Advantages of binary stochastic synapses for hardware spiking neural networks with realistic memristors

NEUROMORPHIC
Computing and Engineering

PAPER

Advantages of binary stochastic synapses for hardware spiking neural networks with realistic memristors

Karolis Sulinskas[1,2] and Mattias Borg[1,2,*]

[1] Electrical and Information Technology, Lund University, 221 00 Lund, Sweden
[2] Ericsson Research, Ericsson AB, 223 62 Lund, Sweden
[*] Author to whom any correspondence should be addressed.

E-mail: mattias.borg@eit.lth.se

Keywords: spiking neural networks, binary, stochastic, MNIST, memristor

Supplementary material for this article is available online

OPEN ACCESS

RECEIVED
10 March 2022

REVISED
21 June 2022

ACCEPTED FOR PUBLICATION
28 June 2022

PUBLISHED
18 August 2022

Abstract

Hardware implementing spiking neural networks (SNNs) has the potential to provide transformative gains in energy efficiency and throughput for energy-restricted machine-learning tasks. This is enabled by large arrays of memristive synapse devices that can be realized by various emerging memory technologies. But in practice, the performance of such hardware is limited by non-ideal features of the memristor devices such as nonlinear and asymmetric state updates, limited bit-resolution, limited cycling endurance and device noise. Here we investigate how stochastic switching in binary synapses can provide advantages compared with realistic analog memristors when using unsupervised training of SNNs via spike timing-dependent plasticity. We find that the performance of binary stochastic SNNs is similar to or even better than analog deterministic SNNs when one considers memristors with realistic bit-resolution as well in situations with considerable cycle-to-cycle noise. Furthermore, binary stochastic SNNs require many fewer weight updates to train, leading to superior utilization of the limited endurance in realistic memristive devices.

1. Introduction

Recently, we have seen the emergence of ubiquitous machine learning (ML) which has revolutionized many areas, including efficient manufacturing, logistics, resource usage, personalized healthcare and virtual meeting places. A major challenge in this development, however, is the significant power consumption and time necessary to train artificial neural networks, with recent estimates highlighting that the energy needed to train a single ML model can be comparable with the CO_2 emissions of a car over its entire life span [1]. In addition, ML applications for the Internet of Things requires conservative power consumption in the microwatt to milliwatt regime, drastically lower than the 100 W or higher required for training with current graphics processing units [2]. Memristor-based hardware implementing spiking neural networks (SNNs) is predicted to achieve inference with orders of magnitude better energy efficiency using in-memory computation concepts and online training realized by spike timing-dependent plasticity (STDP) [3]. Early demonstrations of neuromorphic hardware using 3-bit redox reaction-based resistive random access memory (RRAM) have demonstrated energy efficiencies of more than 50 Teraflops W^{-1} [4]. However, current memristor device technology still faces several serious limitations. The potentiation and depression of the synapse weight is in many cases occurring in a nonlinear as well as an asymmetric fashion, which has severe impact on the learning accuracy [5]. Sophisticated peripheral circuitry is required to compensate for this, which can be energy inefficient and difficult to implement in practice. In addition, the precision of the resistive state is typically limited, most often to 3–4 bits per memristor [4], and in the best case 6–8 bits [6], which limits the smallest deterministic weight update steps that can be realized. Concepts in which the contributions of several memristors are added up to give sufficient weight bit depth are possible, but this has a negative impact on energy and area efficiency. Finally, the number of write cycles that analog memristive devices can endure before failure typically lies in the

range of 10^6–10^9 cycles [7]. This precludes the use of the technology in cases that require many weight updates, such as training of convolutional neural networks [8], and puts strict demands on optimized utilization of the limited available cycles before device failure.

A method for online learning using binary memristive devices has been proposed which utilizes the fact that switching between binary resistive states can be a stochastic process in some memristor types, such as RRAM and spin-transfer torque magnetic random access memory (STT-MRAM) [9, 10]. Recently, ultra-scaled ferroelectric field-effect transistors have also been shown to exhibit such stochastic switching [11]. We here choose to call this method 1-bit stochastic STDP (1s-STDP for brevity), and SNNs containing 1-bit stochastic weights are called 1s-SNNs. In this method, STDP-like learning was achieved in which the probability for changing weight between two binary states follows the same function as for regular STDP. Querlioz *et al* demonstrated that this learning method is very robust against device-to-device variations as well as variations in switching probability [12]. In this work, we investigate how stochastic learning in 1-bit memristors can give performance advantages compared with deterministic learning in realistic analog memristors, providing improved utilization of the often-limited cycling endurance of memristive devices and an improved robustness to cycle-to-cycle noise.

2. Methods

We base our investigation on the task of classifying images of handwritten digits from the standard MNIST dataset [13], in line with earlier reports. Using the package *Brian 2* [14], we implement a two-layer SNN consisting of a 28 × 28 neuron ($N_{\text{input}} = 784$) input layer that has an all-to-all synaptic connection to an output layer consisting of 1024 leaky integrate and fire neurons (figure 1(a)). The neurons in the output layer are modelled by the following differential equation:

$$\frac{\mathrm{d}\nu_i}{\mathrm{d}t} = -\frac{\nu_i}{\tau_\nu} + \sum_{j=1}^{N_{\text{input}}} \Delta V_j(t)\frac{w_{ij}}{w_{\max}} \tag{1}$$

in which ν_i is the membrane potential for neuron i and the leakage time constant $\tau_\nu = 20$ ms. $\Delta V_j(t)$ is defined as the change of potential upon an incident spike from a neuron j in the case when the synaptic weight w_{ij} is in its high state $w_{\max}$. In a hardware system $\Delta V_j(t)$ is a function of the spike amplitude and duration as well as the capacitance of the neuron membrane. For simplicity, here we use $\Delta V_j(t) = 1$ (mV) $* \sum_k \delta(t - t_{j,k})$, where $t_{j,k}$ is the time of spike k originating from neuron j in the input layer. A neuron fires when ν_i reaches above a fixed threshold $\nu_i = 50$ mV, after which ν_i is reset to zero. We also implement winner-takes-all lateral inhibition in the output layer to promote neuron specialization [15] by resetting ν in all neurons of the output layer if one fires. The neurons of the input layer are Poisson sources that map to each pixel in the presented image, and which give out spikes with an average frequency between 0 and 50 Hz, corresponding to the pixel brightness level (values 0 to 255). In this case white pixels in the image are represented by the highest average frequency (50 Hz) and black the lowest (0 Hz). The weighted synaptic connections between the two layers can only have two conductance states, $w_{\min} = 10$ nS and $w_{\max} = 100$ nS, and initially the state is set at random in either of the two states. During training, images from the MNIST training data set are presented to the network for 250 ms/image (simulation time). A 150 ms duration without input stimuli was used in between images to allow the network to settle back to a baseline state. Training of the synaptic weights was performed using a 1s-STDP protocol, visualized in figure 1(b), implemented such that a spike event from the connected neuron in the input layer at time t_{pre} occurring within a defined time window $t_{\text{post}} - t_{\text{pre}} < T_{\text{pot}}$ before the connected neuron in the output layer spikes at time t_{post} leads to a possible change of weight from $w_{\min}$ to $w_{\max}$ (i.e. a digital potentiation). The probability for potentiation to occur is set by a fixed probability in our simulations, p_{pot}. In a practical device p_{pot} can be controlled by the physical programming pulse duration and/or amplitude depending on the type of stochastic memristor device [9, 11, 12]. For $t_{\text{post}} - t_{\text{pre}} > T_{\text{pot}}$ as well as $t_{\text{post}} < t_{\text{pre}}$ instead there is a finite possibility of changing the weight from $w_{\max}$ to $w_{\min}$ (i.e. a digital depression), decided by a corresponding depression probability p_{dep}. As an additional option one can introduce a 'dead zone' in the range $T_{\text{pot}} < t_{\text{post}} - t_{\text{pre}} \leqslant T_{\text{pot}} + T_{\text{dead}}$ in which both probabilities are zero [16].

In this paper, we first explore the effect of varying p_{pot}, p_{dep}, T_{pot} and T_{dead} on the classification performance of the SNN. Before testing the classification performance, the output neurons are labelled according to the type of images (digits) that they reacted (spiked) the most to, following the procedure of Diehl and Cook [17]. This is done by presenting 10 000 random images from the training data set and recording the output spikes. After each presented image the neuron that spiked the most obtains a score of 1 added to the presented image label. The other neurons obtain a score equal to the fraction of the spikes when compared with the strongest spiking neuron. Thus, after labelling each neuron has obtained a certain score for each of the ten labels. The testing is then made by presenting the 10 000 images from the test data set to the network, keeping synapse weights fixed,

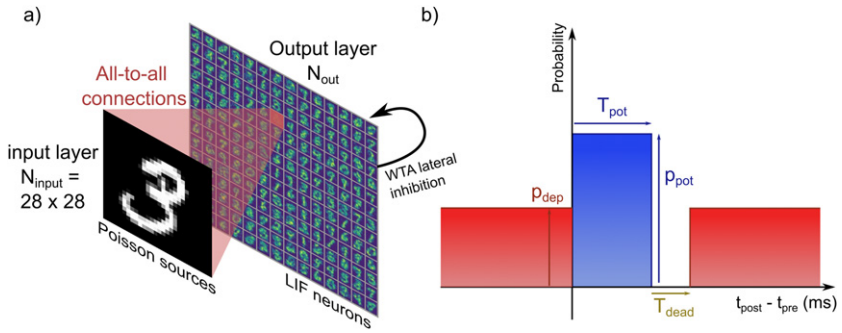

Figure 1. Illustration of (a) the SNN network topology and (b) the parameters of the 1s-STDP training model used in this work.

and collecting the spike count from the output neurons. The neuron scores are multiplied by the number of spikes that neurons produced for a given test image. Then all the scores are summed up and the number label with the highest score is the guess that the network makes. If fewer than five output spikes were registered for an image during labelling or classification the maximum spiking frequency of the input neurons was temporarily increased by 25 Hz and the same image is presented again. This was done to ensure that each neuron can be labelled, or the image classified.

Typically, training is done for one epoch if not explicitly stated otherwise. The run-to-run variation in classification accuracy introduced by the random initialization, choice of images for labelling, etc was estimated by training an identical network 21 times. A standard deviation of 1.60% was obtained here and should be viewed as an estimated error bar for the data points in this paper.

Finally, we compare the 1s-SNNs with deterministic SNNs with the same topology consisting of analog synapses. Deterministic STDP has been realized in experimental memristor devices by several groups [18, 19] via the temporal overlap of specific voltage pulse shapes from both the pre- and post-synaptic neuron [20]. The analog synapses are implemented as ideal memristors with infinite dynamic range ($w_{\min} = 0$ and $w_{\max} = 100$ nS), and with the same type of simplified STDP function as for the 1s-SNN networks, with the difference that potentiation and depression is deterministic and made incrementally such that

$$\Delta w = \pm \mu_0 * (w_{\max} - w_{\min}), \tag{2}$$

with the learning rate μ_0 corresponding to a certain fraction of the dynamic range, which we vary to correspond to various levels of bit depth. Note that we do not consider nonlinearity and asymmetry in weight updates, which is well known to severely affect accuracy for memristors [5]. Even an asymmetry as low as 5% will have an impact on the accuracy [21]. It is thus important to keep in mind that the analog deterministic SNNs here constitute a best-case scenario for a given bit depth.

3. Results and discussion

3.1. Parameter optimization

We begin by investigating the impact of p_{pot} and p_{dep} and their ratio on the classification accuracy of the 1s-SNNs. Here we use $T_{\text{pot}} = 20$ ms and keep $p_{\text{dep}} = 0.1$, while varying p_{pot}, with the classification results presented in figure 2(a). We observe an increase in learning accuracy as p_{pot} dominates over p_{dep} with a maximum for our network for $p_{\text{pot}} = 0.2$, resulting in $p_{\text{pot}}/p_{\text{dep}} = 2$. We find that when this ratio is too low the synapses tend to be overly depressed, while in the other case the tendency is for the network to favour specific numbers so that a large fraction of neurons specialize on, for example, the number '1'. If the ratio is very high (50 or so) the network tends towards having all synapses maximized. It is likely that finding a good balance between potentiation and depression is important for 1s-STDP and will depend on the average frequency level of the input as well as the neuronal thresholds and needs to be optimized depending on the implementations of the network.

While keeping $p_{\text{pot}}/p_{\text{dep}}$ constant, we now investigate the impact of the absolute level of p_{pot}. From figure 2(b) it is clear that the classification accuracy is remarkably insensitive to the absolute value of p_{pot} in the broad range investigated between 0.001 to 0.2, with $T_{\text{pot}} = 20$ ms and $p_{\text{pot}}/p_{\text{dep}} = 2$, and in all cases ends up in the range 70%–75% after one epoch of training. With a larger $T_{\text{pot}} = 50$ ms and $p_{\text{pot}}/p_{\text{dep}} = 1$ even higher results can be achieved, and only a weak maximum may be observed near $p_{\text{pot}} = 0.01$, with a classification accuracy around 80%. The insensitivity to the absolute value of p_{pot} can be understood because a synapse in principle only needs to change state once to acquire its optimal state, and if the switching probabilities are

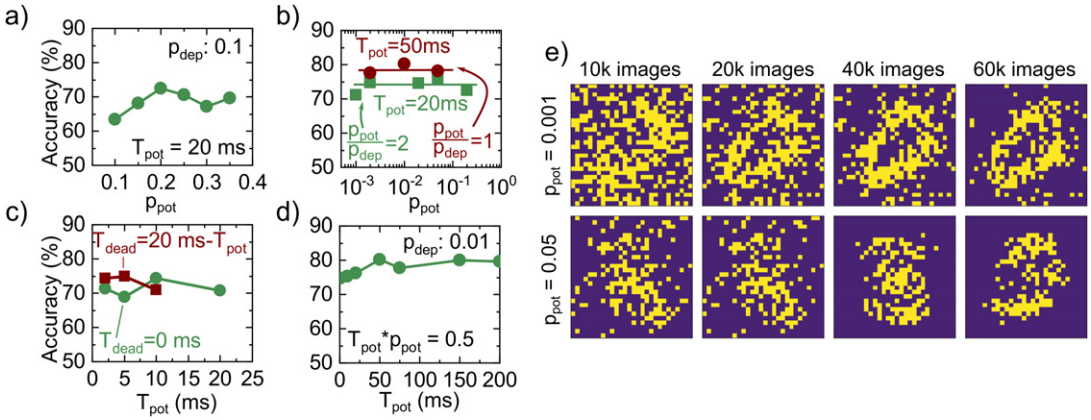

Figure 2. Influence of 1s-STDP parameters on classification accuracy (a)–(d). If not otherwise specified $T_{\text{dead}} = 0$ in these graphs. (e) Evolution of the conductivity map during training for a representative neuron in the output layer for the cases of $p_{\text{pot}} = 0.001$ and 0.05.

low then it will on average require stimuli from more images before switching happens and it is less likely that occasional stimuli away from the ideal synaptic weight distribution will affect it. Thus, we observe a slow but consistent development of the weight distribution towards its final trained state, as exemplified in figure 2(e). For high probabilities switching may occur more frequently and one might expect that the synaptic population would specialize on a subset of images. However, the weights are also more likely to switch back again, leading to a more dynamically changing weight distribution, which appears to converge to a final state with a similar learning outcome as with lower probabilities.

We continue by varying T_{pot} between 2 and 20 ms while keeping $p_{\text{pot}} = 0.1$ and $p_{\text{pot}}/p_{\text{dep}} = 10$. We also evaluate the effect of a 'dead zone', T_{dead}, following T_{pot} in which there is a zero probability for weight change. We consider a total constant window size of 20 ms, such that $T_{\text{pot}} + T_{\text{dead}} = 20$ ms. Varying T_{pot} without a dead zone produces no distinguishable trend, given our estimated run-to-run variation of $\pm 1.60\%$ (figure 2(c)), and it is difficult to draw conclusions on its impact for this range of parameters. With an added dead zone there is possibly some improvement when T_{pot} is small, similar to what was previously observed by Srinivasan and Roy for a convolutional SNN based on 1s-STDP [16] and explained there by having less depression of moderately correlated features.

Finally, we investigate the effect of the total magnitude of potentiation relative to depression by keeping the product of T_{pot} and p_{pot} constant at 0.5 (ms). The results are presented in figure 2(d) as a function of T_{pot}, which is varied between 2 and 200 ms. A slight improvement in accuracy is observed for larger T_{pot}, up to 50 ms, achieving an accuracy as high as 80%. Increasing T_{pot} beyond 50 ms gives no further measurable improvement, which can be understood from the weak temporal correlation between pulses at such large values of $t_{\text{post}} - t_{\text{pre}}$.

3.2. Comparison with analog deterministic synapses

Having obtained a good understanding of the parameters that affect classification accuracy for 1s-SNNs, we now compare the performance and robustness of these networks with SNNs with regular analog deterministic synapses. For this comparison we choose a 1s-STDP network with $T_{\text{pot}} = 20$ ms, $p_{\text{pot}}/p_{\text{dep}} = 2$ and $T_{\text{dead}} = 0$ ms. We begin by a comparison with ideal analog deterministic synapses with varying μ_0, emulating different bit-level precision. We assume that the weight can change in a linear fashion without any noise. Nonlinear and asymmetric behaviour is often observed in practice but can be linearized by various pulse duration or amplitude variation schemes [22]. Lithium ion-based memristors can have a high degree of linearity but with limited dynamic range [23]. The analog deterministic networks used here should thus be considered as a best-case scenario from this point of view. The classification accuracy of analog deterministic SNNs and 1s-SNNs are compared in figure 3. As discussed in conjunction with figure 2(b) the performance of the 1s-SNN is insensitive to p_{pot} and achieves very similar accuracy over five epochs of training regardless of the value of p_{pot}, with an average classification accuracy of 74% (figure 3(a)). The analog deterministic SNNs with small learning rates (figure 3(b)) perform consistently better, reaching an average classification accuracy of 83% for 6-bit and better ($\mu_0 \leqslant 0.016$) precision. Although the simplicity of the SNN topology and labelling method used here results in rather unimpressive absolute results compared with ANNs trained by stochastic gradient descent and backpropagation, what is important here is the comparison between 1s-SNNs and analog deterministic SNNs. Better absolute results can be obtained by a more powerful classification method, for example Yousefzadeh *et al* stored the normalized spike counts of the output layer as frames used to train a fully connected SoftMax

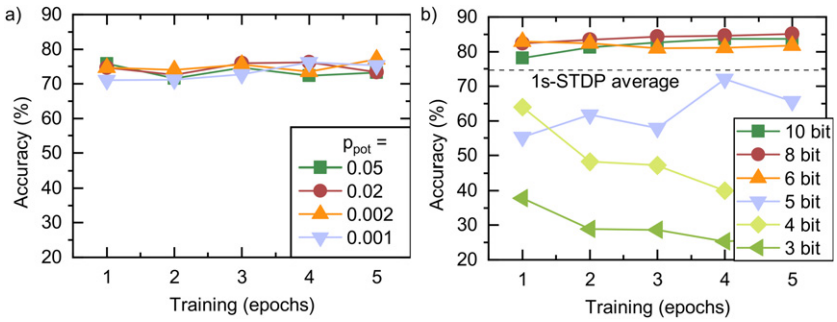

Figure 3. Classification accuracy as a function of the number of epochs trained for (a) 1s-STDP networks with various p_{pot} levels and (b) SNNs trained by deterministic STDP and linear memristors with varying bit precision.

classifier by stochastic gradient descent, improving classification accuracy on MNIST, using the same SNN topology and network size as in our work, from 75.6% to around 94% [24].

That the 1s-SNN underperforms compared with a high-precision analog deterministic SNN is in line with the results of Querlioz *et al*, who observed that for image classification with the same network topology as here 1s-STDP requires a redundancy in terms of the number of output neurons of at least ten times to achieve the same performance as SNNs with ideal analog deterministic synapses, while for other tasks the required redundancy is much lower [12]. Importantly, the smallest conductance change that is reliably achievable in experimental memristor devices is seldom high enough to represent 6 bits or more. More often 2–4 bit precision can be achieved [25, 26]. For learning rates corresponding to bit precisions below 6 bits, the classification accuracy of the deterministic SNNs suffers. Already at 5 bits, the classification accuracy is unstable and dips below the average for 1s-SNNs, in line with other reports [5]. For 3- and 4-bit precision the classification accuracy is as low as 25% and 40% after four epochs, clearly not converging on a reasonable solution. These results highlight that compared with realistic, although still linear, analog deterministic memristors the 1s-STDP approach has considerable performance benefits. In addition, the insensitivity of 1s-STDP to the p_{pot} level means that the absolute switching probabilities do not have to be precisely controlled, further benefitting the realization of the approach in actual hardware.

3.3. Utilization of limited endurance

The finite cycling endurance of memristive devices is an important limiting factor for practical implementation of memristive synapses in real-world neuromorphic systems [3]. Despite the fact that RRAM and phase change memory as well as ferroelectric RAM have demonstrated endurance past 10^9 state switching cycles [6] most report endurance values in the range of 10^5–10^8 cycles [27, 28]. These values are usually presented for full range switching between the two extreme resistive states, while one might expect gradual weight updates such as during analog deterministic learning in a SNN to be more beneficial for the lifetime of the device. Even so, it is worth noting that in contrast to analog deterministic SNNs in which the synapses are constantly exposed to small weight updates, a binary stochastic synapse is only updated a fraction of the time for low switching probabilities ($p_{pot} < 0.01$). In addition, the fact that these synapses only have two distinct states means that attempts to switch are only required if the synapse is not already in the target state. For example, if the relative timing of the input and output spikes end up in the potentiation time window but the synapse is already in the w_{max} state there is no need to expose it to a potentiation pulse. This could lead to a considerably better utilization of the device switching cycles in a real-world application. In figure 4, a histogram of the number of state switching operations performed per synapse is presented for a 1s-SNN with $p_{pot} = 0.01$, $p_{pot}/p_{dep} = 2$ and $T_{pot} = 20$ ms and an analog deterministic SNN with $\mu_0 = 0.032$, corresponding to 5-bit precision. For the 1s-SNN most synapses flip state fewer than ten times for a full epoch of training, with the most active synapse only changing state 28 times. For the analog deterministic SNN the average number of weight updates is around 200, and the most active synapse updates it is 6706 times, a difference of 240 times. For smaller μ_0 the number of weight updates grows even more. An apparent benefit of 1s-SNNs is thus that one can much better utilize the limited number of switching cycles before device failure. Still, one may argue that even attempted weight updates that did not lead to a weight flip can still degrade a device. Therefore, we also compare the number of attempted weight flips, which as expected is much greater than actual flips, with an average number of attempts being 109 and most active synapse attempting to flip state 1943 times (figure 4). However, the number is still more than three times lower than the same number for an analog deterministic SNN.

Figure 5 illustrates the spatial arrangement of synapse weight update activity. Here each pixel in a map corresponds to a single synaptic connection, while the full map represents all the connections to a single output

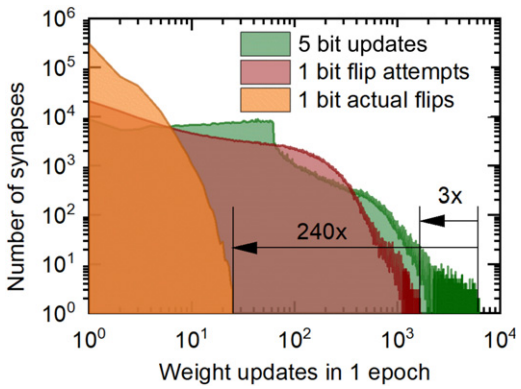

Figure 4. Number of (actual and attempted) weight updates for an 1s-SNN compared with weight updates for an analog deterministic SNN.

neuron. In figure 5(a), the binary stochastic synapses overlapping most with the emerging pattern of high conductance connections are also the synapses that are most actively flipping state. A similar behaviour is also seen for analog deterministic SNNs, shown in figure 5(b), where the highest number of weight updates also closely match the conductance map. However, an additional feature here is that the edge of the pattern has more weight updates than the centre regions, a feature that is not as clear for weight flipping in the 1s-SNN. Such behaviour is indicative of a well-behaving network in which the connectivity map updates gradually to match to new changing input. Strikingly, this pattern is instead clearly observed in the centre row maps in figure 5(a), which show the number of attempted flips. This indicates that the probabilistic features of 1s-STDP learning can approximate a similar behaviour to deterministic STDP with analog weights, as was predicted by Querlioz *et al* [12].

3.4. Robustness to cycle-to-cycle noise

The ability of a neuromorphic system to give reliable output despite the presence of non-idealities such as noise is important for its robustness in a real-world application. Noise can be categorized into device-to-device noise and cycle-to-cycle noise, as well as input noise. Device-to-device noise occurs due to slight variations in geometry or defect density between devices, leading to slightly different switching voltages or conductance levels for example. SNNs trained by STDP are typically quite good at handling this kind of noise as the training method compensates for such intrinsic variation [5]. The same is true for input noise, which has even proven beneficial to generalization of pattern learning [29]. Here we investigate and compare the robustness of 1s-SNNs and analog deterministic SNNs to cycle-to-cycle noise, which corresponds to noise in the weight update probability (1s-SNN) or magnitude (analog deterministic SNN) within the same device.

Cycle-to-cycle noise is inherent to realistic memristor devices, in which synaptic weight updates are induced by exposing the memristor to a certain voltage or current impulse. The precise amplitude and duration of this impulse has direct influence on the resulting new weight state, and thus the weight updates are subject to random fluctuations of voltage or current, leading to noise. In some device types such as RRAM the conductive filament may grow/shrink at slightly different rates with every voltage pulse, depending on microscopic differences in filament geometry from cycle to cycle, leading to additional cycle-to-cycle noise.

For the analog deterministic STDP training we implement noise as an additional random contribution μ_r to the learning rate of the weight change, leading to

$$\Delta w = \pm(\mu_0 + \mu_r r)(w_{\max} - w_{\min}) \tag{3}$$

with the magnitude of μ_r corresponding to the degree of noise in the weight update and r being a uniformly distributed random variable in the range $[-1, 1]$. For $\mu_r \geqslant 1$ the noise is large enough that a weight potentiation may instead lead to a slight depression, or the other way around. The weight is still limited to the range $[w_{\min}, w_{\max}]$.

For a binary stochastic synapse, the weight change is decided by p_{pot} and p_{dep} and cycle-to-cycle noise can thus be modelled as a variation to these probabilities. Here we apply a physically relevant noise model for ferroelectric devices [11] in which the switching probability (for $p \ll 1$) depends exponentially on the write voltage pulse duration via

$$p(t_r) = \mathrm{e}^{\alpha(t_0 + t_r r)} - 1 \tag{4}$$

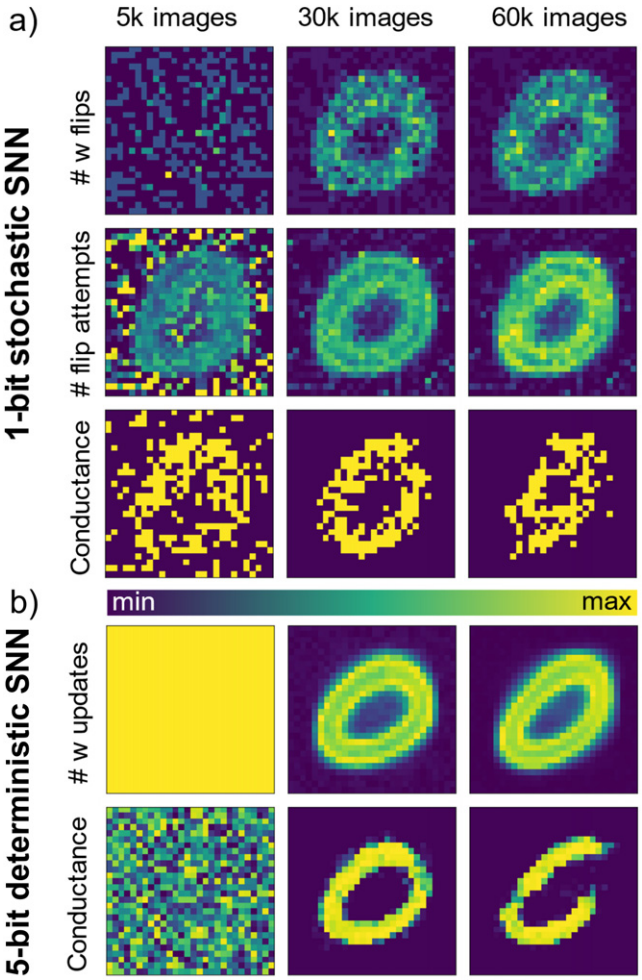

Figure 5. Single output neuron connection map, in which each pixel corresponds to one synapse. (a) Maps for the number of weight flips, attempts to flip weight and conductance for 1s-SNN. (b) Maps for the number of weight updates and conductance for a 5-bit deterministic SNN. The colour scale shown in the centre is a relative scale within each map. The three columns correspond to the maps after training with 5 k, 30 k and 60 k images.

where $\alpha = 10^3$ s^{-1}, t_0 is the time that results in nominal probability, r is a uniformly distributed random variable in the range $[-1, 1]$ and t_r (s) defines the magnitude of the noise in $p(t_r)$ as t_r/t_0. A similar relationship would also apply to other stochastic devices such as STT-MRAM [10].

Figure 6(a) displays the impact of increased cycle-to-cycle noise amplitude on the classification accuracy for a 5-bit deterministic SNN compared with a 1s-SNN with $p_{\text{pot}} = 0.01$, $p_{\text{pot}}/p_{\text{dep}} = 2$, and $T_{\text{pot}} = 20$. Interestingly, here we observed that although the performance of both network types is quite robust to added noise, the 5-bit SNN is degrading by about 5% for $\mu_r > 1.2$ while the 1s-SNN remains unaffected, resulting in converging performance for the two network types at high cycle-to-cycle noise levels. This is an interesting finding given that μ_0 fluctuation of this magnitude could be expected in realistic memristor devices.

The effect of cycle-to-cycle noise on 1s-SNNs compared with analog deterministic SNNs can be understood by variance analysis. Using equation (4) the variance of p_{pot} for 1s-SNNs can be expressed as

$$\text{Var}\left(p_{\text{pot}}\right) \approx \mathrm{e}^{2\alpha t_0}\frac{\alpha^2}{3}t_r^2 \tag{5}$$

which clearly scales with the square of the noise magnitude. A corresponding expression is also valid for p_{dep}. As was observed in figure 2(b), the classification accuracy is insensitive to the absolute value of p_{pot} and p_{dep} as long as the ratio is approximately constant. The fact that added cycle-to-cycle noise only has an indirect effect on the weight values through the switching probabilities can thus explain the negligible impact on the classification performance.

In contrast, cycle-to-cycle noise gives a direct effect on the variance of the weights in analog deterministic SNNs, and using equation (3) can be expressed as

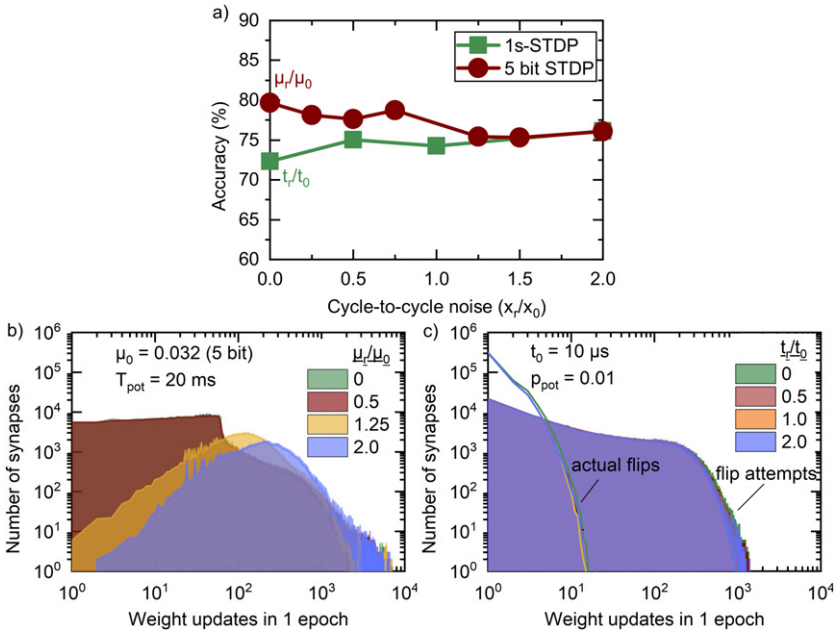

Figure 6. (a) Accuracy of 1s-SNN and 5-bit deterministic SNN as function of increased cycle-to-cycle noise amplitude. Effect of cycle-to-cycle noise on the number of weight updates during one epoch for (b) a 5-bit deterministic SNN and (c) for a 1s-SNN including attempted flips.

$$\mathrm{Var}(w) = (w_{\max} - w_{\min})^2 \mu_0^2 \left(1 + \frac{1}{3}\left(\frac{\mu_r}{\mu_0}\right)^2\right), \tag{6}$$

which scales as the square of the noise magnitude. This direct effect of the noise on the weight values can explain the somewhat reduced classification accuracy at high noise levels. The derivations of equations (5) and (6) are added to the supplementary information (https://stacks.iop.org/NCE/2/034008/mmedia) for reference.

As cycle-to-cycle noise may affect how efficiently a synapse achieves its optimal state we also again investigate the distribution of the number of weight updates in the 5-bit deterministic SNN. Figure 6(b) reveals that the distribution is becoming skewed towards higher number of weight updates as μ_r increases, clearly indicating that the network requires many more weight adjustments when noise level goes beyond $\mu_r/\mu_0 = 1$, and the weight update is less deterministic. In contrast, the corresponding graph for the 1s-SNN in figure 6(c) surprisingly shows no significant difference when adding noise. This can be understood considering that at a given low nominal p_{pot} (and p_{dep}), added noise of the same magnitude may only lead to very few additional weight flips or attempts. This is an effect intrinsic to the binary properties of the weight representation, allowing a weight which was wrongly set to correct itself in just a single weight update. Thus 1s-SNNs appears very attractive for optimizing the lifetime of devices with limited endurance, especially when compared with situations for which one should expect considerable cycle-to-cycle noise.

4. Conclusion

The viability of implementing SNNs in hardware using memristive synaptic devices may be limited by many non-ideal properties of memristors in practice [3]. In this paper we have evaluated SNNs using binary stochastic weights as an alternative to analog deterministic weights. We find that performance of 1s-SNNs can match analog SNNs if their bit depth is less than 5 bits. We also observe that 1s-SNNs require significantly fewer weight updates to reach a trained state, leading to much better utilization of the limited memristor endurance. With added cycle-to-cycle noise, the analog deterministic SNNs require even more weight updates to converge in training by STDP, while 1s-SNNs are unaffected. Cycle-to-cycle noise also leads to a somewhat reduced performance for analog deterministic SNNs, while the performance of 1s-SNNs is unaffected. All in all, the results of this work point to 1s-SNNs as a promising alternative solution for robust ML in hardware that avoids many of the issues related to non-ideal characteristics of currently available analog memristive devices.

Acknowledgments

This work was supported by the Swedish Foundation for Strategic Research (SSF) project no. SM21-0008, and the Swedish Research Counsel (VR) project no. 2018-05379. The authors acknowledge the helpful input of Dr Saeed Bastani.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

ORCID iDs

Karolis Sulinskas https://orcid.org/0000-0003-2606-6414
Mattias Borg https://orcid.org/0000-0003-1217-369X

References

[1] Strubell E, Ganesh A and McCallum A 2019 Energy and policy considerations for deep learning in NLP *Proc. 57th Annual Meeting of the Association for Computational Linguistics* (Florence, Italy) pp 3645–50
[2] Reuther A, Michaleas P, Jones M, Gadepally V, Samsi S and Kepner J 2019 Survey and benchmarking of machine learning accelerators *2019 IEEE High Performance Extreme Computing Conf. (HPEC)* (Waltham, MA, USA) pp 1–9
[3] Musisi-Nkambwe M, Afshari S, Barnaby H, Kozicki M and Sanchez Esqueda I 2021 The viability of analog-based accelerators for neuromorphic computing: a survey *Neuromorph. Comput. Eng.* **1** 012001
[4] Xue C-X *et al* 2019 A 1 Mb multibit ReRAM computing-in-memory macro with 14.6 ns parallel MAC computing time for CNN based AI edge processors *2019 IEEE International Solid-State Circuits Conf. (ISSCC)* (San Francisco, CA, USA) pp 388–90
[5] Islam R *et al* 2019 Device and materials requirements for neuromorphic computing *J. Phys. D: Appl. Phys.* **52** 113001
[6] Cao Q, Lü W, Wang X R, Guan X, Wang L, Yan S, Wu T and Wang X 2020 Nonvolatile multistates memories for high-density data storage *ACS Appl. Mater. Interfaces* **12** 42449–71
[7] Saxena V 2021 Neuromorphic computing: from devices to integrated circuits *J. Vac. Sci. Technol.* B **39** 010801
[8] Tsai H, Ambrogio S, Narayanan P, Shelby R M and Burr G W 2018 Recent progress in analog memory-based accelerators for deep learning *J. Phys. D: Appl. Phys.* **51** 283001
[9] Suri M, Querlioz D, Bichler O, Palma G, Vianello E, Vuillaume D, Gamrat C and DeSalvo B 2013 Bio-inspired stochastic computing using binary CBRAM synapses *IEEE Trans. Electron Devices* **60** 2402–9
[10] Vincent A F *et al* 2015 Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems *IEEE Trans. Biomed. Circuits Syst.* **9** 166–74
[11] Mulaosmanovic H *et al* 2017 Switching kinetics in nanoscale hafnium oxide based ferroelectric field-effect transistors *ACS Appl. Mater. Interfaces* **9** 3792–8
[12] Querlioz D, Bichler O, Vincent A F and Gamrat C 2015 Bioinspired programming of memory devices for implementing an inference engine *Proc. IEEE* **103** 1398–416
[13] Lecun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition *Proc. IEEE* **86** 2278–324
[14] Stimberg M, Brette R and Goodman D F 2019 Brian 2, an intuitive and efficient neural simulator *eLife* **8** e47314
[15] Lobov S A, Chernyshov A V, Krilova N P, Shamshin M O and Kazantsev V B 2020 Competitive learning in a spiking neural network: towards an intelligent pattern classifier *Sensors* **20** 500
[16] Srinivasan G and Roy K 2019 ReStoCNet: residual stochastic binary convolutional spiking neural network for memory-efficient neuromorphic computing *Front. Neurosci.* **13** 189
[17] Diehl P and Cook M 2015 Unsupervised learning of digit recognition using spike-timing-dependent plasticity *Front. Comput. Neurosci.* **9** 99
[18] Wang Z *et al* 2017 Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing *Nat. Mater.* **16** 101
[19] Kim S, Du C, Sheridan P, Ma W, Choi S and Lu W D 2015 Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity *Nano Lett.* **15** 2203–11
[20] Campbell K A, Drake K T and Barney Smith E H 2016 Pulse shape and timing dependence on the spike-timing dependent plasticity response of ion-conducting memristors as synapses *Front. Bioeng. Biotechnol.* **4** 97
[21] Gokmen T and Vlasov Y 2016 Acceleration of deep neural network training with resistive cross-point devices: design considerations *Front. Neurosci.* **10** 333
[22] Begon-Lours L, Halter M, Pineda D D, Bragaglia V, Popoff Y, la Porta A, Jubin D, Fompeyrine J and Offrein B J 2021 A back-end-of-line compatible, ferroelectric analog non-volatile memory *2021 IEEE Int. Memory Workshop (IMW)* (Dresden, Germany) pp 1–4
[23] Zhu Y, Gonzalez-Rosillo J C, Balaish M, Hood Z D, Kim K J and Rupp J L M 2021 Lithium-film ceramics for solid-state lithionic devices *Nat. Rev. Mater.* **6** 313
[24] Yousefzadeh A, Stromatias E, Soto M, Serrano-Gotarredona T and Linares-Barranco B 2018 On practical issues for stochastic STDP hardware with 1 bit synaptic weights *Front. Neurosci.* **12** 665
[25] Wong H-S P, Lee H-Y, Yu S, Chen Y-S, Wu Y, Chen P-S, Lee B, Chen F T and Tsai M-J 2012 Metal-oxide RRAM *Proc. IEEE* **100** 1951–70
[26] Le Gallo M and Sebastian A 2020 Phase-change memory *Memristive Devices for Brain-Inspired Computing* ed S Spiga, A Sebastian, D Querlioz and B Rajendran (Woodhead Publishing) pp 63–96

[27] Athle R, Persson A E O, Troian A and Borg M 2022 Top electrode engineering for freedom in design and implementation of ferroelectric tunnel junctions based on $Hf_{1-x}Zr_xO_2$ *ACS Appl. Electron. Mater.* **4** 1002–9
[28] Nail C *et al* 2016 Understanding RRAM endurance, retention and window margin trade-off using experimental results and simulations *2016 IEEE Int. Electron Devices Meet. (IEDM)*
[29] She X, Long Y and Mukhopadhyay S 2019 Improving robustness of ReRAM-based spiking neural network accelerator with stochastic spike-timing-dependent-plasticity (arXiv:190905401)