 1 & \text{if } i = j\,; \\ 0 & \text{otherwise.} \end{cases}$$

$$H = -\frac{1}{2}\sum_{ij} w_{ij} S_i S_j\,.$$

Fig. 11. The Energy Function H with Hebb Rule for i and j and then the patterns u.

B. Learning Rule For Synaptic Activity

In this section we will have a set of questions that will be answered and will be covered. These questions include Hebbian Learning and STDP on how the two are related, Eligibility Trace and STDP on how the two are related. The most important principle originates from the idea that behavior and memory can be correlated to synaptic activity (Hebb, 1949; Barnes,1979; Morris et al., 1986; Bliss and Collingridge, 1993; Martin et al., 2000) which is a cause of the long-term potentiation (Lomo, 1964; Bliss and Lomo, 1973) and long-term depression (LTD) (Lynch et al., 1977; Levy and Stewart, 1983), or STDP (Markram et al., 1997; Bi and Poo, Stewart, 1998; Sjostrom et al., 2001). The theory of Hebbian STDP can be defended with the following equation:

$$\dot{w} = H(\text{pre}, \text{post})$$

Fig. 12. Hebbian Learning for Plasticity of synapse.

Where W describes the rate of change for the weight w and H is some function that represents the pre and postsynaptic spike trains. The above equation is similar to the Hebb rule derived in the previous section. In there work [10] (Fremaux and Gerstner,2016) formalized a simplified version of the Hebb rule in Figure 7. Since AI is built on principles from nature it is possible that ANN's and SNN's can learn using the same type of learning used in nature. (Arleo and Gerstner, 2000) in there research they demonstrated conceptual examples of a learning rule in nature, for example they conducted a test with the T-Maze in figure 10A.

In figure 10A-C we demonstrate [10] (Fremaux and Gerstner,2016) reward-modulated schematic for STDP where an animal (mouse) learns to perform desired actions in a T-Maze through trial-and-error with cheese (reward). They represented the position of the mouse in the environment by the activity of neurons in the hippocampus. The neuronal activity in the hippocampus was correlated with the spikes emitted from the motor cortex and the dorsal striatum. There work was a

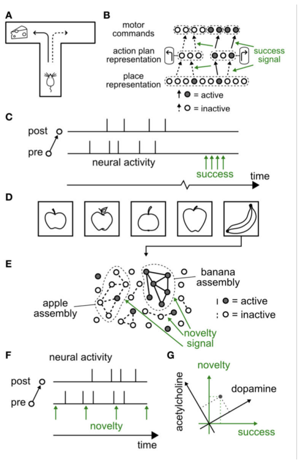


Fig. 13. Schematic of Reward-Based STDP.

good example of how individual LIF Spiking neurons learn and that validation of the learning can be measured in the "population" activity of the neighboring neurons. The research included giving an animal reward based signals based on the actions taken when navigating the maze. These type of animal experiments used a type of reinforce signal that is used for the animal to gain feedback from his actions. This reward feedback was mentioned in [8] (Fremaux, Gerstner, (2016)) where they expressed how "learning" happens with some type of reward based signal mechanism, they mentioned reward based learning can be accounted for in classical Hebb theory (Hebb,1949) assuming that co-activation of presynaptic neurons in the hippocampus with postsynaptic neurons from the striatum reinforces learning on the neuronal level where in NN models the feedback from the behavioral, reward signals are used in the synaptic level to "reinforce" the correct behavior of actions by modulating the Hebbian plasticity. The previous sentence states that granted the correct actions are taken whether it's picking fruit or navigating the maze, actions can be learned and rewarded which can insinuate learning.

C. Plasticity and Neuromodulation

In the previous sections we mentioned how neuron activity in the hippocampus and the striatum can induce learning because the synaptic activity being the indicator, now we our ready to introduce other learning rules that arise from Hebbian Learning and how they are being used to develop SNN's for Neuromorphic computing. We will now cover other Hebbian STDP learning rules and experimental evidence surrounding the topic. For example, the formalization of modulated Hebbian plasticity introduced the theory that Hebbian learning has two main factors, such as the presynaptic activity and the state of the postsynaptic neuron. With these two main factors we can introduce the neuro-modulator "three-factor rule", basically the three-factor synaptic plasticity rule incorporates the neuron-modulation plus the pre and postsynaptic activity:

Where w represents the weight change of a particular synapse. With this information we will now move onto to the next section where we further discuss reward-based learning and cover the RL in SNN's and how reward signals are

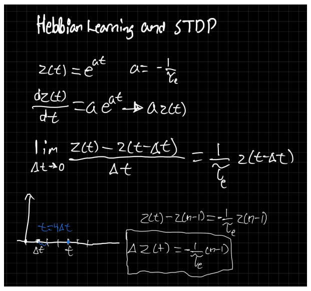
$$\dot{w} = F(M, \text{pre}, \text{post})$$

Fig. 14. Three Factor Synaptic Plasticity Learning Rule

used for Reinforcement Learning. After we will evaluate the performance results of SNN's, followed by experiments and discussion.

VI. Reinforcement Learning In Spiking Neurons

So far we have began to introduce these new temporal concepts of neuronal activity over some duration of time. These concepts are the building blocks for other Hebbian STDP theories in the field of Reinforcement Learning, ML for Unsupervised and Supervised learning, and even in the field of brain interfaces some part of Hebb theory is present. In Reinforcement Learning and biological features from natures traces of data that contains encoding features are extremely common. For example, in Richard Sutton's book for RL they cover Eligibility Trace and how it can be used to calculate the future n-step return using trace of past rewards in a memory vector. It is suitable to say that in SNN's feature encoding using the modulation of neurons can be done with traces or "Eligibility Traces" where synaptic activity is projected by whether or not a of pre-synaptic spike trains triggered a post-synaptic spike, meaning the post-spikes from the neuron are "eligible" for a change. More of this concept will be covered when we briefly cover Reinforcement Learning for Spiking Neural Networks and how rewards can be modulated in STDP. This trace can be represented as a function of time with the following equation:



Hebbian Learning and STDP

$$z(t) = e^{at} \qquad a = -\frac{1}{\tau_e}$$

$$\frac{dz(t)}{dt} = a e^{at} \rightarrow a z(t)$$

$$\lim_{\Delta t \to 0} \frac{z(t) - z(t-\Delta t)}{\Delta t} = \frac{1}{\tau_e} z(t-\Delta t)$$

$$z(t) - z(n-1) = -\frac{1}{\tau_e} z(n-1)$$

$$\boxed{\Delta z(t) = -\frac{1}{\tau_e}(n-1)}$$

Fig. 15. Eligibility Trace in STDP

With the above equation we introduce a temporal approach to calculating the eligibility trace of a neuron. We can see that eligibility trace is an exponential decay function for a constant time, this eligibility trace is used in standard RL as a memory vector. This memory can be encoded into spikes just like in our previous sections as mentioned in [10],[8]. [11] Was able to demonstrate a learning rule using population coding (encoding mechanism) of the population activity of neurons (whether that neuron spiked or not). There learning rule, demonstrated plasticity being modulated with a reward signal and a feedback signal of the encoded population activity "P", which is the sum of neuronal scores.

The figure above shows a schematic of an RL implementation using reward-modulated STDP using the population

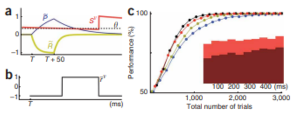


Fig. 16. Reward-Modulated STDP with Population Encoding

activity of spiking neurons. With all the past sections we are now were to move onto the section of Experiments and Results of SNN's and Discussions.

VII. Methods and Results

In this section we will show visualizations charts from our research and we will also show visualizations from referenced related works and there results when using SNN's. In [12] he demonstrated a sparse reward implementation of a multilayer SNN for two common RL algorithms: Q-Learning and Deep Q-Network, the work was able to benchmark two common RL algorithms for SNN's compared to DQN implementation for the Maze problem and the Acrobat problem. Our methods will demonstrate that pre- and post-synaptic neuron spikes of neuron compartments and layers will convert all activation function to any equivalent spiking implementation. Models with LIF neurons can be trained and evaluated in the same way as non-spiking models, and then trained to demonstrate proof of inference. Additionally we will introduce industry standard Neuromorphic chips and how SNN's perform "on-chip" with a Neuromorphic architecture I.e. Intel Loihi Neuromorphic FPGA and "off-chip" on a standard digital computer and benchmark the results from this research with baseline models and algorithms in ANN's. For our research no on-chip learning was conducted due to hardware constraints therefore off-chip learning results are substituted. Our data set that will be trained is standard MNIST and our for the SNN's the code was written in python using Nengo Pytorch. The evaluation metrics to measure the performance of our SNN implementation will include accuracy percentage for each image correctly classified and we will also measure the simulation time to compare that to modern GPU's. We trained our model for 10 epochs, the baseline models is a standard CNN with two layers, no pooling or or feature extraction was performed with the standard CNN. The SNN was also composed of a two layer NN with the same number of neurons for both models. We will also mainly focus on the training accuracy and the test accuracy scores.

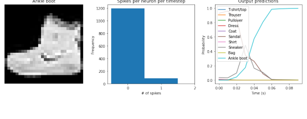


Fig. 17. SNN model Testing Results achieved 86.07 per cent accuracy.

In the above figure we demonstrate the performance of LIF neurons that are not on-chip. Essentially the results are competitive with standard ANN's. Some of the main important features when we developed the SNN model was the Simulation time because by adjusting the rate of change for time we compute temporal sparsity versus accuracy. And then also the Spike rate regularization since the model gains more control after spike trains by directly incorporating synaptic activity normalization into the optimization process of those neurons. Unfortunately the baseline model outperformed our off-chip spiking neural network implementation with an accuracy test score of 91.00 per cent.

VIII. Discussion

From the above results we can conclude that spiking neural networks and Neuromorphic computing has plenty of benefits for computation and advancing AI. We can say that from the above results that the off-chip spiking neural network approach is still competitive and can perform proof of learning and inference when fed a complex data set. Our study will also lead to future works I.e. on-chip learning with a Neuromorphic chip would be beneficial to understanding the limits to the low power consumption and fast parallel processing of Neuromorphic architectures.

IX. Conclusion

To Conclude, this paper was intended to introduce the audience to a another form of compute Neuromorphic computing, just like Quantum computing it has benefits for research and engineering that currently at our current stage we cannot full understand all it's capabilities. Future work would include running our SNN models on-chip Neuromorphic architectures. Another example would be to build a robot using Neuromorphic hardware and SNN's for the computing of the navigation using LIDAR SLAM and the motor encoders to navigate environments. This work was intended to simply do a review and introduce innovative advancements of Deep Learning while also demonstrating our research with SNN's for image classification.

ACKNOWLEDGMENT

Giovanni Michel Deep Learning Fall 2022 Professor Hill final project

1 Paredes-Vallés, Federico Scheper, Kirk Croon, Guido. (2019). Unsupervised Learning of a Hierarchical Spiking Neural Network for Optical Flow Estimation: From Events to Global Motion Perception. 2 M. Davies et al., "Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook," in Proceedings of the IEEE, vol. 109, no. 5, pp. 911-934, May 2021, doi: 10.1109/JPROC.2021.3067593. 3 Pashaie, Ramin, Polina Anikeeva, Jin Hyung Lee, Rohit Prakash, Ofer Yizhar, Matthias Prigge, Divya Chander, Thomas J. Richner, and Justin Williams. "Optogenetic Brain Interfaces." IEEE Rev. Biomed. Eng. 7 (2014): 3–30. 4 Richard S.Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction 5 Book by Alexander Zai and Brandon Brown, Deep Reinforcement Learning in Action 6 Tang, Ping Lin, Tsung-Han Davies, Mike. (2017). Sparse Coding by Spiking Neural Networks: Convergence Theory and Computational Results. 7 Balavoine, Aurèle Romberg, Justin Rozell, Christopher. (2012). Convergence and Rate Analysis of Neural Networks for Sparse Approximation. IEEE transactions on neural networks and learning systems. 23. 10.1109/TNNLS.2012.2202400. 8 Hebb, D. O. (1949). The organization of behavior; a neuropsychological theory. Wiley. 9 Hertz J., Krogh A., Palmer R. G. (1991). Introduction to the Theory of Neural Computation. Redwood City, CA: Addison-Wesley. 10 Frémaux N, Gerstner W. Neuromodulated Spike-Timing-Dependent Plasticity, and Theory of Three-Factor Learning Rules. Front Neural Circuits. 2016 Jan 19;9:85. doi: 10.3389/fncir.2015.00085. PMID: 26834568; PMCID: PMC4717313. 11 Urbanczik, Robert Senn, Walter. (2009). Reinforcement learning in populations of spiking Neurons. Nature neuroscience. 12. 250-2. 10.1038/nn.2264. 12 S. F. Chevtchenko and T. B. Ludermir, "Learning from Sparse and Delayed Rewards with a Multilayer Spiking Neural Network," 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9206846. 13 Bellec, G., Scherr, F., Subramoney, A. et al. A solution to the learning dilemma for recurrent networks of spiking neurons. Nat Commun 11, 3625 (2020). https://doi.org/10.1038/s41467-020-17236-y 14 Pfister JP, Toyoizumi T, Barber D, Gerstner W. Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. Neural Comput. 2006 Jun;18(6):1318-48. doi: 10.1162/neco.2006.18.6.1318. PMID: 16764506.