Fast and Low-Precision Learning in GPU-Accelerated Spiking Neural Network

Xueyuan She, Yun Long, Saibal Mukhopadhyay
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA
xshe6@gatech.edu, yunlong@gatech.edu, saibal.mukhopadhyay@gatech.edu

***Abstract*—Spiking neural network (SNN) uses biologically inspired neuron model coupled with Spike-timing-dependent-plasticity (STDP) to enable unsupervised continuous learning in artificial intelligence (AI) platform. However, current SNN algorithms shows low accuracy in complex problems and are hard to operate at reduced precision. This paper demonstrates a GPU-accelerated SNN architecture that uses stochasticity in the STDP coupled with higher frequency input spike trains. The simulation results demonstrate 2 to 3 times faster learning compared to deterministic SNN architectures while maintaining high accuracy for MNIST (simple) and fashion MNIST (complex) data sets. Further, we show stochastic STDP enables learning even with 2 bits of operation, while deterministic STDP fails.**

I. Introduction

Spiking neural network (SNN) are artificial neural networks (ANN) with biological plausible neuron and synapse models. The SNN has drawn significant attention in the field of artificial intelligence. In particular, SNN demonstrates the ability of unsupervised learning using Spike-Time-Dependent-Plasticity (STDP) in the synapse models. The STDP is a phenomenon observed in biology experiments [1], [2], which can be used as the synapse model of SNN. With STDP, synapse changes its conductance based on the time difference between pre-synaptic and post-synaptic spikes, enabling the learning ability of SNN. Several research efforts have explored STDP algorithms to enable learning in SNN [3] [4] [5]. Simulations of SNN has received significant attention in recent years to facilitate both understanding brain and develop AI algorithms. Early parallel SNN simulation tools like pGENESIS [6] required cluster computers to run. The recent developments focused on achieving SNN simulation with higher accuracy and better simulation speed [7]. In particular, advancements of Graphics Processing Units (GPUs) has led to feasibility of orders of magnitude improvement in computing speed of SNNs. Several SNN simulators with integrated STDP learning have been presented such as Brain [8], NEST [9] and CARLsim [10]. In a recent work, Long et. al. has presented a region-of-interest (ROI) based approach that vary complexity of neuron models based on spiking activity in a region, but they did not discuss STDP-based learning using SNN [11].

The existing SNN simulators such as NEST and CARLSim use deterministic STDP learning which suffers from several drawbacks, as shown in Section IV for details. First, while networks demonstrate good accuracy for simple tasks such as MNIST-based digit recognition [12], the learning accuracy for difficult tasks such as Fashion-MNIST [13] (images of apparel items that contains complex features) is much lower. Second, deterministic STDP provides limited opportunity for fast and low-precision simulation of unsupervised learning in SNN. For example, deterministic STDP for MNIST performed in 8-bit fixed-point (28%) shows significantly lower accuracy than floating-point (92%). This paper presents a GPU-accelerated SNN simulator, ParallelSpikeSim for high-accuracy, fast, and low-precision unsupervised learning. The key innovation of this paper is to demonstrate stochastic STDP for unsupervised learning in SNN, instead of well-explored deterministic STDP algorithms [3] [4] used in prior simulators. Moreover, we provide controllability to precision (down to 2-bit) with different rounding options, and frequency of input spike trains during unsupervised learning (and inference) in SNN to effectively exploit stochastic STDP for fast and low-precision learning. This paper makses following key contributions:

- We present the ParallelSpikeSim as a GPU-acclerated SNN simulator supporting unsupervised learning. The simulator is designed for parallel computing, programmed in C++ using CUDA libraries, and support different neuron/synaptic models.
- We demonstrate that the stochastic STDP allows good learning accuracy for both simple (MNIST [12], 96.1% accuracy) and complex (feature-rich) (Fashion-MNIST [13], 77.2% accuracy) data sets.
- We show that the stochastic STDP allows SNN to operate under input frequency ranges much higher than that of deterministic STDP design, and hence, enables up to 3x lower learning time with graceful quality degradation.
- We show that stochastic STDP enables robust learning from 32-bit floating-point (accuracy 96.1%) down to 2-bit fixed-point (accuracy 64.6%) learning, whereas deterministic STDP based SNN fails to provide meaningful results (accuracy drops from 92.2% to 9.6%).

II. Spiking Neural Network Models

A. Spiking Neuron Model

The spiking neuron model used in this work is leaky integrate-and-fire (LIF). For LIF model, membrane potential of a neuron is described by:

$$dv/dt = a + bv + cI \tag{1}$$

$$v = v_{reset}, \text{ if } v > v_{threshold} \tag{2}$$

978-3-9819263-2-3/DATE19/©2019 EDAA

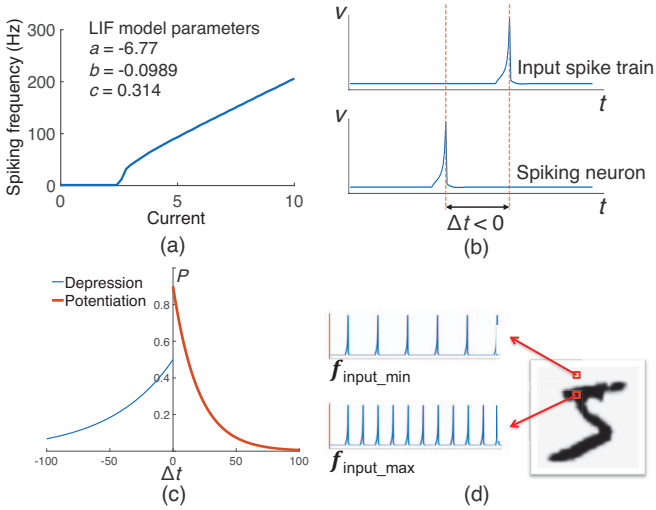


Fig. 1: Neuron models: (a) Spiking frequency vs. input current of LIF neurons, (b) spiking behavior (c) synaptic behavior under stochastic STDP, (d) Conversion of input image to spike trains;

The current I received by neuron a is described by:

$$I_a = \sum_{n=0}^{N} g_{n,a} v_{pre_n} \tag{3}$$

N is the total number of pre-neurons connected to neuron a. $g_{n,a}$ is the conductance of the synapse connecting neuron a and its pre-neuron n. v_{pre_n} is the voltage spike of pre-neuron n. Fig. 1 (a) shows change of spiking frequency of LIF model with parameters used in this work with respect to different input current.

B. Synapse Model

In an SNN, a synapse connects two neurons, which are referred to as the pre-neuron and post-neuron. The synapse transmits signals when its pre-neuron spikes and sends current signal to its post-neuron. The conductance of synapses is the weight of the connection and learning is achieved through modulating the conductance of synapses. Conductance modulation algorithm used in this unsupervised learning architecture is STDP. With STDP, magnitude and direction of each synaptic operation is determined by the temporal relationship between the spiking activities of the pre-neuron and post-neuron. When post-neuron spikes closely after a pre-neuron spike, the synapse experience long-term potentiation (LTP); when post-neuron spikes before a pre-neuron spike, the synapse experience long-term depression (LTD). As a result, the level of causal relationship between two neurons can be encoded as conductance of the synapse. Using model introduced in [4], conductance is modulated by equations

$$\Delta G_p = \alpha_p e^{-\beta_p (G - G_{min})/(G_{max} - G_{min})} \tag{4}$$

$$\Delta G_d = \alpha_d e^{-\beta_d (G_{max} - G)/(G_{max} - G_{min})} \tag{5}$$

ΔG_p is the increase of conductance for potentiation operations, and ΔG_d is the decrease of conductance for depression operations. α_p, α_d, β_p, β_d, G_{max} and G_{min} are parameters that are tuned based on input spiking frequency and voltage, as well as bit-width under which synapses are operating.

C. Stochastic behavior of synapses

For synapses with stochastic STDP behavior, potentiation or depression of synapses is not deterministic, but has a probability that depends on the time difference of the two spike events that initiates the modulation of conductance. For instance, as shown in Fig. 1 (b), Δt is below zero when spiking neuron spikes before a spike from input train arrives at the synapse. Stochastic STDP is achieved with an algorithm inspired by the work of Srinivasan [14]. The probabilities for potentiation and depression are defined by:

$$P_{pot} = \gamma_{pot} e^{(-\Delta t/(\tau_{pot}))} \tag{6}$$

$$P_{dep} = \gamma_{dep} e^{(\Delta t/(\tau_{dep}))} \tag{7}$$

The probabilities are exponentially related to time difference, with maximum value controlled by γ_{pot} and γ_{dep}, as shown in Fig. 1 (c). In the event of potentiation, the probability is higher when Δt is smaller, indicating a stronger causal relationship. As for depression, the probability is higher when Δt is larger.

III. Experimental Platform

A. Design of the Simulator

Fig. 2 shows the flowchart of the unsupervised learning architecture with SNN achieved with ParallelSpikeSim. The SNN simulator has two major components. First, a spiking neuron simulator to simulate the differential equations governing the neuron dynamics (i.e. equations (i.e. (1),(2),(3)) for a given synapse conductance. The second component is the learning module that implement the synaptic models and allow synapse conductance to be updated based on the spiking activity using STDP rules (i.e. equations (4),(5)). Many past SNN simulators (e.g. CARLSim, BRAIN, NEST, etc.) includes the STDP based learning modules. The key innovation in ParallelSpikeSim is to augment the learning modules to include stochastic STDP (i.e. equations (6),(7)) and various precision control and rounding options (see Section III-C). Moreover, we introduce an additional module between input images and spiking neuron simulator that allows controlling the frequency of the input spike train as shown in Fig. 2.

CPU serves as data I/O and controls data flow of GPU. It constructs the simulation environment with configuration and input data file, allocate memory and transfer data in unified data structures to GPU memory when simulation starts. The unified data structures of ParallelSpikeSim encapsulate all network information into the network object and all input into the data object, to facilitate swift addition of functionality and customization of network hierarchy, layer connectivity and behavior of each synapse and neuron. After initialization, simulation of spiking neurons runs in parallel on GPU threads.

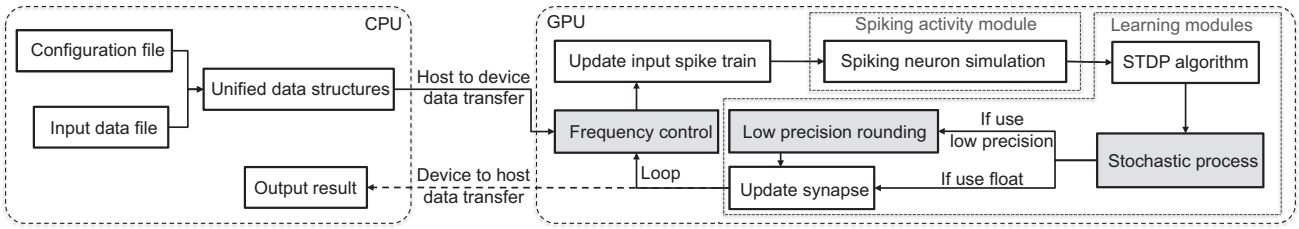


Fig. 2: ParallelSpikeSim: a GPU accelerated SNN simulator with stochastic STDP, low precision learning and frequency control module

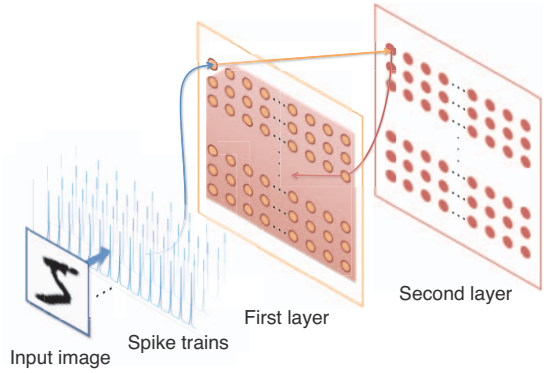


Fig. 3: Network architecture of the SNN implemented in this work.

Stochastic STDP module uses spike timers to track the temporal relationship between pre-synaptic and post-synaptic spikes, and performs stochastic process on-board the GPU to leverage the fast CUDA random number generator. Low precision learning module operate with reduced bit-width down to 2 bits, and has three available rounding options: bit truncation, rounding to nearest and stochastic rounding. Frequency control module works in two phases: frequency boost and learning time reduction. More details about the impact of the three modules are discussed in the following sections.

B. Network Architecture and Configuration

We implement the SNN architecture shown in Fig. 3 to demonstrate the proposed simulator. Input image is converted to a spike train array (one spike train per pixel) connected to the first layer of neurons. The frequency of the spike trains can be controlled. The first layer consists of 1000 spiking (LIF) neurons. Each neuron receives signal from the input spike trains and when it spike, it sends excitatory signal to one neuron in the corresponding position on the second layer. The excitatory signal activates the second layer neuron, which then sends inhibitory signal to all other neurons on the first layer for t_{inh} amount of time. This inhibition behavior achieves a winner-take-all (WTA) principle, preventing more than one neuron to learn one specific pattern. Input spike trains and first layer are connected by synapses in an all-to-all fashion. Conductance of each synapse connecting input to first layer neurons collectively forms a conductance array that learns to recognize a specific pattern.

The causal relationship between pre-synapse and post-synapse neurons explored by STDP algorithm makes it possible for the network to achieve unsupervised learning. The MNIST and Fashion-MNIST data sets contain 60,000 training images and 10,000 testing images. In this work, the SNN learns the full set of training images. Pixel intensity of input images, which is an 8-bit value, is encoded into specific spiking frequency of one spike train. For darker pixels, the spiking frequency is higher, as shown in Fig. 1 (d). Frequency is in a range between f_{input_max} and f_{input_min}, and proportional to the pixel intensity. Each image is presented to the network for t_{learn} ms. After learning is complete, the first 1000 images in the test set are used to label all the neurons in the first layer. The rest of the test set, which contains 9000 images, are used for inference.

C. Low precision learning and rounding options

For low precision learning, conductance of synapses is represented in numbers with precision no greater than 32 bits. Quantization for low precision learning is performed before the LTP/LTD phase of synapse conductance. For 16-bit and 32-bit learning, after floating point calculation of change in conductance ΔG, the result is rounded to a value that can be represented in the current bit width. For 8-bit and lower precision learning, ΔG is set to $1/2^n$, with n being the bit width. Low precision learning in other neural networks such as recurrent neural network (RNN) is shown to have different performance with different rounding options [15]. For low precision learning in SNN, we study the impact of rounding options to determine if there exists a similar influence. Three rounding options including rounding to nearest, bit truncation and stochastic rounding are tested in this work. For stochastic rounding, the probability of rounding up is related to the position between the two quantized value, and is defined as:

$$P_{round\_up} = (\Delta G - \Delta G_{truncated}) \times 2^n \tag{8}$$

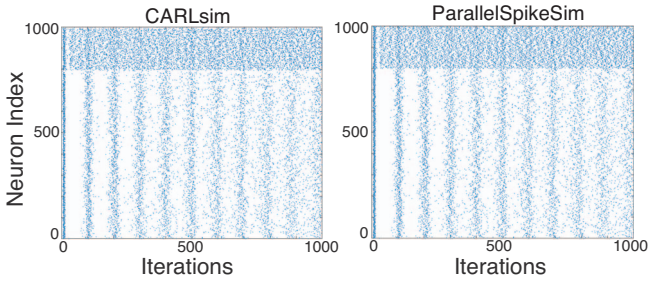
When a value does not round up in the stochastic operation, it rounds down automatically, i.e. probability of round down is $(1 - P_{round\_up})$.

D. Parameters and other details

For the LIF model used in this work, V_th is -60.2, V_reset is -74.7, a is -6.77, b is -0.0989 and c is 0.314. Parameters for STDP algorithm and stochastic behavior of synapses in different precision learning are shown in TABLE I. Initial membrane potential of all neurons on the first layer is -70.0 and conductance of each synapse is initialized with

TABLE I: Parameters for different learning options

	α_P	β_P	α_D	β_D	G_{max}	G_{min}	γ_{pot}	τ_{pot}	γ_{dep}	τ_{dep}	f_{input_max}	f_{input_min}
2 bit	-	-	-	-	-	-	0.2	20	0.2	10	22	1
4 bit	-	-	-	-	-	-	0.3	30	0.3	10	22	1
8 bit	-	-	-	-	-	-	0.5	30	0.5	10	22	1
16 bit	0.01	3	0.005	3	1.0	0	0.9	30	0.9	10	22	1
high frequency	0.01	3	0.005	3	1.0	0	0.3	80	0.2	5	78	5



Simulator	CARLsim [9]	ROI [10]	ParallelSpikeSim
Simulation time	109 ms	91 ms	114 ms

Fig. 4: Simulation of spiking activity and performance

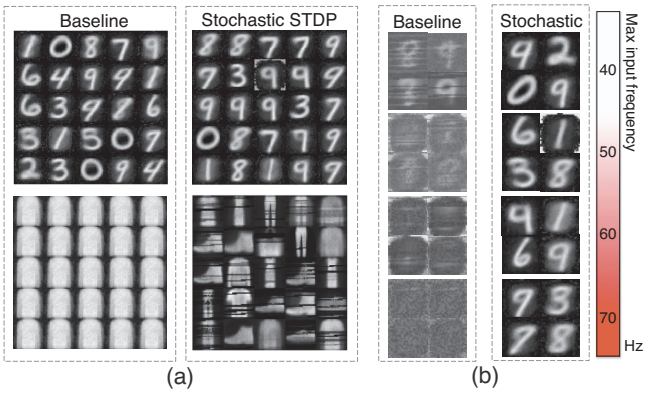


Fig. 5: Visualization of synapse conductance (a) Baseline and stochastic STDP for MNIST and Fashion MNIST, and (b) effect of input spike train frequency on stochastic STDP.

random number in the range of 0.2 to 0.7. Simulations are performed on a desktop machine with Intel Core i7-7700k and NVIDIA GTX 1080 Ti.

IV. Experimental Results

A. Accuracy Comparison with Existing Simulators

We first evaluate the accuracy of the spiking neuron simulation (no learning) considering an SNN of 10^3 LIF neurons and 10^4 synapses. Fig. 4 shows that our platform is able to produce spiking activities similar to CARLsim [10]. However, we observe an increased simulation time in ParallelSpikeSim due to the use of more complex unified data structures. The impact of this increased spike simulation time is overshadowed by the higher learning rate achieved using stochastic STDP. We next verify that deterministic STDP (defined as baseline) in ParallelSpikeSim shows comparable accuracy with the state-of-the-art SNN design with deterministic STDP from Diehl [3]. In Diehl's work, the network yields an accuracy of 91.9% for the MNIST data set while our baseline test achieves 92.2%.

B. Improved Learning Accuracy with Stochastic STDP

We observe that both baseline and stochastic STDP are able to produce good accuracy in learning the MNIST data set. Each is able to provide conductance array with good contrast for each class of image as can be seen in Fig. 5 (a). Inference result shows that Stochastic STDP is able to provide better result with around 4% higher accuracy. However, for the Fashion MNIST data set, baseline test fails to gain accuracy even after learning all 60,000 images. Visualization in Fig. 5(a) shows that all synapses learns the overlapping features of all classes. On the other hand, stochastic STDP is able to learn the more complex data set. Comparing visualization of synapse conductance from stochastic STDP learning on the right of Fig. 5 (a), baseline design struggles to learn any unique features from input images. This result shows that the level of causal relationship implied by stochastic STDP provides SNN with additional learning ability, and this effect is more prominent in more complex learning tasks.

C. Fast Learning with Higher Input Frequency

ParallelSpikeSim allows controlling the frequency of the input spike train to enable trade-off between learning rate and accuracy. One of the bottleneck of SNN learning rate is the time it takes to learn features in each individual image. Due to the influence of inhibition period of the WTA principle, and inherent nature of spiking neurons such as the reset of membrane potential after spikes, learning requires each image to be presented to the network for an extended period of time, so that a sufficient amount of spikes are generated. Since information in this spiking neural network architecture is transmitted in form of spikes, the more frequently spikes can be sent, the faster information can be delivered. Therefore it is desirable to make the spike train operate with higher frequency.

On the other hand, a higher input spike frequency can degrade the learning accuracy. Fig. 7 (a) shows learning accuracy loss for different input spike train frequency. We observe that using a value of f_{input_max} above certain value will cause the network to drop sharply in accuracy. This is because at higher input frequencies, the rapid arriving current signal drives multiple spiking neurons to spiking state disregard of their previous learned features, making the inhibition layer

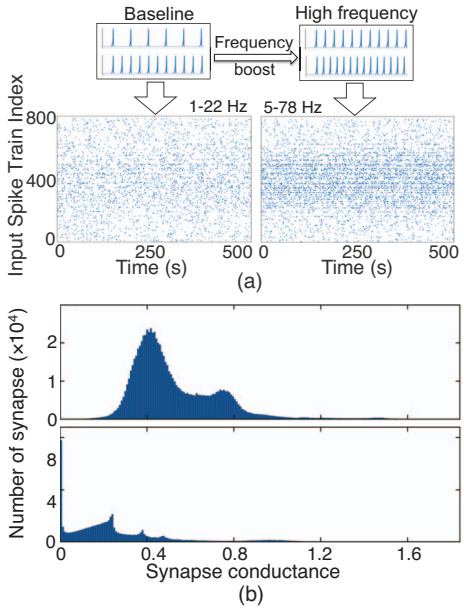


Fig. 6: High-frequency and low-precision operations (a) Input spike trains at low (left) and high (right) frequencies (each dot represents one spike). (b) Conductance distribution of Q1.7 precision (MNIST) with stochastic (top) and deterministic STDP (bottom).

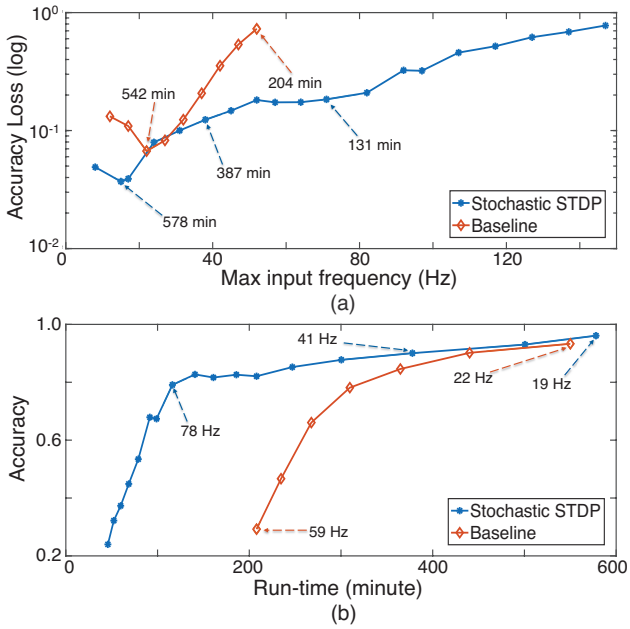


Fig. 7: High-frequency learning (a) Accuracy loss vs. max input frequency, and (b) Accuracy vs. run-time.

less useful and the network gradually shifts to chaotic states. This effect can be observed in the conductance visualization of four frequency ranges shown in Fig. 5 (b). As a result, for SNN using deterministic STDP, the optimal f_{input_max} is limited to a relatively low value. In baseline test, the optimal spiking frequency range of neurons in the input layer is 1-22 Hz. At such frequency range, 500 ms learning time for each image is used in order to generate sufficient spikes. For a total simulation time of 542 minutes the baseline architecture is able to learn the 60,000 MNIST images.

In this work, we find that using stochastic STDP with short-term behavior, working frequency range of f_{input_max} can be expanded, as can be seen in Fig. 5 (b). More specifically, higher τ_{pot} and lower τ_{dep} values for (6) and (7) are used to create a short-term stochastic STDP behavior, which enhances its ability to adapt to the fast switching input feature. We find the frequency range with maximum error rate of 20%, and the result is f_{input_min}-f_{input_max} at 5-78 Hz. Comparing Fig. 6 (a) left and Fig. 6 (a) right, which show spike train behavior of baseline and high frequency learning for the MNIST data set, it can be observed that the pattern of darker region, where the written digit is located, is more distinct in high frequency learning. Learning efficiency is therefore significantly increased as time for the network to learn features in each image is reduced. In this high frequency learning mode, frequency range of input spike train can be expanded up to 5-78 Hz before significant decrease of accuracy occurs. At this frequency range, learning time for each image is reduced to 100 ms, leading to a total simulation time of 131 minutes to learn the entire MNIST data set. In baseline test, as shown in Fig. 7 (b), achieving the same accuracy takes around 380 minutes, around 3 times longer than stochastic STDP.

D. Low precision learning

In this work, we performed learning in 2, 4, 8 and 16 fixed point numbers. Baseline test shows poor accuracy result for low precision learning as shown in Table II. This is due to the fact that quantization of conductance in low precision learning increases gap between adjacent conductance values. This leads to rapid changes in conductance during LTP/LTD process and the network quickly lose memory of learned features. This effect is shown in Fig. 6 (b), which is the distribution of conductance of all 784,000 synapses connecting input and first layer, for Q1.7 precision learning of the MNIST data set. Distribution of stochastic STDP is on the top of Fig. 5 (b) and deterministic STDP on the bottom. Deterministic STDP results in less ideal distribution, in which a large portion of synapses drops to the minimal conductance value. Stochastic STDP in synapse model greatly improves accuracy in low

TABLE II: Accuracy results (%) for rounding options

	Truncation	Rounding to nearest	Stochastic
Baseline			
Q0.2	9.6	11.3	16.8
Q0.4	13.1	16.3	21.3
Q1.7	28.2	30.8	33.7
Q1.15	52.6	52.8	55.2
Stochastic			
Q0.2	62.3	66.7	64.6
Q0.4	72.4	77.6	79.0
Q1.7	88.5	91.1	90.1
Q1.15	93.2	94.2	94.7

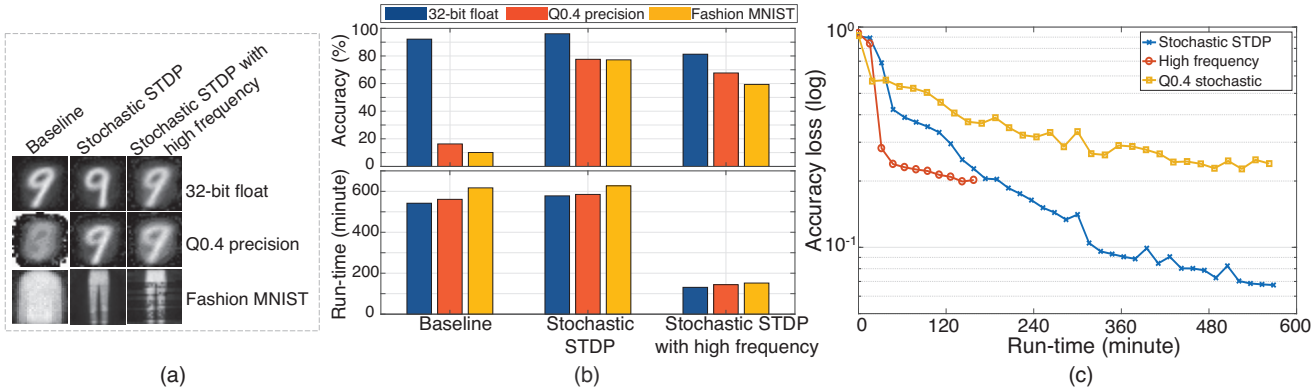


Fig. 8: Comparison of learning configurations: (a) conductance map, (b) Accuracy and run-time, and (c) accuracy loss vs. simulation time

precision learning as it prevents rapid changes from loosely correlated spiking events to help retain memory and at the same time guard the network from fast convergence. The improvement is present in all precision tested, as shown in Table II. Low precision learning, especially the ones with Q1.7 and lower, exhibit larger gain in accuracy from the application of stochastic STDP. Such robustness observed in this SNN design is important as there are many well-known benefits for digital systems to operate in lower bit-width, including less memory usage and less power consumption.

As shown in Table II, for learning in different precision, accuracy drops significantly from Q1.15 to lower precision fixed point. Three rounding options for low precision learning tested exhibit different learning performance. Bit truncation shows the lowest accuracy in all precision tested, while stochastic rounding performs the best in most cases. This is because in low precision learning, stochastic rounding helps to maintain information about numeric position between two quantization points on a statistical point of view. It is also worth noting that stochastic rounding and round to nearest shows similar results for network using stochastic STDP, and the benefit of using stochastic rounding diminishes as bit width increases.

E. Summary of Results

Fig. 8 summarizes the comparison of comparison of baseline and stochastic STDP learning results. Stochastic STDP shows higher accuracy in challenging learning tasks and lower precision while using similar simulation time as baseline. The high frequency learning (with stochastic STDP) greatly reduces learning time (moving error rate reduces quickly as shown in Fig. 8 (c)) but with a graceful degradation of final accuracy.

V. Conclusions

This paper has presented ParallelSpikeSim, a SNN simulator with unsupervised learning using stochastic STDP. We show that ParallelSpikeSim enables accurate learning for complex tasks, enables fast learning, and allows low-precision operation even down to 2 bits. The ParallelSpikeSim will be released as an open-source, GPU-based SNN simulation platform to help researchers explore applications of SNN to various AI problems.

Acknowledgment

This work was supported in part by the Office of Naval Research Young Investigator Program.

References

[1] W. B. Levy and O. Steward. Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. *Neuroscience*, 8(4):791–797, 1983.

[2] B Gustafsson, H Wigström, W C Abraham, and Y Y Huang. Long-term potentiation in the hippocampus using depolarizing current pulses as the conditioning stimulus to single volley synaptic potentials. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 7(3):774–80, 1987.

[3] Peter Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9(August):99, 2015.

[4] Damien Querlioz et al. Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Transactions on Nanotechnology*, 12(3):288–295, 2013.

[5] Michael Beyeler et al. Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule. *Neural Networks*, 48:109–124, 2013.

[6] J M Bower, D Beeman, and M Hucka. The GENESIS simulation system. *The Handbook of Brain Theory and Neural Networks*, (August 2000):475–478, 2003.

[7] Romain Brette, Michelle Rudolph, Ted Carnevale, et al. Simulation of networks of spiking neurons: A review of tools and strategies, 2007.

[8] Dan F.M. Goodman and Romain Brette. The brian simulator, 2009.

[9] Susanne Kunkel, Susanne Kunkel, Abigail Morrison, et al. Nest 2.12.0. *doi.org*, pages –, 2017.

[10] Ting Shuo Chou et al. CARLsim 4: An Open Source Library for Large Scale, Biologically Detailed Spiking Neural Network Simulation using Heterogeneous Clusters. *IJCNN*, 2018.

[11] Yun Long, Xueyuan She, and Saibal Mukhopadhyay. Accelerating biophysical neural network simulation with region of interest based approximation. *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 159–164, 2018.

[12] Yann LeCun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998.

[13] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR*, abs/1708.0, 2017.

[14] Gopalakrishnan Srinivasan, Abhronil Sengupta, and Kaushik Roy. Magnetic Tunnel Junction Based Long-Term Short-Term Stochastic Synapse for a Spiking Neural Network with On-Chip STDP Learning. *Scientific Reports*, 6, 2016.

[15] Taesik Na, Jong Hwan Ko, Jaeha Kung, and Saibal Mukhopadhyay. On-chip training of recurrent neural networks with limited numerical precision. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2017-May, pages 3716–3723, 2017.