Milestone Report

Thomas Twomey
Department of Computer Science
Virginia Tech
Blacksburg, VA 24060
twomey@vt.edu

Abstract

This project is a literature survey of low-hardware cost Spiking Neural Networks as applied to classification networks, learning algorithms, and hardware acceleration. Spiking Neural Networks are relatively immature types of neural networks that are more closely representative of the neurons and synapses in the mammalian brain than typical Artificial Neural Networks. This similarity presents potential orders of magnitude power efficiency benefits. A variety of neural models, encoding schemes, and network structures are currently being studied with success similar to that of typical artificial neural networks and vast increases in power efficiency.

1 Contributions

All analysis, synthesis, and writing is my own. This past semester I participated in a partially overlapping research effort with the MICS lab at Virginia Tech and some of the referenced papers were found by other students and then shared with me. That project was more directed toward Field Programmable Gate Array implementations of spiking neural network acceleration hardware.

2 Motivation and Introduction

The growing prevalence of practical applications for neural networks and the increasing complexity of those networks is outpacing the increases in the computational devices used to train and run them. Most existing neural networks rely on the propagation of values between a large number of nodes and relatively expensive computation for each node. Using general purpose computing devices is no longer feasible, or will soon not be feasible. Much of the existing neural network computation is done with specialized accelerators such as consumer graphics cards and specialized silicon in the new apple chips. Neural networks running on these accelerators are pushing the boundaries of what is possible with Von-Neuamann style computation and the limitations of accessing memory. [Bouvier et al. 2019] Spiking, or event based, Neural Networks (SNNs) show potential to be much more power efficient and theoretically match the computational abilities of ANNs. [Maas 1997]

Derived from the interactions between human neurons and synapses, SNNs aim to accomplish the power efficiency observed in the brain. The large amount of computation done by the human brain consumes approximately twenty watts compared to the tens of thousands of watts required for the best hardware we have developed. These power efficiency gains can only be realised on specialised hardware which will be discussed later. As with much of machine learning, development of the theory has outpaced the development of the hardware and thus SNNs are often developed and trained with software simulation. SNNs are explored in a shared realm between computational neuroscience looking for insights into the brain and computer science looking for a practically applicable model.

This paper has been partially constrained to literature concerned with the recent developments of SNNs that handle classification tasks. Special attention was given to neural primitives that are

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

biologically-plausible and have known low cost hardware implementation. Further, for ease of comparison among papers, focus was placed on papers concerning the common classification task of the MNIST digit classification data-set.

3 Spiking Neural Network Background

3.1 General

Spiking Neural Networks(SNNs) are a form of Artificial Neural Network (ANN) that uses spikes, or impulses of a single bit, to send information between neurons. This means of sending information is thought to be more similar to the signaling of the brain and propagation of voltage through synapses. Further, the spikes are thought to partially explain the difference in power efficiency between human brains and Von-Neumann computers. [DeBole et al. 2019] [Bouvier et al. 2019]

3.2 Encoding

To use a SNN on most standard data-set and sensor inputs, the input values must be encoded from its original format to spikes by an input neuron. The two general techniques used for this encoding are rate-based, and temporal encoding. Neurons with rate based encoding create a spike train(series of spikes) with a density(frequency of spikes) proportional to the intensity of the input. Temporal encoding relates the intensity of the input to the time in which a single spike is sent from the input neuron. Rate-based encoding is more common in the literature while temporal encoding is thought to be more brain like and power efficient.[Panda et al. 2017] With fewer spikes there are fewer events and less computation has to occur. [P et al. 2020] A visual comparison of the two techniques is shown below.

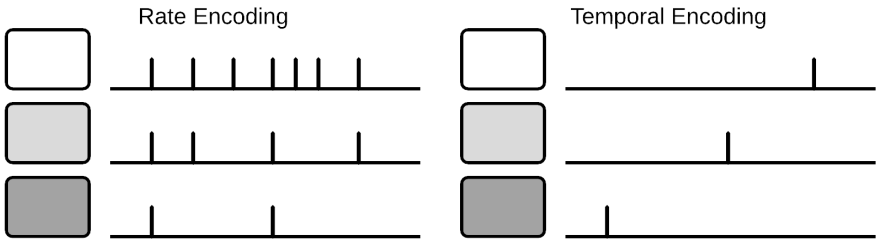

Figure 1: Encoding Techniques

3.3 Neural Models

Neurons or neural primitives, in the network are designed to mirror the functionality of human neurons. They have a membrane potential that is defined as a differential equation dependent on the current potential and input from the synapses connected to it. Neuroscientists have created many equations to enable neural primitives that emulate experimental findings. Integrate and Fire(IF) and Leaky Integrate and Fire(LIF) are the simple neural models that are most common in the computing literature. [Bouvier et al. 2019] Both of those algorithms have a membrane potential that stores that state of the the neural unit and a threshold. At that threshold the neural unit spikes and resets. Burkitt [2006] This membrane potential is increased by the synaptic weight associated with the firing or spiking of an input neuron. LIF extends the IF model with a leak rate that is a time dependent reduction in the membrane potential proportional to the difference of its current value and its resting value. These models lose some of the complexity associated with more advanced models but have produced networks with better classification accuracy. [Ali Samadzadeh 2020]

Formally, the standard LIF model is defined by:

$$\frac{dV}{dt} = (E_{rest} - V)C(E - V) \tag{1}$$

Visually the membrane potential can be thought of like something below:

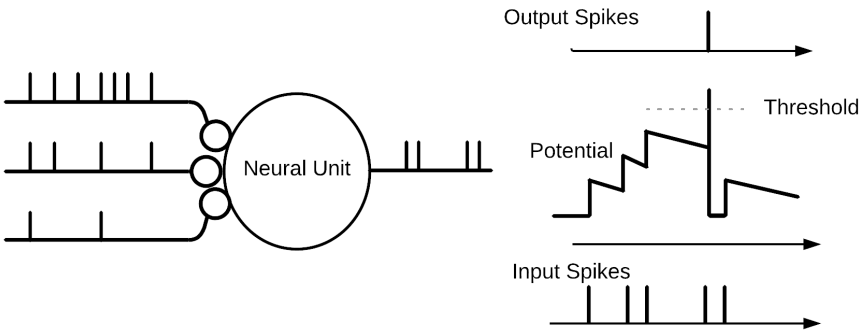

Figure 2: Leaky Integrate and Fire

The neuron will spike when a predefined threshold potential is reached. After this spike it returns to its resting potential and during the following refractory period it is inhibited from firing.[Diehl and Cook 2015][Burkitt 2006]

3.4 Topology

The range of possible network typologies in SNNs is relatively similar to that of typical ANNs. Classification SNNs use the same typologies as other machine learning algorithms, with convolution and fully connected feed forward networks being common.[Diehl and Cook 2015]

4 Hardware Accelerators

4.1 Analog

The brain can be thought to be more similar to an analog computer to a digital one. In the vein of replicating the functionality of the brain, researchers are also attempting to replicate its construction. Memristors are electronic devices that can in effect be programmed to have variable resistance and function vary similarly to a the synapses in the human brain. When organized in a crossbar fashion they become very useful in neuromphic chips to compute the weighted spike values. [Mukhopadhyay et al. 2018] This research is rather far from commercialization and various labs are looking to make better memristors and identify applications where they are most effective. [Bouvier et al. 2019]
Two chips, Neurogrid and BrainScaleS, are hybrid digital and analog implementations and are more aimed at simulating complex models of neural primitives instead of ubiquitous computational devices. The goal of these devices is narrow and they cannot generally be used to accelerate the classification tasks of SNNs. [DeBole et al. 2019]

4.2 Digital

Because of the difficulty of analog computation, and the relative maturity of CMOS digital implementations significant investment has been made in digital SNN accelerators. Essentially having many specialized cores, these devices map many software neurons to the necessary hardware for the neural primitive's computation. Between these neural processing units there is either a direct or packet based networking system to propagate the spikes to the next relevant neurons. [Bouvier et al. 2019][Davies et al. 2018] Intel, and IBM have both created similar digital implementations that follow this model. [DeBole et al. 2019] Intel's Loihi neuromorphic chip allows for online unsupervised and

reinforcement learning. Additionally because of the packet based networking structure many chips can be networked together to enable the simulation of many neurons. [Davies et al. 2018]

5 SNN Training

5.1 Supervised Learning

Training in a SNN differs from typical ANNs in that they are not differentiable and thus cannot be trained with standard supervised back-propagation. The two prominent methods to get around that non-differentiability are training a typical ANN and then using software to convert the weights to values that work for SNN or using function approximation of the neural models and then applying standard gradient descent and back propagation techniques.[Bouvier et al. 2019]

This conversion can be made with a relatively low cost to the accuracy of the network and significant reductions in the number of operations needed. [Rueckauer et al. 2017] It seems that research is moving away from conversion to training SNNs directly. The current state of the art SNN on MNIST is using function approximation of LIF neurons and achieves 99.4% accuracy. [Ali Samadzadeh 2020] With that level of accuracy, the paper claims there is no longer a need to go through the process of training ANNs and then converting them. There still may be a use case for this conversion methodology in converting legacy ANNs to SNNs when spiking hardware becomes more prevalent. There is no evidence that any means of back-propagation is biologically plausible or occurs in real neurons. [Legenstein et al. 2008]

5.2 Unsupervised Learning

The most common form of unsupervised learning the SNN literature is "Hebbian Learning" or Spike-Timing-Dependent Plasticity (STDP) [Lee et al. 2019]. Hebbian Learning comes from neuro-psychologist Donald Hebb and can be over simplified as "Neurons that fire together wire together." It is derived from empirical findings and theoretical reasoning of real biological neurons. There are a few different algorithms that implement it in practice, but the general idea is that for a neuron, the synaptic weight of an input is increased when that input event happens close in time to the output spike of the neuron. Various hardware accelerators have this functionality built in to allow for accelerated online training. It is difficult for STDP to train over multiple layers as the its inputs are inherently local.[Legenstein et al. 2008] STDP is also thought to be useful part of more complex training techniques. [Bouvier et al. 2019]

5.3 Reinforcement Learning

Reinforced learning has been implemented as Reinforced Spike-Timing-Dependent Plasticity (R-STDP) where the STDP determined change in synaptic weight is negated when the result is incorrect. [Mozafari et al. 2019] There is biological evidence that neurotransmitters can act as a signal to many neurons simultaneously and create a reward modulation effect. Further, R-STDP has been shown to work with both timing and rate encoding and can avoid some of the downfalls of STDP with locality [Legenstein et al. 2008]

6 MNIST Accuracy

For classification tasks a common data set among SNN and ANN papers is MNIST digit classification. Effort has also gone into creating a Neuromorphic-MNIST (N-MNIST) data set that covers the same set of training and testing samples as MNIST, but is natively spiking and mirrors the actual function of the human eye [Orchard et al. 2015]. MNIST results from the SNN and ANN literature are included in the chart below. MNIST may not be the best benchmark to show the performance capabilities of SNNs especially in areas where they outperform ANNs, but MNIST results are prevalent.

Table 1: MNIST Performance

Paper	MNIST Accuracy	Training	Topology	Neuron	Encoding
Byerly	99.84%	ANN			
Samadzadeh	99.4 %	Back-prop	CNN	LIF	Rate-Based
Mozafari	97.2%	R-STDP	3 layer DCSNN	IF	Intensity-to-latency
Diel and Cook	95%	STDP	2 Layer CNN	LIF	Poisson spike-train
Lee	91.1%	STDP	DSCNN	LIF	Poisson spike-train
Panda	>80%	STDP	CNN	LIF	Poisson spike-train

7 Conclusion

The scope of this project was perhaps too broad. It does not seem that the research is focused on improving in a single domain. Research in this area is sprawling in many directions with compelling applications and potential commercialization with many different permutations of the various techniques. It does seem clear that some sort of specialized hardware acceleration is needed to make SNNs significantly more useful than typical ANNs. Further while the best performance on MNIST is achieved with function approximation and back-propagation, STDP and R-STDP seem like plausible means of training networks that can also be implemented in embedded hardware. SNNs are a compelling candidate for the future of neural networks and making computers comparable to brains.

Broader Impact

Improving SNNs has the potential to produce much more efficient computer accelerators and gain insight to the functionality of the brain. More efficient chips allow lower energy consumption and thus more widespread deployment of advanced neural networks. This efficiency would affect embedded devices as well as data-centers, which are often constrained by their power efficiency. With better insights into the brain we get closer to replicating its structure and functionality and enabling more capable algorithms and computers.

References

A. J. A. N. M. H. C. Ali Samadzadeh, Fatemeh Sadat Tabatabaci Far. Convolutional spiking neural networks for spatio-temporal feature extraction. 2020. URL `https://arxiv.org/pdf/2003.12346.pdf`.

M. Bouvier, A. Valentian, T. Mesquida, F. Rummens, M. Reyboz, E. Vianello, and E. Beigne. Spiking neural networks hardware implementations and challenges: A survey. *J. Emerg. Technol. Comput. Syst.*, 15(2), Apr. 2019. ISSN 1550-4832. doi: 10.1145/3304103. URL `https://doi.org/10.1145/3304103`.

A. N. Burkitt. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological cybernetics*, 95(1):1–19, 2006.

M. Davies, N. Srinivasa, T. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y. Weng, A. Wild, Y. Yang, and H. Wang. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018. doi: 10.1109/MM.2018.112130359.

M. V. DeBole, B. Taba, A. Amir, F. Akopyan, A. Andreopoulos, W. P. Risk, J. Kusnitz, C. Ortega Otero, T. K. Nayak, R. Appuswamy, P. J. Carlson, A. S. Cassidy, P. Datta, S. K. Esser, G. J. Garreau, K. L. Holland, S. Lekuch, M. Mastro, J. McKinstry, C. di Nolfo, B. Paulovicks, J. Sawada, K. Schleupen, B. G. Shaw, J. L. Klamo, M. D. Flickner, J. V. Arthur, and D. S. Modha. Truenorth: Accelerating from zero to 64 million neurons in 10 years. *Computer*, 52(5):20–29, 2019. doi: 10.1109/MC.2019.2903009.

P. Diehl and M. Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9:99, 2015. ISSN 1662-5188. doi: 10.3389/

fncom.2015.00099. URL https://www.frontiersin.org/article/10.3389/fncom.2015.00099.

C. Lee, G. Srinivasan, P. Panda, and K. Roy. Deep spiking convolutional neural network trained with unsupervised spike-timing-dependent plasticity. *IEEE Transactions on Cognitive and Developmental Systems*, 11(3):384–394, 2019. doi: 10.1109/TCDS.2018.2833071.

R. Legenstein, D. Pecevski, and W. Maass. A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback. *PLOS Computational Biology*, 4(10):1–27, 10 2008. doi: 10.1371/journal.pcbi.1000180. URL https://doi.org/10.1371/journal.pcbi.1000180.

W. Maas. Networks of spiking neurons: The third generation of neural network models. *Trans. Soc. Comput. Simul. Int.*, 14(4):1659–1671, Dec. 1997. ISSN 0740-6797.

M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, S. J. Thorpe, and T. Masquelier. Bio-inspired digit recognition using reward-modulated spike-timing-dependent plasticity in deep convolutional networks. *Pattern Recognition*, 94:87 – 95, 2019. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2019.05.015. URL http://www.sciencedirect.com/science/article/pii/S0031320319301906.

A. K. Mukhopadhyay, I. Chakrabarti, A. Basu, and M. Sharad. Power efficient spiking neural network classifier based on memristive crossbar network for spike sorting application. *CoRR*, abs/1802.09047, 2018. URL http://arxiv.org/abs/1802.09047.

G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, 9:437, 2015. ISSN 1662-453X. doi: 10.3389/fnins.2015.00437. URL https://www.frontiersin.org/article/10.3389/fnins.2015.00437.

S. P, K. T. N. Chu, B. Amornpaisannon, Y. Tavva, V. P. K. Miriyala, J. Wu, M. Zhang, H. Li, and T. E. Carlson. You only spike once: Improving energy-efficient neuromorphic inference to ann-level accuracy, 2020.

P. Panda, G. Srinivasan, and K. Roy. Convolutional spike timing dependent plasticity based feature learning in spiking neural networks. *arXiv preprint arXiv:1703.03854*, 2017.

B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, 11:682, 2017. ISSN 1662-453X. doi: 10.3389/fnins.2017.00682. URL https://www.frontiersin.org/article/10.3389/fnins.2017.00682.