

Improving Robustness of ReRAM-based Spiking Neural Network Accelerator with Stochastic Spike-timing-dependent-plasticity

Xueyuan She, Yun Long, Saibal Mukhopadhyay

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA
xshe@gatech.edu, yunlong@gatech.edu, saibal.mukhopadhyay@ece.gatech.edu

Abstract—Spike-timing-dependent-plasticity (STDP) is an unsupervised learning algorithm for spiking neural network (SNN), which promises to achieve deeper understanding of human brain and more powerful artificial intelligence. While conventional computing system fails to simulate SNN efficiently, process-in-memory (PIM) based on devices such as ReRAM can be used in designing fast and efficient STDP based SNN accelerators, as it operates in high resemblance with biological neural network. However, the real-life implementation of such design still suffers from impact of input noise and device variation. In this work, we present a novel stochastic STDP algorithm that uses spiking frequency information to dynamically adjust synaptic behavior. The algorithm is tested in pattern recognition task with noisy input and shows accuracy improvement over deterministic STDP. In addition, we show that the new algorithm can be used for designing a robust ReRAM based SNN accelerator that has strong resilience to device variation.

Index Terms—ReRAM, spiking neural network, spike-timing-dependent-plasticity(STDP), process-in-memory(PIM)

I. INTRODUCTION

Spiking neural network (SNN) is a neuromorphic computing paradigm that mimics the behavior of animal nervous systems at the level of neurons and synapses. The development of SNN promotes deeper understanding of cognition system, and at the same time serves as a potential approach to achieving artificial neural networks (ANN) that are as efficient as the human brain, which outperforms state-of-the-art ANNs with only fraction of their power consumption [1]. Originating from the synaptic modulation rule first studied by Hebb [2], which has been the foundation of research in learning and memory ever since 1949, spike-timing-dependent-plasticity or STDP [3], is now a widely adopted weight update algorithm in SNN. The temporal relationship between spiking events in the network as determined by STDP makes it possible for SNN to achieve learning with unlabeled input data, i.e. learning is unsupervised. In fact, SNN with STDP learning rule has been used in various computer vision related machine learning tasks [4]–[6] and shows accuracy results comparable with supervised neural network designs [7] [8].

Conventional SNN accelerators suffer from the "memory wall" of von Neumann architecture, which hinders the development of systems with higher performance as memory access speed becomes the bottleneck. One promising solution to this problem, process-in-memory, or PIM, has been actively

studied in the design of novel neural network accelerators. The integration of memory and computation supports potential breakthrough of the bandwidth barrier, and can fully take advantage of the parallel operations in SNN. Resistive random access memory (ReRAM), a type of memory device that supports non-volatile modification of resistance and exhibits behavior similar to biological synapses [9], is an ideal candidate for such PIM SNN accelerators. ReRAM also has the advantage of high read-and-write speed as well as simple structure [10]. In addition, the non-volatile nature of ReRAM has the benefit of better energy efficiency as no power is need to maintain the information stored in memory.

Resistance of ReRAM is modified by voltage/current signal which changes the conducting micro-structure between two electrodes. With controlled pulse width and amplitude of spike signal, ReRAM devices can be modified to continuous states of resistance [11], enabling analog representation of synapse conductance. Based on ReRAM, PIM design with a simple cell structure, in which only one ReRAM device is needed per synapse, can be achieved. Such design requires much less components compared to conventional CMOS implementation of SNN [12]. For instance, in Indiveri's work the circuit design for SNN with STDP based on CMOS needs about 30 transistor per synapse [13].

Meanwhile, existing designs of ReRAM based SNN accelerators [14] [15] still face challenge caused by the stochastic nature in both the manufacturing process and operation of ReRAM. It is reported in Querlioz's work [5] that device variation of ReRAM significantly degrades learning accuracy for MNIST dataset [16]. The reason is that device variation leads to difficulties in achieving reliable STDP learning behavior as synapse conductance is stored in term of resistance of ReRAM, and each individual ReRAM device shows different resistance modulation characteristics. Beside the negative impact of device variation, noise is another factor that adversely affects learning in real life neural network applications. The event based system of SNN is sensitive to temporal and spatial distortion of spikes, and input data used in the learning process of SNN can contain noisy signal from sources such as the data collection process. In this work we propose a novel STDP learning rule and use it to implement an algorithmic approach to designing a robust ReRAM based PIM accelerator for SNN.

This paper makes three key contributions:

- We propose a frequency-dependent (FD) stochastic STDP algorithm based on learning rules observed in neurophysiological experiments, which can be easily integrated into most STDP based SNNs.
- Compared to deterministic STDP, the proposed design is able to achieve better accuracy when learning under noisy input conditions. The improvement can be observed across different noise types as well as a wide range of noise levels.
- The proposed algorithm shows strong resilience to the impact of device variation in ReRAM based networks, while deterministic STDP suffers from accuracy drop under the same circumstances.

In the following sections, we first discuss the fundamental theory and algorithm of SNN in section II, then present the PIM structure considered in this work in section III. In section IV we demonstrate results from learning the MNIST dataset and compare the performance of networks based on deterministic STDP and FD stochastic STDP.

II. SPIKING NEURAL NETWORK MODEL

A. Spiking Neuron Model

There are different models that are developed to capture the firing pattern of real biological neurons. Long [6] shows that with parameter tuning it is possible to achieve similar spiking frequency in different neuron models, such as Hodgkin-Huxley and integrate-and-fire (LIF), when operating under a specific range of input current. This indicates that for an event based network such as the one presented in this work, which uses spikes rather than membrane potential of neurons to encode information, a mathematically less complex model can be used to achieve same level of performance as the more complex ones. To optimize speed and power consumption of our network design, we choose to use LIF model in this work. The model is described by:

$$dv/dt = a + bv + cI \quad (1)$$

$$v = v_{reset}, \text{ if } v > v_{threshold} \quad (2)$$

In the equation, a , b and c are decided based on the specific network settings. I is the sum of current signal from all synapses that connects to neuron m . I is evaluated by:

$$I_m = \sum_{n=0}^N g_{n,m} v_{pre_n} \quad (3)$$

Here $g_{n,m}$ is the conductance of the synapse connecting neuron n and m . And v_{pre_n} is the voltage signal resulting from spike of neuron n .

B. Synapse Model

In SNN, two neurons connected by a synapse are called pre-synaptic neuron and post-synaptic neuron. When the pre-synaptic neuron spikes, current signal is sent through the synapse to the post-synaptic neuron. Conductance of the

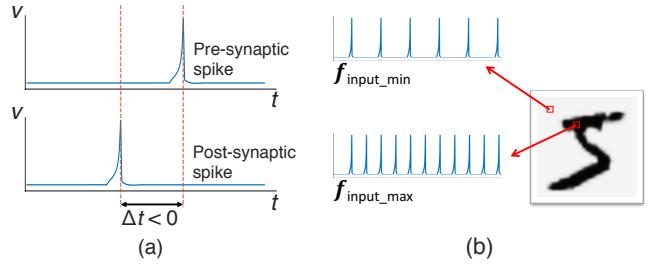


Fig. 1: (a) In STDP learning rule, when pre-synaptic spikes arrive at a synapse after post-synaptic spikes, Δt is smaller than zero. (b) Conversion of input image to spike trains: darker pixels have higher frequency

synapse determines how strongly two neurons are connected and can be considered the connection weight between neurons. Learning is achieved through modulating the conductance following an algorithm developed from the learning rule by Hebb [2] and the temporal synaptic modification rule observed in hippocampus [17]. A more detailed theoretical work by Gerstner explains learning from the perspective of spatial-temporal patterns of spikes [18] and the algorithm is later named spike-timing-dependent-plasticity (STDP). Since then multiple experimental evidence of STDP has been observed [19]–[22], making STDP a biologically plausible algorithm that is suitable for the purpose of this work.

With STDP learning rule integrated, the network is able to extract the causality between spikes of two connected neurons from their temporal relationship. As a result, the SNN can perform conductance update without using label of the input data. More specifically, there are two operations of STDP: long-term potentiation (LTP) and long-term depression (LTD). LTP is triggered when post-synaptic neuron spikes closely after a pre-synaptic neuron spike, indicating a causal relationship between the two events, and the conductance of the synapse is increased. On the other hand, when a post-synaptic neuron spikes before pre-synaptic spike arrives or without receiving a pre-synaptic spike at all, the synapse goes through LTD which decreases its conductance.

We choose to use STDP model presented by Querlioz [5], as it matches experimental measurement of memristive devices [9] [23] and has been tested in neural network application [5]. The model is described by the following equations:

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

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

ΔG_p is the magnitude of LTP actions, and ΔG_d is the magnitude of LTD actions. α_p , α_d , β_p , β_d , G_{max} and G_{min} are parameters that are tuned based on other network configurations such as input matrix size, input spiking frequency and voltage.

C. Stochastic Behavior of Synapses

In stochastic SNN, the potentiation and depression of synapses are non-deterministic, and the probability of the two

actions depends on the temporal relationship between pre-synaptic and post-synaptic spikes. We consider the algorithm presented in [24] by Srinivasan, as probabilities are determined by:

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

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

In the functions above, τ_{dep} and τ_{pot} are time constant parameters. Δt is determined by subtracting the arrival time of the pre-synapse spike from that of the post-synapse spike ($t_{post} - t_{pre}$), as shown in Fig. 1 (a). Probability of potentiation P_{pot} is higher with smaller Δt , which indicates a stronger causal relationship. The probability of depression P_{dep} is higher when Δt is larger. γ_{pot} and γ_{dep} controls the peak value of probabilities.

D. Proposed Stochastic STDP Model

Srinivasan's algorithm captures the exponential dependence on timing of synaptic behavior as observed in biological experiments [3]. However it falls short to resolve the associative potentiation issue in STDP, which is directly related to the theoretical construct of the Hebbian synapse [2]. Associativity in biological synaptic behavior is first reported by Levy [25] and is proved to be an important role in the forming of classical conditioning of nervous systems [26] [27]. As shown in Fig. 2 (a), associativity is a temporal specificity such that when a

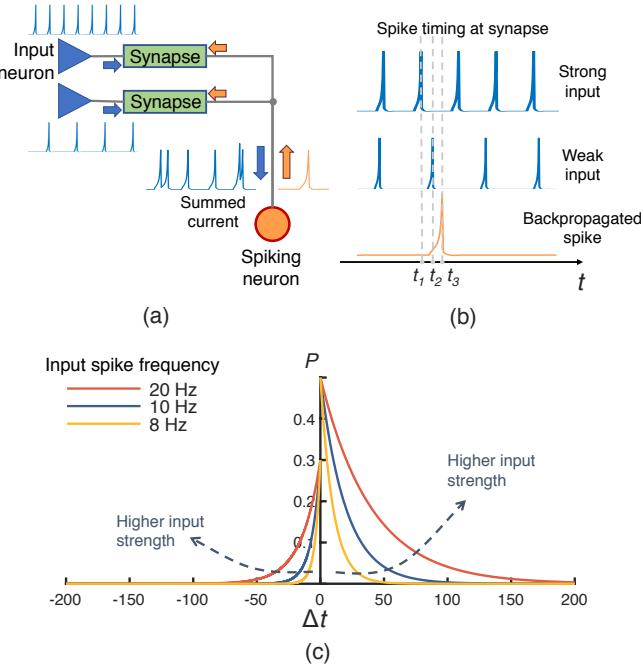


Fig. 2: (a) The associative LTP behavior: convergent of two input (strong and weak) and induction of post-synaptic spike. (b) Spiking timing analysis of associative LTP. (c) Probability curves of the proposed frequency-dependent stochastic STDP with different input spike frequency.

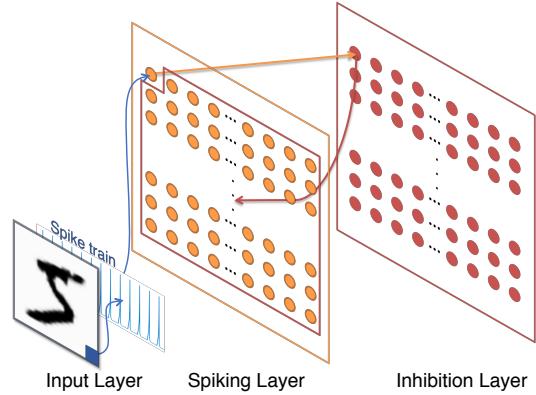


Fig. 3: Network architecture of the SNN implemented in this work.

strong (in case of our SNN model, more frequent) input and a weak (less frequent) input into one neuron induce a post-synaptic spike, a following conductance modulation process is triggered. For timing of spikes received by the synapse t_1 , t_2 and t_3 , as shown in Fig. 2 (b), $\Delta t = t_3 - t_2$ for the weak input is smaller than $\Delta t' = t_3 - t_1$ for the strong input. The occurrence of the post-synaptic spike in this case has more correlation to the strong input, whereas the coincidental timing of spikes from the weak input presents an "illusory" causal relationship. Experimental result from Levy [28] shows that in hippocampus, if the weak input spike arrives before (by as much as 20 ms) or at the same time with the strong input spike, LTP of the synapse transmitting the weak input spike is induced. If the weak input spike arrives after the strong input spike, LTD is induced. In this way the nervous system detects the "illusory" event and react properly.

In the context of STDP based SNNs, associativity can cause erroneous conductance modulation if unaccounted for. Therefore, we propose a frequency-dependent (FD) stochastic STDP that dynamically adjust the width of LTP/LTD window based on input signal frequency. The algorithm is now described by:

$$P_{pot} = \gamma_{pot} e^{(-\Delta t / (\tau_{pot}(1 + \Phi_{pot})))} \quad (8)$$

$$P_{dep} = \gamma_{dep} e^{(\Delta t / (\tau_{dep}(1 + \Phi_{dep})))} \quad (9)$$

$$\Phi_{dep} = \phi_{dep} \frac{f - f_{min}}{f_{max} - f_{min}} \quad (10)$$

$$\Phi_{pot} = \phi_{pot} \frac{f - f_{min}}{f_{max} - f_{min}} \quad (11)$$

Shown in Fig. 2 (c) are probability curves of this algorithm. When input spike originates from a weak input, the probability declines faster than that from a strong input. As a result, spike arriving time for weak input needs to be much closer to the post-synaptic spike to have the same probability of inducing LTP, i.e. the window for LTP is narrower for weak input. The same rule applies to LTD behavior.

E. Network Architecture

In this work we use an SNN architecture as shown in Fig. 3. This architecture is designed for pattern recognition tasks and consists of three layers. First is the input layer. Each neuron in this layer corresponds to one pixel in the input image. During learning process, the 8-bit pixel intensity from the input data is converted into spiking frequency over a range from f_{min} to f_{max} , as shown in Fig. 1 (b), and the relationship is direct proportional. Each input unit keeps track of the corresponding spiking frequency and during learning period t_{learn} of one image, constantly sends excitatory spike signals to the next layer at such frequency. The input layer connects to the spiking neuron layer (second layer) in an all-to-all fashion (fully connected). The inhibition layer (third layer) has the same dimension as the second layer. For a neuron in the third layer at location $< i, j >$, there is only one input connection to it which is from the neuron at the same location, i.e. $< i, j >$, in the second layer, and its output connects to all second layer neurons except for $< i, j >$.

As a result, when one neuron in the second layer spikes, it sends excitatory signal to the one corresponding neuron in the third layer. The inhibitory neuron has low threshold that it activates immediately after receiving one excitatory signal, and then sends inhibitory signal to all other neurons in the second layer for a period of time t_{inh} . Membrane potential of neurons that receive the inhibitory signal is decreased by a value of v_{inh} , and can not spike during t_{inh} . With the inhibition layer implemented, the network achieves a winner-take-all principle throughout the spiking neuron layer, preventing multiple neurons from learning the same pattern. STDP is applied to connections between the first and second layer. The conductance matrix of synapses connected to a single spiking neuron forms a learned template that contains features of one pattern.

III. MODEL OF HARDWARE ARCHITECTURE

A. ReRAM based SNN Accelerator Design

ReRAM is a two terminal device where the resistive switching layer (e.g. HfO_x , NiO , TiO_2 , Al_2O_3 , or their combinations) sandwiched between the top and bottom electrodes as shown in Fig. 4 (c). Device resistance can be modulated by applying set/reset voltage with different directions. In general, ReRAM can be defined as devices that exhibit a voltage-current characteristic [29] as shown in Fig. 4 (a). Pinched at 0 Volt, the curve shows resistance variation of the ReRAM device under different injection current. Specifically, the set process happens when a positive voltage is applied on the top electrode which causes the oxygen ion migration, leaving oxygen vacancy to form a conductive filament (CF). On the other hand, device is reset by applying a reversed voltage which causes the recombination of oxygen ion and vacancy. The formation and rupture of conductive filament determines the device resistance, and thus, different data storage status.

The non-volatile memory and synapse like resistance modulation process of ReRAM make it an ideal candidate for STDP

based SNN circuit. We consider the crossbar structure that has been explored in several ReRAM based SNN hardware implementations [5] [14] [15] to be a good paradigm of designing SNN accelerators. As shown in Fig. 4 (b), in this crossbar array wordlines (WL) and bitlines (BL) are connected by ReRAM at intersections.

B. Mapping SNN into ReRAM Accelerator

Based on the SNN architecture shown in Fig. 3, an ReRAM crossbar PIM accelerator can be constructed accordingly for pattern recognition tasks. In our simulation that uses MNIST dataset, which has a dimension of 28x28, a total of 784 input neurons are connected to WL. While with more spiking neurons the network is able to learn more distinct templates and therefore achieve better classification accuracy, in this work we consider a balanced option between area and performance, and use a second layer with 1000 spiking neuron. It is possible to use ReRAM as binary device by separating the high resistance state and low resistance state. However it would require more devices for each synapse to achieve enough bit width. Here we choose to utilize the continuous resistance states of ReRAM so that only one analog device is need per synapse. With ReRAM placed at each intersection of the crossbar structure, an all-to-all connection is achieved between input neuron and spiking neuron. The result network is shown in Fig. 5 (a). During operation, spike trains (pre-synaptic spikes) from input neurons are sent through the horizontal lines. Signal from each individual synapse on one vertical line is collected and the sum sent to the receiving neuron. If membrane potential of the receiving neuron exceeds its spiking threshold, a post-synaptic spike (backpropagated spike) is sent backwards into the vertical line. The waveform

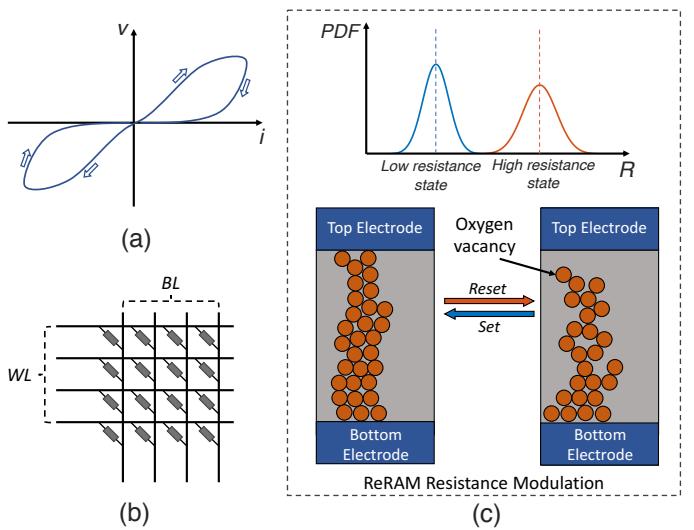


Fig. 4: (a) ReRAM voltage-current characteristic with the "pinched at zero" curve. (b) ReRAM crossbar array. (c) Probability density function (PDF) of ReRAM resistance distribution at low resistance state and high resistance state (top); the change of conductive filaments geometry during resistance modulation of ReRAM (bottom): as gaps forms between oxygen vacancies, resistance of ReRAM increases.

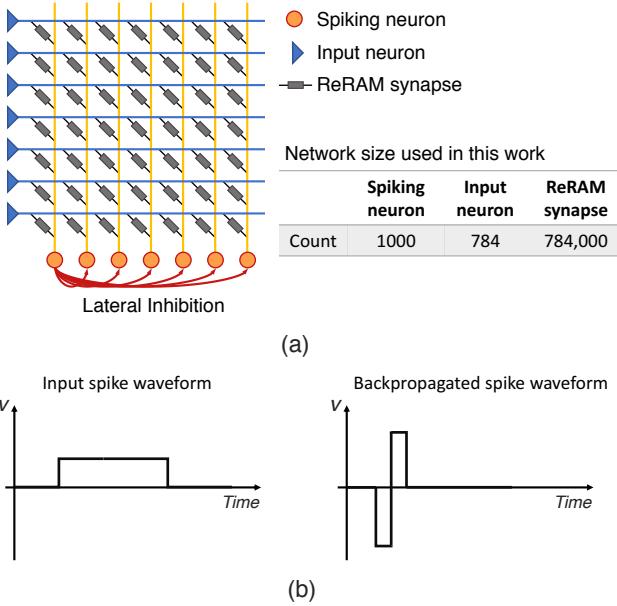


Fig. 5: (a) Crossbar array design of SNN circuit with ReRAM as synapses. (b) Waveform of input spike (signal from input neuron) and backpropagated spike (signal from spiking neuron).

of input spikes and backpropagated spikes used in this design, as demonstrated in Fig. 5 (b), has been shown in [5] to create resistance modulation behavior on ReRAM as described by (4) and (5).

C. Device Variation in ReRAM

Recent study of ReRAM shows that device variations is one of its intrinsic properties [30]. Such variations is a result of the unique resistance modulation operation of ReRAM. In different resistance states of ReRAM distinct conduction mechanisms dominate. At low resistance state, the conduction is similar to metallic conductors as CF constitutes of dense oxygen vacancies. On the other hand, at high resistance state, gaps form between vacancies and electrons move in hopping motion as described by [31]. In case of ultra-high resistance state, connections between CF and electrode are broken and electrons need to overcome large tunneling barrier during conduction.

The resistance modulation process thus changes the fundamental conduction mechanism of an ReRAM device across resistance states, and its randomness originates from a few different sources. First, in order to change device resistance, current impulses are injected through the ReRAM in a process called electric formation [32]. During electric formation oxygen vacancies generation and redistribution happens randomly in the crystal structure, as a result of random walk [33]. Those vacancies constitute CFs that are in stochastic geometry and have different resistance. In addition to that, electric formation also creates different gaps between vacancies that affect electron hopping conductivity. Similarly, it leads to varying tunneling distance between tip of CF and device terminals [34] in ultra-high resistance state. Therefore, a fixed current impulse

can produce different resistance modulation among ReRAM devices and even across operation cycles on the same device. Those variations lead to a statistical distribution of device resistance rather than a specific value. Lin and Li finds in their work that ReRAM device variation can be characterized by a normal or log-normal distribution [35] [36].

Prior works have demonstrated that the intrinsic device variation can be a major concern in terms of computing accuracy [37]. In this paper, we test the proposed FD stochastic STDP algorithm for enhancing the system robustness for device variation. When simulating the SNN, we model such variation by applying parametrized Gaussian noise to the conductance value after each resistance modification process. Specifically, the distribution of an expected G is:

$$X_G \sim N(G, \sigma_{dv}^2 = (\gamma_{dv}G)^2) \quad (12)$$

The distribution is centered at the expected value, with standard deviation proportional to the expected G . And γ_{dv} is a parameter used to control variation level.

IV. RESULTS

A. Simulation Settings

In order to test the SNN performance under noisy input condition, noise is added to the original image data prior to the learning process. As shown in Fig. 6 (a), two types of noise: Additive white Gaussian noise (AWGN) and salt-and-pepper, with different noise levels are tested. AWGN noise is measured in Signal-to-noise ratio (SNR). For input image to spike train conversion, f_{min} is 5 Hz and f_{max} is 22 Hz. For the LIF model V_{th} is -60.2, V_{reset} is -74.7, a is -6.77,

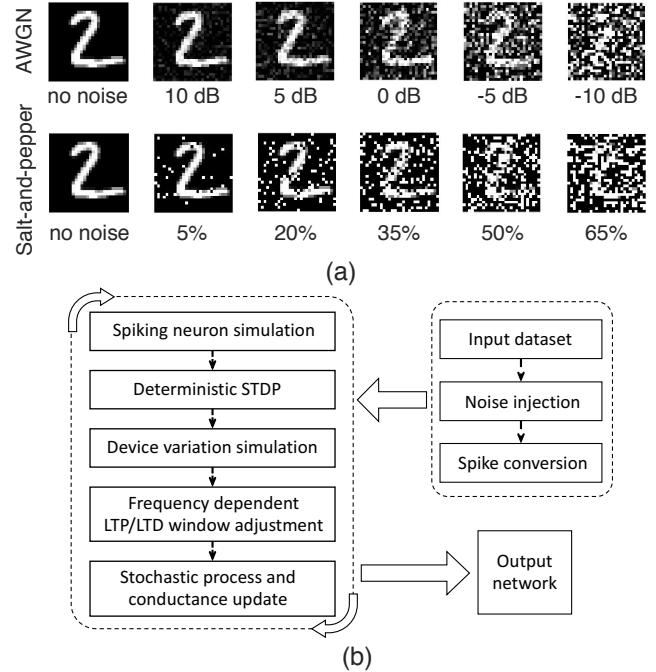


Fig. 6: (a) Examples of different level of AWGN and salt-and-pepper noises used in this work. (b) Program flow of this SNN simulator.

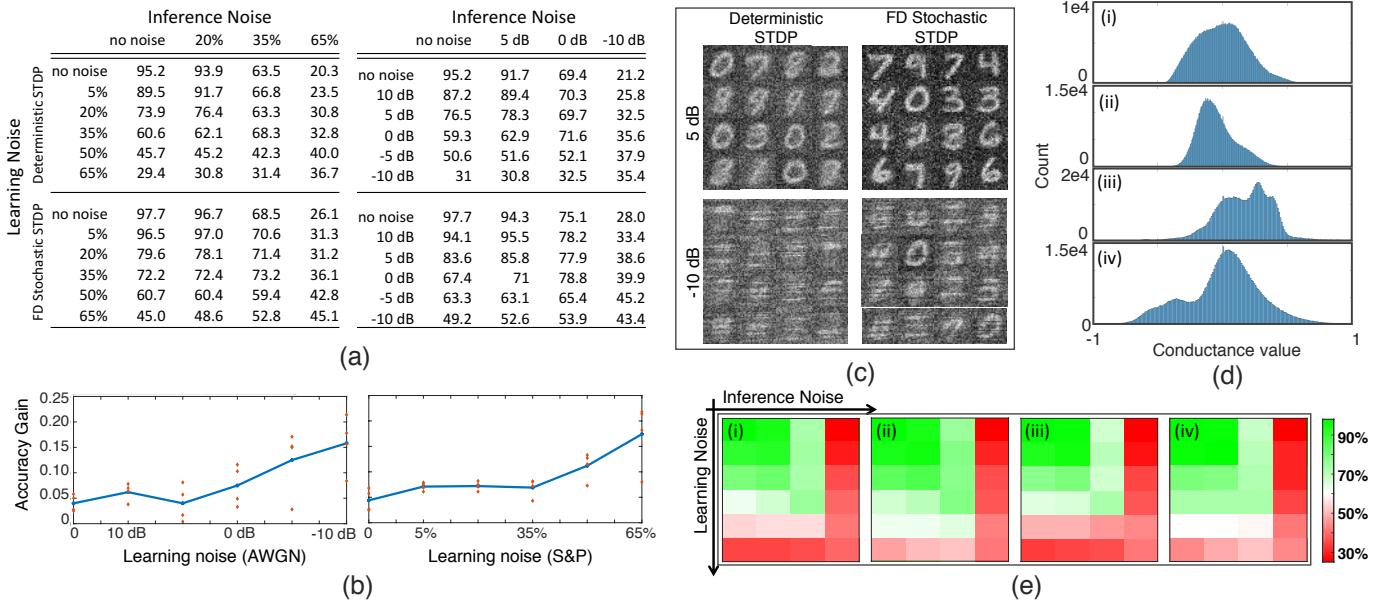


Fig. 7: (a) Accuracy of learning input with AWGN noise (left) and with salt-and-pepper noise (right). (b) Accuracy gain of FD stochastic STDP over deterministic STDP for AWGN noise (left) and salt-and-pepper noise (right); red dots represent results from different inference noise levels and blue line tracks average accuracy gain over all inference noise levels. (c) Visualization of patterns learned from 5 dB and -10 dB input dataset by two network configurations. (d) Distribution of synapse conductance for (i) deterministic STDP learning 5 dB noise input, (ii) FD stochastic STDP learning 5 dB noise input, (iii) deterministic STDP learning -10 dB noise input, and (iv) FD stochastic STDP learning -10 dB noise input. (e) Heat maps of classification accuracy of (i) deterministic STDP learning input with AWGN noise, (ii) FD stochastic STDP learning input with AWGN noise, (iii) deterministic STDP learning input with salt-and-pepper noise and (iv) FD stochastic STDP learning input with salt-and-pepper noise.

b is -0.0989 and c is 0.314. For the base STDP algorithm, α_P is 0.01, β_P is 3, α_D is 0.005, β_D is 3, G_{max} is 1.0 and G_{min} is 0. For FD stochastic STDP, γ_{pot} is 0.3, τ_{pot} is 80, γ_{dep} is 0.2, τ_{dep} is 5, ϕ_{dep} is 0.3 and ϕ_{pot} is 0.1. The MNIST dataset contains 60,000 training images and 10,000 test images. For each learning process all 60,000 training images are shown to the network. The first 1000 images in the test set are used to label all learned patterns and the remaining 9,000 are used to test classification accuracy. In this work we use a GPU accelerated spiking neural network simulator called ParallelSpikeSim [38] running on a desktop machine with Intel Core i7-7700K and NVIDIA GTX 1080 Ti GPU. The program flow is shown in Fig. 6 (b).

B. Learning with Noisy Input Data

For the two types of noise tested, FD stochastic STDP based network shows better performance in almost all test conditions as shown in Fig. 7 (a). Accuracy gain at each learning noise level is shown in Fig. 7 (b) as discrete points and average accuracy gain over all inference noise conditions is shown as a line. It can be observed that accuracy improvement over network based on deterministic STDP is more prominent when learning input is more noisy, e.g. with non-noisy input learning the average of all test cases is 2.9% higher for FD stochastic STDP, while at 0 dB noise the improvement rises to 8.1%. A similar trend exists for salt-and-pepper noise, as average accuracy improvement increases from 2.7% at non-noisy input to 8.6% at 35% noise. It is also worth noting that networks

which learn noisy input dataset show better performance at classifying test images with noise than those with no noise, and maximum accuracy result of a specific inference noise level comes from networks that receives similarly noisy input for learning. This is illustrated in the heatmaps from Fig. 7 (e). Each heatmap shows accuracy in one simulation setting (e.g. FD Stochastic STDP learning input with AWGN noise). The x-axis represents inference noise and y-axis represents test noise; higher accuracy is displayed in green and lower in red. The diagonal positions of each heatmap have higher accuracy than neighbor positions.

Visualization of conductance matrix after learning input with AWGN noise is shown in Fig. 7 (c). At 5 dB input noise both deterministic and FD stochastic STDP are able to learn patterns with distinct features of different digits. Fig. 7 (d) shows the distribution of conductance of all 786,000 synapses in the network, with deterministic STDP and FD stochastic STDP both showing distribution resembling normal distribution. Indeed, when looking at the extreme case where SNR decreases to -10 dB, learning abilities of two network configurations are negatively affected. However, with FD stochastic STDP, the network can extract more features from the noisy input, and achieve around 20% higher classification accuracy. This performance difference can also be reflected in the distribution of conductance in the network as shown in Fig. 7 (d). The distribution of deterministic STDP, which has a wide and flat region, is less ideal than that of FD stochastic

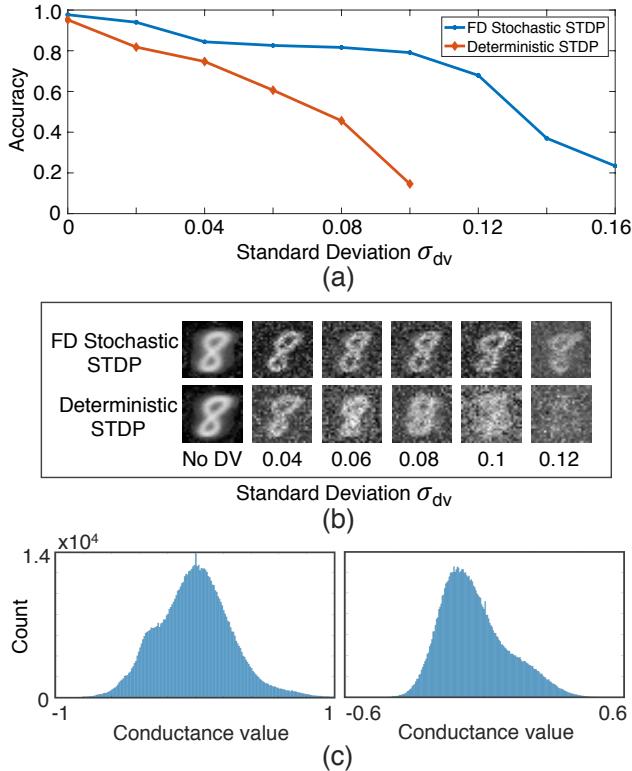


Fig. 8: (a) Accuracy of the network with different standard deviation of device variation. (b) Examples of patterns learned by networks with no device variation (No DV) and different standard deviation of device variation. (c) Distribution of synapse conductance for deterministic STDP (left) and FD stochastic STDP (right) at $\sigma_{dv} = 0.08$.

STDP which is closer to normal distribution.

C. Network with ReRAM Device Variation

In order to understand quantitatively how ReRAM device variation affects learning ability of SNN, we test networks with a range of variations and compare accuracy of the each setting. In this part, no input images noise is used, and device variation during inference has the same distribution as that for learning. Fig. 8 (a) shows change in the result accuracy when ReRAM devices are at different levels of variation. Deterministic STDP and FD stochastic STDP both experience accuracy drop. However, accuracy of deterministic STDP drops around 3 times larger than FD stochastic STDP from no variation to a value of $\sigma_{dv} = 0.02$, and degrade faster with increasing device variation. On the other hand, FD stochastic STDP is more robust to device variation. It has the capability to adapt to a wide range of randomness in the resistance modulation process and achieves above 80% accuracy with up-to $\sigma_{dv} = 0.1$, at which deterministic STDP fails to provide meaningful accuracy.

FD stochastic STDP provides accuracy improvement as it is able to filter out modulation activities induced from spike pairs that have low causality, or from associative synaptic events, therefore statistically keeps the learned pattern from being disrupted by random modulation values. Apparently, such

filtering can maintain a promising level of effectiveness under impact of an extended range of variations until the randomness becomes too high. Fig. 8 (b) shows example patterns learned by networks with different device variation. Patterns from FD stochastic STDP network experience minor degradation from $\sigma_{dv} = 0.04$ to $\sigma_{dv} = 0.12$, while patterns from deterministic STDP network lose their distinct features rapidly as device variation increases. Fig. 8 (c) shows conductance distribution of deterministic STDP (left) and FD stochastic STDP based network with device variation of $\sigma_{dv} = 0.08$.

D. Learning with Device Variation and Noisy Input

In this test, we investigate performance of the proposed algorithm when device variation and input noise are present at the same time. Accuracy is tested with images that have the same type and level of noise as the ones used in learning, and device variation during inference has the same distribution as that for learning. The result is shown in Table I, with each column showing accuracy under certain device variation σ_{dv} . FD stochastic STDP outperforms deterministic STDP in all test conditions, and achieves greater accuracy gain at higher device variation. For both algorithm, compared to when no noise is present in the input, accuracy drops slower with increased device variation when input is noisy. Also, comparing results from one specific device variation above zero, maximum accuracy occurs when input is noisy, instead of when no noise is applied. This behavior shows that when learning noisy input the network experiences increase in the robustness, and smooth out the degradation of accuracy when device variation is present in STDP learning.

V. CONCLUSION

We present FD stochastic STDP as an algorithmic development that serves several purposes. It attempts to addresses the associativity behavior in nervous systems, which has been absent in conventional STDP algorithm used in SNN. It

TABLE I: Accuracy results (%) for learning noisy input (top: deterministic STDP, bottom: FD stochastic STDP) and with device variation from $\sigma_{dv} = 0.04$ to $\sigma_{dv} = 0.14$.

	No DV	0.04	0.08	0.1	0.12	0.14
no noise	95.2	74.7	45.6	14.6	10.3	10.5
10 dB	89.5	79.0	56.0	19.1	13.9	11.7
5 dB	78.3	78.1	62.9	23.9	17.0	12.5
0 dB	71.6	69.6	60.8	25.4	22.7	14.5
5%	92.1	83.4	57.3	19.3	14.0	12.3
20%	76.4	74.2	52.0	22.9	16.7	13.2
35%	68.3	66.5	47.6	15.3	14.8	13.9
	No DV	0.04	0.08	0.1	0.12	0.14
No noise	97.7	84.4	81.6	79.1	67.9	37.0
10 dB	95.9	87.4	83.4	80.4	76.8	43.3
5 dB	85.8	85.3	84.0	81.5	78.3	46.7
0 dB	78.8	76.8	74.6	76.7	77.2	51.6
5%	97.2	88.9	85.6	81.1	74.6	42.5
20%	78.1	77.6	76.7	78.1	75.7	49.3
35%	73.2	71.8	71.3	69.8	68.6	49.5

provides a more biologically plausible learning rule while at the same time achieves better performance. FD stochastic STDP also has the benefit of easy adoption and high efficiency as it uses the frequency information readily available in spike-timing based SNN. We show that FD stochastic STDP based SNN can achieve better accuracy when learning noisy input data compared to network based on deterministic STDP. It also exhibits higher robustness to randomness in conductance modulation resulting from device variation in ReRAM based SNN accelerators, thus provides an solution to designing reliable ReRAM based PIM accelerator for SNN algorithmically.

ACKNOWLEDGMENT

This work is supported in parts by Office of Naval Research Young Investigator Program, National Science Foundation, and Semiconductor Research Corporation nCORE Program.

REFERENCES

- [1] Fahad Javed, Qing He, Lance E. Davidson, et al. Brain and high metabolic rate organ mass: Contributions to resting energy expenditure beyond fat-free mass. *American Journal of Clinical Nutrition*, 2010.
- [2] D O. Hebb, Fred Attneave, and M. B. The Organization of Behavior; A Neuropsychological Theory. *The American Journal of Psychology*, 1950.
- [3] Guo-qiang Bi and Mu-ming Poo. Synaptic Modification by Correlated Activity: Hebb's Postulate Revisited. *Annual Review of Neuroscience*, 2001.
- [4] Peter Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9(August):99, 2015.
- [5] Damien Querlioz, Olivier Bichler, Philippe Dollfus, and Christian Gamrat. Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Transactions on Nanotechnology*, 12(3):288–295, 2013.
- [6] 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.
- [7] Alex Krizhevsky and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, 2012.
- [8] Tara N. Sainath, Abdel Rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. Deep convolutional neural networks for LVCSR. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2013.
- [9] Sung Hyun Jo, Ting Chang, Idongesit Ebong, Bhavitavya B. Bhadviya, Pinaki Mazumder, and Wei Lu. Nanoscale memristor device as synapse in neuromorphic systems. *Nano Letters*, 2010.
- [10] Akifumi Kawahara, Ryotaro Azuma, Yuuichiro Ikeda, et al. An 8 mb multi-layered cross-point ReRAM macro with 443 MB/s write throughput. *IEEE Journal of Solid-State Circuits*, 2013.
- [11] Tsung-Wen Lee and Janice H. Nickel. Memristor Resistance Modulation for Analog Applications. *IEEE Electron Device Letters*, 2012.
- [12] Johannes Fieres, Johannes Schemmel, and Karlheinz Meier. Realizing biological spiking network models in a configurable wafer-scale hardware system. In *Proceedings of the International Joint Conference on Neural Networks*, 2008.
- [13] Giacomo Indiveri, Elisabetta Chicca, and Rodney Douglas. A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Transactions on Neural Networks*, 2006.
- [14] Chenchen Liu, Hao Jiang, Bonan Yan, Chaofei Yang, Linghao Song, Zheng Li, Beipei Liu, Yiran Chen, Hai Li, and Qing Wu. A spiking neuromorphic design with resistive crossbar. In *Proceedings of the 52nd Annual Design Automation Conference on - DAC '15*, 2015.
- [15] Sylvain Saighi, Christian G. Mayr, Teresa Serrano-Gotarredona, et al. Plasticity in memristive devices for spiking neural networks, 2015.
- [16] 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.
- [17] T. V P Bliss and A. R. GardnerMedwin. Longlasting potentiation of synaptic transmission in the dentate area of the unanaesthetized rabbit following stimulation of the perforant path. *The Journal of Physiology*, 1973.
- [18] Wulfram Gerstner, Raphael Ritz, and J. Leo van Hemmen. Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns. *Biological Cybernetics*, 1993.
- [19] Henry Markram, Joachim Lübke, Michael Frotscher, and Bert Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 1997.
- [20] Dominique Debanne, Beat H. Gähwiler, and Scott M. Thompson. Long-term synaptic plasticity between pairs of individual CA3 pyramidal cells in rat hippocampal slice cultures. *Journal of Physiology*, 1998.
- [21] Curtis C. Bell, Victor Z. Han, Yoshiko Sugawara, and Kirsty Grant. Synaptic plasticity in a cerebellum-like structure depends on temporal order. *Nature*, 1997.
- [22] Jeffrey C. Magee and Daniel Johnston. A synaptically controlled, associative signal for Hebbian plasticity in hippocampal neurons. *Science*, 1997.
- [23] Kyungah Seo, Insung Kim, Seungjae Jung, et al. Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device. *Nanotechnology*, 2011.
- [24] Gopalakrishnan Srinivasan, Abhroniil 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.
- [25] William B. Levy and Oswald Steward. Synapses as associative memory elements in the hippocampal formation. *Brain Research*, 1979.
- [26] T J Carew, E T Walters, and E R Kandel. Classical conditioning in a simple withdrawal reflex in Aplysia californica. *J Neurosci*, 1981.
- [27] R. D. Hawkins, T. W. Abrams, T. J. Carew, and E. R. Kandel. A cellular mechanism of classical conditioning in Aplysia: Activity-dependent amplification of presynaptic facilitation. *Science*, 1983.
- [28] W. B. Levy and O. Steward. Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. *Neuroscience*, 8(4):791–797, 1983.
- [29] Leon Chua. If its pinched its a memristor. *Semiconductor Science and Technology*, 29(10):104001, 2014.
- [30] Francis Napolean. Overview of Current Compliance Effect on Reliability of Nano Scaled Metal Oxide Resistive Random Access Memory Device. 2018.
- [31] Nevill F. Mott and E. A. Davis. *Electronic processes in non-crystalline materials*. 1979.
- [32] Zhiqiang Wei and Koji Eriguchi. Analytic Modeling for Nanoscale Resistive Filament Variation in ReRAM with Stochastic Differential Equation. *IEEE Transactions on Electron Devices*, 2017.
- [33] G. R. Grimmett, H. Kesten, and Y. Zhang. Random walk on the infinite cluster of the percolation model. *Probability Theory and Related Fields*, 1993.
- [34] Shimeng Yu, Ximeng Guan, and H. S Philip Wong. On the stochastic nature of resistive switching in metal oxide RRAM: Physical modeling, Monte Carlo simulation, and experimental characterization. In *Technical Digest - International Electron Devices Meeting, IEDM*, 2011.
- [35] Meng-Yao Lin, Hsiang-Yun Cheng, Wei-Ting Lin, et al. DL-RSIM: A Simulation Framework to Enable Reliable ReRAM-based Accelerators for Deep Learning. In *Proceedings of the International Conference on Computer-Aided Design, ICCAD '18*, pages 31:1–31:8, New York, NY, USA, 2018. ACM.
- [36] H Li, Z Jiang, P Huang, Y Wu, H Chen, B Gao, X Y Liu, J F Kang, and H P Wong. Variation-Aware, Reliability-Emphasized Design and Optimization of RRAM Using SPICE Model. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, 2015.
- [37] Yun Long and Saibal Mukhopadhyay. Design of Reliable DNN Accelerator with Un-reliable ReRAM. *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2019.
- [38] Xueyuan She, Yun Long, and Saibal Mukhopadhyay. Fast and Low-Precision Learning in GPU-Accelerated Spiking Neural Network. *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2019.