frontiers in Neuroscience

ORIGINAL RESEARCH
published: 29 June 2016
doi: 10.3389/fnins.2016.00241

Stochastic Synapses Enable Efficient Brain-Inspired Learning Machines

Emre O. Neftci[1], Bruno U. Pedroni[2], Siddharth Joshi[3], Maruan Al-Shedivat[4] and Gert Cauwenberghs[2]*

[1] Department of Cognitive Sciences, University of California, Irvine, Irvine, CA, USA, [2] Department of Bioengineering, University of California, San Diego, La Jolla, CA, USA, [3] Electrical and Computer Engineering Department, University of California, San Diego, La Jolla, CA, USA, [4] Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, USA

Recent studies have shown that synaptic unreliability is a robust and sufficient mechanism for inducing the stochasticity observed in cortex. Here, we introduce Synaptic Sampling Machines (S2Ms), a class of neural network models that uses synaptic stochasticity as a means to Monte Carlo sampling and unsupervised learning. Similar to the original formulation of Boltzmann machines, these models can be viewed as a stochastic counterpart of Hopfield networks, but where stochasticity is induced by a random mask over the connections. Synaptic stochasticity plays the dual role of an efficient mechanism for sampling, and a regularizer during learning akin to DropConnect. A local synaptic plasticity rule implementing an event-driven form of contrastive divergence enables the learning of generative models in an on-line fashion. S2Ms perform equally well using discrete-timed artificial units (as in Hopfield networks) or continuous-timed leaky integrate and fire neurons. The learned representations are remarkably sparse and robust to reductions in bit precision and synapse pruning: removal of more than 75% of the weakest connections followed by cursory re-learning causes a negligible performance loss on benchmark classification tasks. The spiking neuron-based S2Ms outperform existing spike-based unsupervised learners, while potentially offering substantial advantages in terms of power and complexity, and are thus promising models for on-line learning in brain-inspired hardware.

Keywords: stochastic processes, spiking neural networks, synaptic plasticty, unsupervised learning, Hopfield networks, regularization, synaptic transmission

OPEN ACCESS

Edited by:
Themis Prodromakis, University of Southampton, UK

Reviewed by:
Damien Querlioz, University Paris-Sud, France
Doo Seok Jeong, Korea Institute of Science and Technology, South Korea

****Correspondence:***
Emre O. Neftci
eneftci@uci.edu

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *26 January 2016*
Accepted: *17 May 2016*
Published: *29 June 2016*

Citation:
Neftci EO, Pedroni BU, Joshi S, Al-Shedivat M and Cauwenberghs G (2016) Stochastic Synapses Enable Efficient Brain-Inspired Learning Machines. Front. Neurosci. 10:241. doi: 10.3389/fnins.2016.00241

1. INTRODUCTION

The brain's cognitive power emerges from a collective form of computation extending over very large ensembles of sluggish, imprecise, and unreliable components. This realization spurred scientists and engineers to explore the remarkable mechanisms underlying biological cognitive computing by reverse engineering the brain in "neuromorphic" silicon, providing a means to validate hypotheses on neural structure and function through "analysis by synthesis." Successful realizations of large-scale neuromorphic hardware (Schemmel et al., 2010; Indiveri et al., 2011; Benjamin et al., 2014; Merolla et al., 2014; Park et al., 2014; Giulioni et al., 2015) are confronted with challenges in configuring and deploying them for solving practical tasks, as well as identifying the domains of applications where such devices could excel. The central challenge is to devise a neural computational substrate that can efficiently perform the necessary inference and learning operations in a scalable manner using limited memory resources and limited computational primitives, while relying on temporally and spatially local information.

Recently, one promising approach to configure neuromorphic systems for practical tasks is to take inspiration from machine learning, namely artificial and deep neural networks. Despite the fact that neural networks were first inspired by biological neurons, the mapping of modern machine learning techniques onto neuromorphic architectures requires overcoming several conceptual barriers. This is because machine learning algorithms and artificial neural networks are designed to operate efficiently on digital processors, often using batch, discrete-time, iterative updates (lacking temporal locality), shared parameters and states (lacking spatial locality), and minimal interprocess communication that optimize the capabilities of GPUs or multicore processors.

Here, we report a class of stochastic neural networks that overcomes this conceptual barrier while offering substantial improvements in terms of performance, power, and complexity over existing methods for unsupervised learning in spiking neural networks. Similarly to how Boltzmann machines were first proposed (Hinton and Sejnowski, 1986), these neural networks can be viewed as a stochastic counterpart of Hopfield networks (Hopfield, 1982), but where stochasticity is caused by multiplicative noise at the connections (synapses).

As in neural sampling (Fiser et al., 2010; Berkes et al., 2011), neural states (such as instantaneous firing rates or individual spikes) in our model are interpreted as Monte Carlo samples of a probability distribution. The source of the stochasticity in neural samplers is often unspecified, or is assumed to be caused by independent, background Poisson activities (Petrovici et al., 2013; Neftci et al., 2014). Background Poisson activity can be generated self-consistently in balanced excitatory-inhibitory networks (van Vreeswijk and Sompolinsky, 1996), or by using finite-size effects and neural mismatch (Amit and Brunel, 1997). Although a large enough neural sampling network could generate its own stochasticity in this fashion, the variability in the spike trains decreases strongly as the firing rate increases (Fusi and Mattia, 1999; Brunel, 2000; Moreno-Bote, 2014), unless there is an external source of noise or some parameters are fine-tuned. Furthermore, when a neural sampling network generates its own noise, correlations in the background activity can play an adverse role in its performance (Probst et al., 2015). Correlations can be mitigated by adding feed-forward connectivity between a balanced network (or other dedicated source of Poisson spike trains) and the neural sampler, but such solutions entail increased resources in neurons, connectivity, and ultimately energy. Therefore, the above techniques for generating stochasticity are not ideal for neural sampling.

On the other hand, synaptic unreliability can induce the necessary stochasticity without requiring a dedicated source of Poisson spike trains. The unreliable transmission of synaptic vesicles in biological neurons is a well studied phenomenon (Katz, 1966; Branco and Staras, 2009), and many studies suggested it as a major source of stochasticity in the brain (Abbott and Regehr, 2004; Faisal et al., 2008; Yarom and Hounsgaard, 2011; Moreno-Bote, 2014). In the cortex, synaptic failures were argued to reduce energy consumption while maintaining the computational information transmitted by the post-synaptic neuron (Levy and Baxter, 2002). More recent work suggested synaptic sampling as a mechanism for representing uncertainty in the brain (Aitchison and Latham, 2015), and its role in synaptic plasticity and rewiring (Kappel et al., 2015).

We show that uncertainty at the synapse is sufficient in inducing the variability necessary for probabilistic inference in brain-like circuits. Strikingly, we find that the simplest model of synaptic unreliability, called *blank-out* synapses, can vastly improve the performance of spiking neural networks in practical machine learning tasks over existing solutions, while being extremely easy to implement in hardware (Goldberg et al., 2001), and often naturally occurring in emerging memory technologies (Yu et al., 2013; Saïghi et al., 2015; Al-Shedivat et al., 2015a). In blank-out synapses, for each pre-synaptic spike-event, a synapse evokes a response at the post-synaptic neuron with a probability smaller than one. In the theory of stochastic processes, the operation of removing events from a point process with a probability p is termed p-thinning. Thinning a point process has the interesting property of making the process more Poisson-like. Along these lines, a recent study showed that spiking neural networks endowed with stochastic synapses robustly generate Poisson-like variability over a wide range of firing rates (Moreno-Bote, 2014), a property observed in the cortex. The iterative process of adding spikes through neuronal integration and removing them through probabilistic synapses causes the spike statistics to be Poisson-like over a large range of firing rates. A neuron subject to other types of noise, such as white noise injected to the soma, spike jitter and random synaptic delays tends to fire more regularly for increasing firing firing rates.

Given the strong inspiration from Boltzmann machines, and that probabilistic synapses are the main source of stochasticity for the sampling process, we name such networks Synaptic Sampling Machines (S2Ms) (**Figure 1**). Synaptic noise in the S2M is not only an efficient mechanism for implementing stochasticity in spiking neural networks but also plays the role of a regularizer during learning, akin to DropConnect (Wan et al., 2013). This technique was used to train artificial neural networks used multiplicative noise on one layer of a feed-forward network for regularization and decorrelation.

Compared to the neural sampler used in Neftci et al. (2014), the S2M comes at the cost of a quantitative link between the parameters of the probability distribution and those of the spiking neural network. In a machine learning task, this does not pose a problem since the parameters of the spiking neural network can be trained with a learning rule such as event-driven Contrastive Divergence (eCD). ECD is an on-line training algorithm for a subset of stochastic spiking neural networks (Neftci et al., 2014): The stochastic neural network produces samples from a probability distribution, and Spike Timimg Dependent Plasticity (STDP) carries out the weight updates according to the Contrastive Divergence rule (Hinton, 2002) in an online, asynchronous fashion.

S2Ms trained with eCD significantly outperform the Restricted Boltzmann Machine (RBM), reaching error rates in an MNIST hand-written digit recognition task of 4.4%,

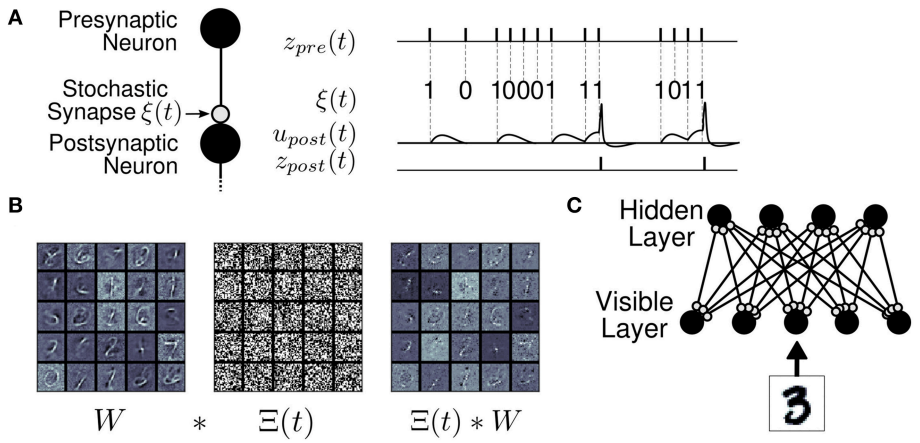

FIGURE 1 | The Synaptic Sampling Machine (S2M). (A) At every occurrence of a pre-synaptic event, a pre-synaptic event is propagated to the post-synaptic neuron with probability p. **(B)** Synaptic stochasticity can be viewed as a continuous DropConnect method (Wan et al., 2013) where weights are masked by a binary matrix $\Theta(t)$, where * denotes element-wise multiplication. **(C)** S2M Network architecture, consisting of a visible and a hidden layer.

while using a fraction of the number of synaptic operations during learning and inference compared to our previous implementation (Neftci et al., 2014). Furthermore, the system has two appealing properties: (1) The activity in the hidden layer is very sparse, with less than 10% of hidden neurons being active at any time; (2) The S2M is remarkably robust to the pruning and the down-sampling of the weights: Upon pruning the top 80% of the connections (sorted by weight values) and re-training (32 k samples), we observed that the error rate remained below 5%. Overall, the S2M outperforms existing models for unsupervised learning in spiking neural networks while potentially using a fraction of the resources, making it an excellent candidate for implementation in hardware.

The article is structured as follows: In the Methods section, we first describe the S2M within a discrete-time framework to gain insight into the functionality of the sampling process. We then describe the continuous-time, spiking version of the S2M. In the Results section, we demonstrate the ability of S2Ms to learn generative models of the MNIST dataset, and illustrate some of its remarkable features.

2. MATERIALS AND METHODS

2.1. The Synaptic Sampling Machine (S2M) As a Hopfield Network with Multiplicative Noise

We first define the S2M as a modification of the original Boltzmann Machine. Boltzmann machines are Hopfield networks with thermal noise added to the neurons in order to avoid overfitting and falling into spurious local minima (Hinton and Sejnowski, 1986). As in Boltzmann Machines, the S2M introduces a stochastic component to Hopfield networks. But rather than units themselves being noisy, in the S2M the connections are noisy.

In the Hopfield network, neurons are threshold units:

$$z_i[t] = \Theta(u_i[t]) = \begin{cases} r_{\text{on}} & \text{if } u_i[t] \geq 0 \\ r_{\text{off}} & \text{if } u_i[t] < 0 \end{cases} \forall i, \tag{1}$$

and connected recurrently through symmetric weights $w_{ij} = w_{ji}$, where $r_{\text{on}} > r_{\text{off}}$ are two numbers indicating the values of the on and off states, respectively.

For simplicity, we chose blank-out synapses as the model of multiplicative noise. With blank-out synapses, the synaptic input to Hopfield unit i is:

$$u_i[t+1] = \sum_{j=1}^{N} \xi_{ij}^p[t] z_j[t] w_{ij} + b_i, \tag{2}$$

where $\xi_{ij}^p[t] \in \{0, 1\}$ is a Bernoulli process with probability p and b_i is the bias. This corresponds to a weighted sum of Bernoulli variables. Since the $\xi_{ij}^p[t]$ are independent, for large N and p not close to 0 or 1, we can approximate this sum with a Gaussian with mean $\mu_i[t] = b_i + \sum_j p w_{ij} z_j[t]$ and variance $\sigma_i^2[t] = p(1-p)\sum_j (w_{ij} z_j[t])^2$[1]. So,

$$u_i[t+1] = \mu_i[t] + \sigma_i[t]\eta_i[t], \tag{3}$$

where $\eta \sim N(0, 1)$. In other terms $u_i[t+1] \sim N(\mu_i[t], \sigma_i^2[t])$. Since $P(z_i[t+1] = 1|\mathbf{z}[t]) = P(u_i[t+1] \geq 0|\mathbf{z}[t])$, the probability that unit i is active given the network state is equal to one minus the cumulative distribution function of u_i:

$$P(u_i[t+1] \geq 0|\mathbf{z}[t]) = \frac{1}{2}\left(1 + \text{erf}\left(\frac{\mu_i[t]}{\sigma_i[t]\sqrt{2}}\right)\right),$$

[1]The Gaussian approximation of a sum of Bernouilli variables is deemed valid when Np, $N(1-p) > 5$, where N is here the number of draws.

where "erf" stands for the error function. Plugging back to Equation (1) gives:

$$P(z_i[t+1]=1|\mathbf{z}[t]) = P(u_i[t+1]\geq 0|\mathbf{z}[t]) = \frac{1}{2}\left(1+\mathrm{erf}\left(\beta\frac{\sum_j w_{ij}z_j[t]+\frac{b_i}{p}}{\sqrt{\sum_j w_{ij}^2 z_j[t]^2}}\right)\right), \quad (4)$$

$$\beta = \sqrt{\frac{p}{2(1-p)}}.$$

This activation function is the S2M-equivalent of the activation function as defined for neurons of the Boltzmann machines (i.e., the logistic function). However, four properties distinguish the S2M from the Boltzmann machine: (1) The activation function is a (shifted) error function rather than the logistic function, (2) the synaptic contributions to each neuron i are normalized such that Euclidian norm of the vector $(w_{i1}z_1,\ldots,w_{iN}z_N)^\top$ is 1. Interestingly, this means that according to Equation (4), a rescaling of the weights has the same effect of scaling the bias. If the biases are zero, the rescaling has no effect; (3) β, which depends on the probability of the synaptic blank-out noise and the network state, plays the role of thermal noise in the Boltzmann machine; and (4) In general, despite that the connectivity matrix is symmetric, the interactions in the S2M are asymmetric because the effective slope of the activation function depends on the normalizing factor of each neuron.

In the general case, the last property described above suggest that the S2M cannot be expressed as an energy-based model (as in the case for Boltzmann machines), and therefore we cannot easily derive the joint distribution $P(\mathbf{z}[\mathbf{t}])$. Some insight can be gained in the particular case where $r_{\mathrm{on}} = -r_{\mathrm{off}}$. In this case, the denominator in Equation (4) simplifies such that:

$$P(z_i[t+1]=1|\mathbf{z}[t]) = P(u_i[t+1]\geq 0|\mathbf{z}[t]) \quad (5)$$

$$= \frac{1}{2}\left(1+\mathrm{erf}\left(\beta\frac{\sum_j w_{ij}z_j[t]+\frac{b_i}{p}}{r_{\mathrm{on}}\sqrt{\sum_j w_{ij}^2}}\right)\right). \quad (6)$$

In other words: the standard deviation of the inputs σ_i^2 does not depend on the neural states. With the additional constraint that $\sum_i w_{ij}^2 = \sum_j w_{ij}^2$, the connectivity matrix becomes symmetric. In this case, the resulting system is *almost* a Boltzmann machine, with the only exception that the neural activation function is an error function instead of the logistic function. Generally speaking, the error function is reasonably close to the logistic function after a rescaling of the argument. Therefore, the parameters of a Boltzmann machine can be approximately mapped on the S2M with $r_{\mathrm{on}} = -r_{\mathrm{off}}$. To test the quality of this approximation, we compute the Kullback-Leibler (KL) divergence between an exact restricted Boltzmann distribution (P_{exact}) and the distribution sampled by a S2M with $r_{\mathrm{on}} = 1, r_{\mathrm{off}} = -1$ (**Figure 2**) using Equation (6). This computation is repeated in the case of the distribution obtained with Gibbs sampling in the RBM. In order to avoid zero probabilities, we added 1 to the number of occurrences of each state. As expected, the KL-divergence between the S2M and the exact distribution reaches a plateau due to the non-logistic activation function. However, the results show that in networks of this size the distribution sampled by the S2M has the same KL-divergence as the RBM obtained after 10^5 iterations of Gibbs sampling, which is more than the typical number of iterations used for many tasks (Hinton et al., 2006). We premise that the S2Ms with all or none activation values ($r_{\mathrm{on}} \neq r_{\mathrm{off}} = 0$) behave in a manner that is sufficiently similar, so that learning in the S2M with Contrastive Divergence (CD) is approximately valid. In the Results section, we test this premise on the MNIST hand-written digit machine learning task.

2.1.1. Learning Rule for RBM and S2Ms

The training of RBMs proceeds in two phases. During the first "data" phase, the states of the visible units are clamped to a given vector of the training set, then the states of the hidden units are sampled. In a second "reconstruction" phase, the network is allowed to run freely. Using the statistics collected during sampling, the weights are updated in a way that they maximize the likelihood of the data. Collecting equilibrium statistics over the data distribution in the reconstruction phase is often computationally prohibitive. The CD algorithm has been proposed to mitigate this (Hinton, 2002; Hinton and Salakhutdinov, 2006): the reconstruction of the visible units' activity is achieved by sampling them conditioned on the values of the hidden units. This procedure can be repeated k times (the rule is then called CD_k), but relatively good convergence is obtained for the equilibrium distribution even for one iteration. The CD learning rule is summarized as follows:

$$\Delta w_{ij} = \epsilon(\langle v_i h_j\rangle_{\mathrm{data}} - \langle v_i h_j\rangle_{\mathrm{recon}}), \quad (7)$$

where v_i and h_j are the activities in the visible and hidden layers, respectively. The notations $\langle\cdot\rangle_{data}$ and $\langle\cdot\rangle_{recon}$ refer to expectations computed during the "data" phases and the "reconstruction" sample phases, respectively. The learning of the biases follows a similar rule. The learning rule for the S2M was identical to that of the RBM.

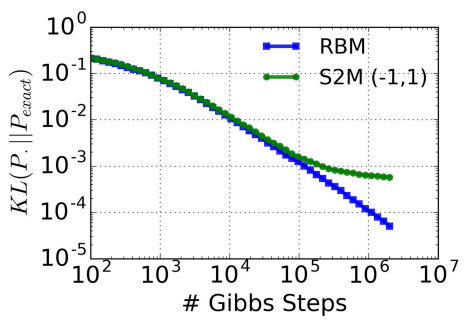

FIGURE 2 | KL-divergence between a restricted Boltzmann distribution (P_{exact}, 5 hidden units, 5 visible units) and the distribution sampled by a S2M with $r_{\mathrm{on}} = 1, r_{\mathrm{off}} = -1$ and matched parameters. As a reference, the KL-divergence using an RBM is shown (blue curve). The saturation of the green curve is caused by the inexact activation function of the S2M unit (error function instead of the logistic function).

2.2. Spiking Synaptic Sampling Machines

2.2.1. Neuron and Synapse Model

Except for the noise model, the neuron and synapse models used in this work are identical to those described in Neftci et al. (2014) and are summarized here for convenience. The neuron's membrane potential below firing threshold θ is governed by the following differential equation:

$$C\frac{\mathrm{d}}{\mathrm{d}t}u_i = -g_L u_i + I_i(t), \quad u_i(t) \in (-\infty, \theta), \tag{8}$$

where C is a membrane capacitance, u_i is the membrane potential of neuron i, g_L is a leak conductance, $I_i(t)$ is the time-varying input current and θ is the neuron's firing threshold. When the membrane potential reaches θ, an action potential is elicited. After a spike is generated, the membrane potential is clamped to the reset potential u_{rst} for a refractory period τ_r.

In the spiking S2M, the currents $I_i(t)$ depend on the layer the neuron is situated in (visible v or hidden h). For a neuron i in the visible layer v

$$\begin{aligned} I_i(t) =& I_i^d(t) + I_i^v(t), \\ \tau_{syn}\frac{\mathrm{d}}{\mathrm{d}t}I_i^v(t) =& -I_i^v + \sum_{j=1}^{N_h} \xi_{h,ji}^p(t) q_{ji} h_j(t) + b_{v_i}, \end{aligned} \tag{9}$$

where $I_i^d(t)$ is a current representing the data (i.e., the external input), $I^v(t)$ is the feedback from the hidden layer activity and the bias. The q's are the respective synaptic weights, $\xi_{v,ij}^p$ is a Bernoulli process (i.e., "coin flips") with probability p implementing the stochasticity at the synapse, and b_{v_i} are constant currents representing the biases of the visible neurons. Spike trains are represented by a sum of Dirac delta pulses centered on the respective spike times:

$$h_j(t) = \sum_{k \in Sp_j^h} \delta(t - t_k), \quad v_i(t) = \sum_{k \in Sp_i^v} \delta(t - t_k) \tag{10}$$

where Sp_j^h, Sp_j^v are the sets the spike times of the hidden neuron h_j and visible neurons v_i, respectively, and $\delta(t) = 1$ if $t = 0$ and 0 otherwise.

For a neuron j in the hidden layer h,

$$\begin{aligned} I_j(t) =& I_j^h(t), \\ \tau_{syn}\frac{\mathrm{d}}{\mathrm{d}t}I_j^h(t) =& -I_j^h + \sum_{i=1}^{N_v} \xi_{v,ij}^p(t) q_{ij} v_i(t) + b_{h_j}, \end{aligned} \tag{11}$$

where $I^h(t)$ is the feedback from the visible layer, and $v(t)$ are Poisson spike trains of the visible neurons defined in Equation (10) and b_{h_j} are the biases of the hidden neurons. The dynamics of I^h and I^v correspond to a first-order linear filter, so each incoming spike results in Post–Synaptic Potentials (PSPs) that rise and decay exponentially (i.e., alpha-PSP) (Gerstner and Kistler, 2002).

2.2.2. Event-Driven CD Synaptic Plasticity Rule

Event-driven Contrastive Divergence is an online variation of CD amenable for implementation in neuromorphic and brain-inspired hardware: by interpreting spikes as samples of a probability distribution, a possible neural mechanism for implementing CD is to use STDP synapses to carry out CD-like updates.

The weight update in eCD is a modulated, pair-based STDP rule with learning rate ϵ_q:

$$\frac{\mathrm{d}}{\mathrm{d}t}q_{ij} = \epsilon_q g(t)\,\mathrm{STDP}_{ij}(v_i(t), h_j(t)) \tag{12}$$

where $g(t) \in \mathbb{R}$ is a zero-mean global gating signal controlling the data vs. reconstruction phase, q_{ij} is the weight of the synapse and $v_i(t)$ and $h_j(t)$ refer to the spike trains of neurons v_i and h_j, defined as in Equation (10). The same rule is applied to learn biases b_v and b_h:

$$\frac{\mathrm{d}}{\mathrm{d}t}b_{v_i} = \epsilon_b \frac{1}{2} g(t)\,\mathrm{STDP}_i(v_i(t), v_i(t)), \tag{13}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}b_{h_i} = \epsilon_b \frac{1}{2} g(t)\,\mathrm{STDP}_i(h_i(t), h_i(t)), \tag{14}$$

where ϵ_b is the learning rate of the biases. The weight update is governed by a symmetric STDP rule with temporal window $K(t) = K(-t), \forall t$:

$$\begin{aligned} \mathrm{STDP}_{ij}(v_i(t), h_j(t)) =& v_i(t) A_{h_j}(t) + h_j(t) A_{v_i}(t), \\ A_{h_j}(t) =& A \int_{-\infty}^{t} \mathrm{d}s K(t-s) h_j(s), \\ A_{v_i}(t) =& A \int_{-\infty}^{t} \mathrm{d}s K(s-t) v_i(s), \end{aligned} \tag{15}$$

with $A > 0$ defining the magnitude of the weight updates.

Although any symmetric learning window can be used, for simplicity, we used a nearest neighbor update rule where:

$$K(t-s) = \begin{cases} 1 & \text{if } |t-s| < \tau_{STDP} \\ 0 & \text{otherwise} \end{cases}$$

In our implementation, updates are additive and weights can change polarity. A global modulatory signal that is synchronized with the data clamping phase modulates the learning to implement CD:

$$g(t) = \begin{cases} 1 & \text{if } mod(t, 2T) \in (\tau_{br}, T) \\ -1 & \text{if } mod(t, 2T) \in (T + \tau_{br}, 2T)\,, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

The signal $g(t)$ switches the behavior of the synapse from Long-Term Potentiation (LTP) to Long-Term Depression (LTD) (i.e., Hebbian to Anti-Hebbian). The temporal average of $g(t)$ vanishes to balance LTP and LTD. The modulation factor is zero during some time τ_{br}, so that the network samples closer to its stationary distribution when the weights updates are carried out. The time constant τ_{br} corresponds to a "burn-in" time of Markov Chain

Monte Carlo (MCMC) sampling and depends on the overall network dynamics. (Neftci et al., 2014) showed that, when pre-synaptic neurons and post-synaptic neurons fire according to Poisson statistics, eCD is equivalent to CD. The effective learning window is:

$$\epsilon = 2A\frac{T-\tau_{br}}{2T}.$$

We note that the learning rate between the S2M and the spiking spiking S2M cannot be compared directly because there is no direct relationship between the synaptic weights q and w.

In the spiking S2M, weight updates are carried out even if a spike is dropped at the synapse. This speeds up learning without adversely affecting the entire learning process because spikes dropped at the synapses are valid samples in the sense of the sampling process. During the data phase, the visible units were driven with constant currents equal to the logit of the pixel intensity (bounded to the range $[10^{-5}, 0.98]$ in order to avoid infinitely large currents), plus a white noise process of low amplitude σ to simulate sensor noise.

2.2.3. Synaptic Blank-Out Noise

Perhaps the simplest model of synaptic noise is the blank-out noise: for each spike-event, the synapse has a probability p of evoking a PSP, and a probability $(1-p)$ of evoking no response. This is equivalent to modifying the input spike train to each synapse such that spikes are dropped (blanked-out) with probability p. In particular, for a Poisson spike train of rate ν, the blank-out with probability p gives a Poisson spike train with rate $p\nu$ (Parzen, 1999; Goldberg et al., 2001). For a periodic (regular) spike train, stochastic synapses add stochasticity to the system. The coefficient of variation of the Inter–Spike Interval (ISI) becomes $\sqrt{1-p}$. The periodic spike train thus tends to a Poisson spike train when $p \to 0$, with the caveat that events occur at integer multiples of the original ISI. Intuitively, the neural network cycles through deterministic neural integration and stochastic synapses. (Moreno-Bote, 2014) found that the recurring process of adding spikes through neuronal integration and removing them through stochastic synapses causes the spike statistics to remain Poisson-like (constant Fano Factor) over a large dynamical range. Synaptic stochasticity is thus a biologically plausible source of stochasticity in spiking neural networks, and can be very simply implemented in software and hardware (Goldberg et al., 2001).

2.2.4. Spiking Neural Networks with Poisson Input and Blank-Out Synapses

This section describes the neural activation function of leaky Integrate & Fire (I&F) neurons. The collective dynamics of spiking neural circuits driven by Poisson spike trains is often studied in the diffusion approximation (Brunel and Hakim, 1999; Fusi and Mattia, 1999; Wang, 1999; Brunel, 2000; Renart et al., 2003; Tuckwell, 2005; Deco et al., 2008). In this approximation, the firing rates of individual neurons are replaced by a common time-dependent population activity variable with the same mean and two-point correlation function as the original variables, corresponding here to a Gaussian process. The approximation is true when the following assumptions are verified:

(1) the charge delivered by each spike to the post-synaptic neuron is small compared to the charge necessary to generate an action potential,
(2) there is a large number of afferent inputs to each neuron,
(3) the spike times are uncorrelated.

In the diffusion approximation, only the first two moments of the synaptic current are retained. The currents to the neuron, $I(t)$, can then be decomposed as:

$$I(t) = \mu + \sigma\,\eta(t), \tag{17}$$

where $\mu = \langle I(t)\rangle = p\sum_j q_j\nu_j + b$ and $\sigma^2 = p\sum_j q_j^2\nu_j$, ν_j is the firing rate of pre-synaptic neuron j, and $\eta(t)$ is the white noise process. Note that a Poisson spike train of mean rate ν with spikes removed with probability p is a Poisson spike train with parameter $p\nu$. As a result, blank-out synapses have the effect of scaling the mean and the variance by p.

In this case, the neuron's membrane potential dynamics is an Ornstein-Uhlenbeck process (Gardiner, 2012). For simplicity of presentation, the reset voltage was set to zero ($u_{rst} = 0$). We consider the case where the synaptic time constant dominates the membrane time constant. In other words, the membrane potential closely follows the dynamics of the synaptic currents and the effect of the firing threshold and the reset in the distribution of the membrane potential can be neglected. This problem was studied in great detail with first order approximations in τ_m/τ_{syn} (Petrovici et al., 2013). For comparison with the S2M, we focus here on the case $\tau_m = 0$. In this case, the stationary distribution is a Gaussian distribution:

$$u \sim N(\frac{\mu}{g_L}, \frac{\sigma^2}{2g_L^2\tau_{syn}}).$$

Neurons such that $p(u > 0|\nu)$ fire at their maximum rate of $\frac{1}{\tau_{ref}}$. Following similar steps as in the S2M, the firing rate of a neuron i becomes:

$$\begin{aligned}\nu_i &= \frac{1}{\tau_{ref}}\frac{1}{2}\left(1+\mathrm{erf}\left(\frac{\mu_i}{\sigma_i}\sqrt{\tau_{syn}}\right)\right),\\ &= \frac{1}{\tau_{ref}}\frac{1}{2}\left(1+\mathrm{erf}\left(\sqrt{p}\frac{\sum_j q_{ij}\nu_j + \frac{b_i}{p}}{\sqrt{\sum_j q_{ij}^2\nu_j}}\sqrt{\tau_{syn}}\right)\right).\end{aligned} \tag{18}$$

Equation (18) clarifies how the noise amplitude σ affects the neural activation ν, and thus allows a quantitative comparison with the S2M. Similarly to Equation (4), the input is effectively normalized by the variability in the inputs (where on/off values are replaced by firing rates). These strong similarities suggest that the S2M and the spiking neural network are very similar. Using computer simulations of the MNIST learning task, in the Results section we show that the two networks indeed perform similarly.

2.3. Training Protocol and Network Structure for MNIST

2.3.1. RBM and the S2M

The network consisted of 1294 neuron, partitioned into 794 visible neurons and 500 hidden neurons (**Figure 3**). In the results, we used $r_{on} = 1$ and $r_{off} = 0$ for the S2M neurons for a closer comparison with spiking neurons and the spiking S2M. The visible neurons were partitioned into 784 sensory (data) neurons, denoted $\mathbf{v_d}$ and 10 class label neurons, denoted $\mathbf{v_c}$. The S2M is similar to the RBM in all manners, except that the units are threshold functions, and each weight is multiplied by independently drawn binomials ($p = 0.5$) at every step of the Gibbs sampling.

In both cases the training set was randomly partitioned into batches of equal size N_{batch}. One epoch is defined as the presentation of the full MNIST dataset (60,000 digits). We used CD-1 to train the RBM and the S2M.

Classification is performed by choosing the most likely label given the input, under the learned model. To estimate the discrimination performance, we sampled the distribution of class units $P(class|digit)$ using multiple Gibbs Sampling chains (50 parallel chains, 2 steps). We take the classification result to be the label associated to the most active class unit averaged across all chains.

For testing the discrimination performance of an energy-based model such as the RBM, it is common to compute the free-energy $F(\mathbf{v_c})$ of the class units (Haykin, 2004), defined as:

$$\exp(-F(\mathbf{v_c})) = \sum_{\mathbf{v_d},\mathbf{h}} \exp(-E(\mathbf{v_d}, \mathbf{v_c}, \mathbf{h})), \tag{19}$$

and selecting $\mathbf{v_c}$ such that the free-energy is minimized. The free-energy computes the probabilities of a given neuron to be active, using the joint distribution of the RBM. Classification using free energy is inapplicable to S2Ms because it cannot be expressed in terms of an energy-based model (See Section 2). Therefore, throughout the article, free energy-based classification was used only for the RBM.

The learning rate in the RBM and the S2M was linearly reduced during the training, reaching 0 at the end of the learning. Both RBM and S2M were implemented on a GPU using the Theano package (Bergstra et al., 2010).

2.3.2. Spiking S2M

The spiking S2M network structure is identical to above. Similarly to Neftci et al. (2014), for a given digit, each neuron in layer $\mathbf{v}$ was injected with currents transformed according to a logit function $I_i^d \propto \log\left(\frac{s_i}{1-s_i\tau_r}\right)$, where s_i is the value of pixel i. To avoid saturation of the neurons using the logit function, the pixel values of the digits were bounded to the range $[10^{-5}, 0.98]$

Training followed the eCD rule described in Equation (12). The learning rate was decayed linearly during the training, reaching 0 at the end of the learning. Similarly to the S2M, the discrimination performance is evaluated by sampling the activities of the class units for every digit in the test dataset. In the spiking S2M, this is equivalent to identifying the neuron that has the highest firing rate. The spiking neural network was implemented in the spike-based neural simulator Auryn, optimized for recurrent spiking neural networks with synaptic plasticity (Zenke and Gerstner, 2014). Connections in the S2M are symmetric, but due to the constraints in the parallel simulator, the connections were implemented using two separate synapses (one in each direction), and periodically symmetrized to maintain symmetry (every 1000 s of simulated time). A complete list of parameters used in our simulator is provided in the Appendix **(Table A1)**.

3. RESULTS

We demonstrate two different implementations of synaptic sampling machines (S2Ms), one spiking and one non-spiking. The non-spiking S2M is a variant of the original RBM, used to provide insight into the role of stochastic synapses in neural networks, and to justify spiking S2Ms. The spiking S2M consisted of a network of deterministic I&F spiking neurons, connected through stochastic (blank-out) synapses (**Figure 1**). In the Section 2, we show that S2Ms with activation values $(-1, +1)$ are similar to Boltzmann machines, except that the activation function of the neuron is an error function instead of the logistic function, and where the input is invariant to rescaling. We assumed that the S2M with activation levels (0,1) and the spiking S2M are sufficiently similar to the Boltzmann distribution, such that CD and eCD as originally described in Neftci et al. (2014) are approximately valid. We test this assumption through computer simulations of the S2Ms in an MNIST hand-written digit learning task. Independently, Müller explored the idea of unreliable connections in Hopfield units, reaching similar conclusion on the role of the blank-out probabilities in the Boltzmann temperature, as well as such networks being good approximations of restricted Boltzmann machines (Müller, 2015).

3.1. Unsupervised Learning of MNIST Handwritten Digits in Synaptic Sampling Machines

Similarly to RBMs, S2Ms can be trained as generative models. Generative models have the advantage that they can act simultaneously as classifiers, content-addressable memories, and carry out probabilistic inference. We demonstrate these features in a MNIST hand-written digit task (LeCun et al., 1998), using networks consisting of two layers, one visible and one hidden (**Figure 3**).

Learning results are summarized in **Table 1**. The spiking S2M appears to slightly outperform the S2M, although a direct comparison between the two is not possible because the sampling mechanism and the batch sizes are different. We find that spiking S2Ms attain classification error rates that slightly outperform the machine learning algorithm (error rates: spiking S2M 4.4 vs. RBM 5%), even after much fewer repetitions of the dataset (**Figure 4**). To date, this is the best performing spike-based unsupervised learner on MNIST. The spiking network implementing the spiking S2M is many times

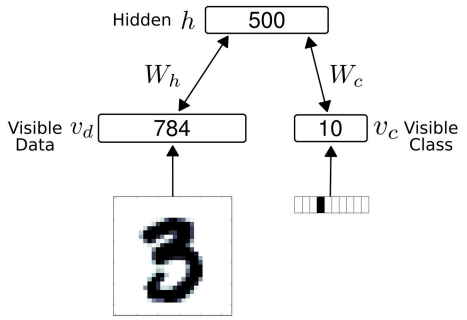

FIGURE 3 | The net work architecture consists of a visible and a hidden layer. The visible layer is partitioned into 784 sensory (data) neurons ($\mathbf{v_d}$) and 10 class label neurons ($\mathbf{v_c}$). During data presentation, the activities in the visible layer are clamped to the data, consisting of a digit and its label (one-hot coded). In all models, the weight matrix between the visible layer and the hidden layer is symmetric.

TABLE 1 | Error rates on MNIST hand-written digits task.

Model, $n_H = 500$, 60 k digits training set	MNIST Error
spiking S2M + Event-driven CD (this work)	4.4%
spiking S2M + Event-driven CD + 74% connection pruning + relearning (this work)	5.0%
spiking S2M + Event-driven CD + 4 bit synaptic weights (this work)	5.2%
spiking S2M + Event-driven CD + 2 bit synaptic weights (this work)	7.8%
S2M + Standard CD (this work)	4.5%
Gibbs Sampling + Standard CD	4.7%
Neural Sampling + Event-driven CD (Neftci et al., 2014)	8.1%
Neural Sampling + Event-driven CD (Neftci et al., 2014) (5 bits post-learning rounding)	10.6%

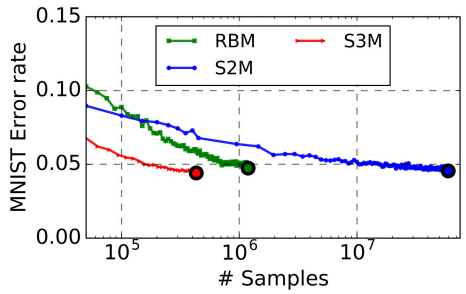

FIGURE 4 | The S2M outperforms its RBM counterpart at the MNIST task. The RBM, the S2M, and the continuous-time spiking S2M were trained to learn a generative model of MNIST handwritten digits. The S2M model is identical to the RBM except that it consists of threshold units with stochastic blank-out synapses with probability 50%. The recognition performance is assessed on the testing dataset which was not used during training (10,000 digits). Error rate in the RBM starts increasing after reaching peak performance, mainly due to decreased ergodicity of the Markov chains and overfitting. Learning in the S2M is slower than in the RBM, as reported with other models using DropConnect (Wan et al., 2013) but it is effective in preventing overfitting. At the end of the training, the recognition performance of the spiking S2Ms (S3M) averaged over eight runs with different seeds reached 4.6% error rate. Due to the computational load of running the spike-based simulations on the digital computer, the spiking S2M was halted earlier than the RBM and the S2M. In spite of the smaller number of digit presentations, the spiking S2M outperformed the RBM and the S2M. This is partly because weight updates in the spiking S2M are undertaken during each digit presentation, rather than after each minibatch. The curves are plotted up to the point where the best performance is reached.

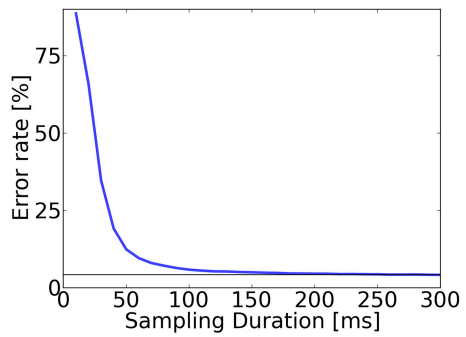

FIGURE 5 | Accuracy evaluation in the spiking S2M. To test recognition, for each digit in the test dataset (10,000 digits), class neurons in the S2Ms are sampled for up to 300 ms. The classification is read out by identifying the group of class label neurons that had the highest activity and averaging over all digits of the test set. The error rate after 50 ms of sampling was above 13% and after 250 ms the error rates typically reached their minimum for this trained network (4.2%, horizontal bar).

smaller than the best performing spike-based unsupervised learner to date (1294 neurons 800 k synapses vs. 7184 neurons, 5 M synapses) (Diehl and Cook, 2015). For comparisons with other recent techniques for unsupervised learning in spiking neural networks, we refer the reader to Diehl and Cook (2015), which provides a recent survey of the MNIST learning task with spiking neural networks.

We tested the speed of digit classification under the trained S2M. We computed the prediction after sampling for a fixed time window that we varied from 0 to 300 ms (**Figure 5**). Results show that the trained network required approximately 250 ms to reach the lowest error rate of 4.2%, and that only 13% of the predictions were incorrect after 50 ms.

Similarly to RBMs, the S2M learns a generative model of the MNIST dataset. This generative model allows to generate digits and reconstruct them when a part of the digit has been occluded. We demonstrate this feature in a pattern completion experiment where the right half of each digit was presented to the network, and the visible neurons associated to the left half of the digit were sampled (**Figure 6**). In most cases, the S2M correctly completed the left half of the digit. **Figure 6** also illustrates the dynamics of the completion, which appears to reach a stationary state after about 200ms. Overall, these results show that the S2M can achieve similar tasks as in RBMs, at a similar or better recognition performance while requiring fewer dataset iterations during learning.

3.2. Representations Learned with Synaptic Sampling Machines are Sparse

In deep belief networks, discriminative performance can improve when using binary features that are only rarely active (Nair and

Hinton, 2009). Furthermore, in hardware sparser representations result in lower communication overhead, and therefore lower power consumption.

To quantify the degree of sparsity in the RBM and the S2M, we computed the average fraction of active neurons. For a similar parametrization, the average activity ratio of S2Ms is much lower than that of RBMs (**Figure 7**). RBMs can be trained to be sparse by added sparsity constraints during learning (Lee et al., 2008), but how such constraints can map to spiking neurons and synaptic plasticity rules is not straightforward. In this regard, it is remarkable that S2Ms are sparse without including any sparsity constraint in the model. The difference in S2M and the S3M curves is likely to be caused by different parameters in the spiking and the non-spiking S2M, as these cannot be exactly matched.

One can gain an intuition on the cause for sparsity in the S2M by examining the network states during learning: In the absence of additive noise, the input-output profile of leaky I&F neurons near the rheobase (minimal current required to cause the neuron to spike) is rectified-linear. Consequently, without positive inputs and positive bias values, the spiking neuron cannot fire, and thus the pre-synaptic weights cannot potentiate. By selecting bias values at or below zero, this feature causes neurons in the network to be progressively recruited, thereby promoting sparsity. This is to be contrasted in the case of neurons with additive noise (such as white noise with constant amplitude), which can fire even if the inputs are well below rheobase. In the S2M, the progressive recruitment can be observed in **Figure 8**: early in the training, hidden neurons are completely silent during the reconstruction phase. Because there is no external stimulus during the reconstruction phase, only neurons that were active during the data phase are active during the reconstruction phase. After the presentation of 60 digits, the activity appears to "grow" from the data phase. Note that similar arguments can be made in the case of S2Ms implemented with threshold neurons.

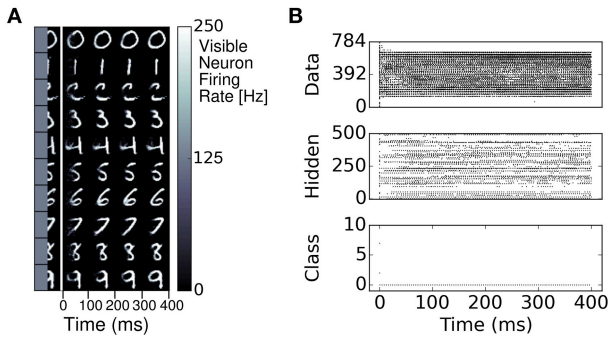

FIGURE 6 | Pattern completion in the spiking S2M. (A) The right half of a digit and its label is presented to the visible layer of the S2M. The ensuing activity in the visible layer is partitioned in 100 ms bins and the average activity during each bin is presented in the color plots to the right. The network completes the left half of the digit. **(B)** Raster plot of the pattern completion for the first row of **(A)**.

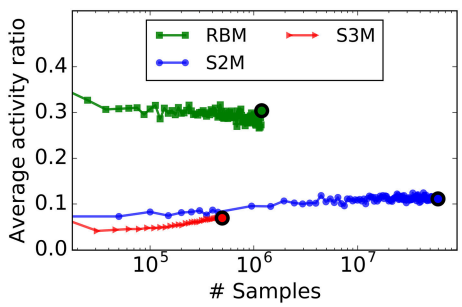

FIGURE 7 | For a similar parametrization, S2Ms learn sparser representations than RBMs. The curves plot the average number of active units during learning in the hidden layer, computed over 50 digits for the S2M and the RBM, and 1000 digits for the spiking S2M. In the case of the spiking S2M (S3M), the average activity ratio is the average firing rate of the neurons in the network divided by the maximum firing rate t_{ref}^{-1}. The colored dot indicates the point where the peak of the recognition accuracy was reached. Nearly three times fewer units were active in the S2M than in the RBM at the designated points, and even fewer in the case of the spiking S2M.

3.3. Robustness of spiking S2Ms to Synapse Pruning and Weight Down-sampling

Sparse networks that require using very few bits for storage can have a lower memory cost, along with a smaller communication and energy footprint. To test the storage requirements for the spiking S2M and its robustness to pruning connections, we removed all synapses whose weights were below a given threshold. We varied the threshold such that the ratio of remaining connections spanned the range [0%, 100%].

The resulting recognition performance, plotted against the ratio of connections retained, is shown in **Figure 9A** (black curve). We then re-trained the spiking S2M over 32 epochs and tested against 10, 000 images from the MNIST test dataset. The re-learning substantially recovered the performance loss caused by the weight pruning as shown in **Figure 9A** (red curve). This result suggests that only a relatively small number of connections play a critical role in the underlying model for the recognition task.

3.3.1. Learning Low Precision Weight Synapses

Dedicated memory for storing the synaptic connectivity table and the synaptic weights often occupies the largest area in neuromorphic devices (Merolla et al., 2014; Moradi and Indiveri, 2014). Therefore, the ability to reduce the synaptic precision required for the operation of an algorithm can be very beneficial in hardware. We quantify the effects of lowered precision in inference by downsampling the synaptic weights, while performing computation at full precision. In the context of a hardware implementation, this results in lower memory costs since fewer bits can be used to store the same network. To test the performance of the spiking S2M with lowered resolution, we truncated the weights to a single decimal point. The resulting weights were restricted to less than 128 unique values (7 bits). In **Figure 9B**, we truncated the weights of the previously pruned network to 7 bits, and examined the results over the range of retained connections. The results of testing 10,000 samples of MNIST on this network are shown in **Figure 9B**. The error rate

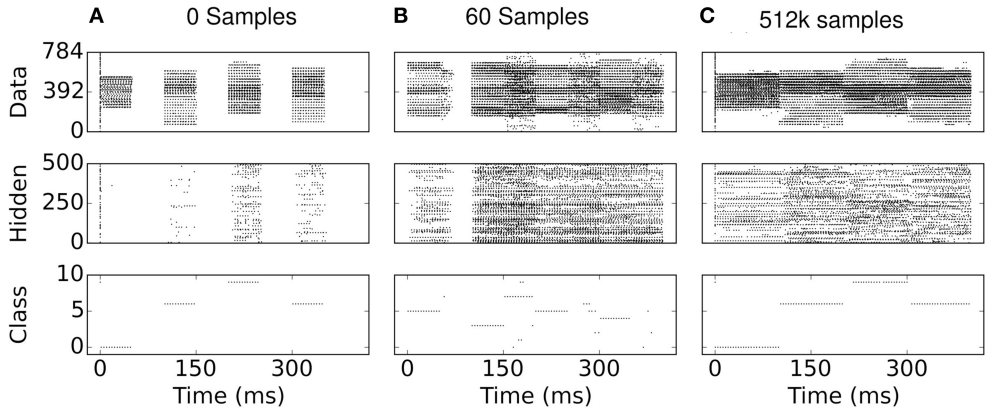

FIGURE 8 | (A–C) Spike rasters during eCD learning. The data phase is 0 – 50 ms and the reconstruction phase is 50 – 100 ms, repeated every 100 ms for a different digit of the training set. Due to the deterministic neural dynamics, a neuron receiving no input can never fire if its bias is below the rheobase (i.e., the minimum amount of current necessary to make a neuron fire). Consequently, at the beginning of learning, the hidden neurons are completely silent in the reconstruction phases, and are gradually recruited during the data phase of the eCD rule (e.g., see 50 ms in **B**).

was about 8 with 12.5% of the total synaptic weights retained and about 5 with 49% of the connections retained.

Another recently proposed technique for reducing the precision of the weights while minimizing impact on performance is the dual-copy rounding technique (Stromatias et al., 2015). In the context of our sampling machine, the key idea is to sample using reduced precision weights, but learn with full precision weights. Dual copy rounding was shown to outperform rounding of the weight after learning.

Training the spiking S2M with dual-copy rounding at 4-bit weights (16 different weight values) resulted in 5.2% error rate at the MNIST task, and rounding at 2 bits (4 weight values) increased this number to 7.8%. Synaptic weight resolution of 4 bits is recognized as a sweet spot for hardware (Pfeil et al., 2012; Merolla et al., 2014). Furthermore, it is a plausible synaptic weight precision for biological synapses: recent analysis of synapses in the rat Hippocampus suggested that each synapse could store about 4–5 bits of information (Bartol et al., 2015).

We note that the dual-copy approach is only beneficial at the inference stage (post-learning). During inference, the high-precision weights can be dropped and only the low-precision weight are maintained. However, during learning it is necessary to maintain full precision weights. Learning with low precision weights is possible using stochastic rounding techniques (Muller and Indiveri, 2015). We could not test stochastic rounding in spiking S2Ms because the symmetry requirements in the spiking neural network connectivity prevent a direct, efficient implementation in multithreaded simulators (such as the used Auryn neural simulator).

3.4. Synaptic Operations and Energy Efficiency in S2Ms

Power consumption in brain-inspired computers is often dominated by synaptic communication and plasticity. Akin to the energy required per multiply accumulate operation (MAC) in digital computers, the energy required by each synaptic operation (SynOp) is one representative metric of the efficiency of brain-inspired computers (Merolla et al., 2014). The reason is that every time one neuron spikes, a large number of synapses of other, possibly physically distant neurons are updated (in practice, hundreds to ten of thousands of synapses).

A SynOp potentially consumes much less energy than a MAC even on targeted GPGPUs, and improvements in energy per SynOP directly translate into energy efficiencies at the level of the task. However, these operations are not directly comparable because it is unclear how many SynOps provide the equivalent of a MAC operation at a given task. To provide a reference to this comparison, we count the number of SynOps and approximate number of MACs necessary to achieve a target accuracy at the MNIST task for spiking S2Ms and RBMs, respectively (**Figure 10**). Strikingly, the spiking S2M achieves SynOp-MAC parity at this task. Given that the energy efficiency of a SynOp is potentially orders of magnitude lower than the MAC (Merolla et al., 2014; Park et al., 2014), this result makes an extremely strong case for hardware implementations of spiking S2Ms. The possible reasons for this parity are two-fold: (1) weight updates are undertaken after presentation of every digit, which mean that fewer repetitions of the dataset are necessary to achieve the same performance. (2) only active connections incur a SynOp. In the RBM, the operations necessary for computing the logistic function, random numbers at the neuron, and the weight updates were not taken into account and would further favor the spiking S2M. The operation necessary for the stochastic synapse in the spiking S2M (a Bernoulli trial) is likely to be minimal because the downstream synapse circuits do not need to be activated when the synapse fails to transmit. Furthermore, the robustness of spiking S2Ms to the pruning of connections (described in Section 3.3) can further strongly reduce the number of SynOps after learning.

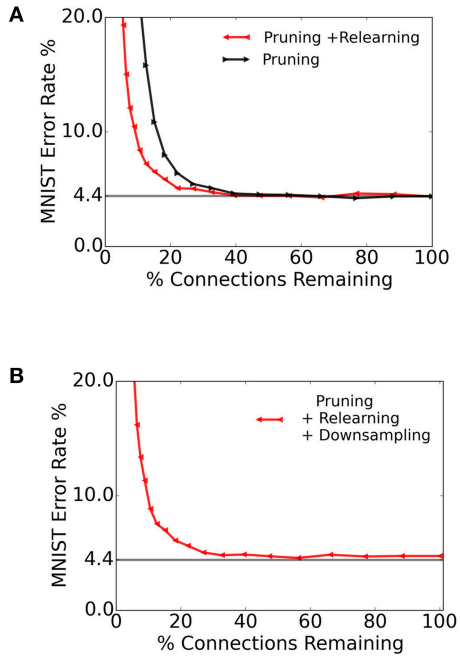

FIGURE 9 | Pruning connections in the spiking S2M and re-learning. **(A)** We test the spiking S2M's robustness to synapse pruning by removing all connections whose weights were below a given threshold. We quantified the effects of pruning by testing the pruned spiking S2M over the entire MNIST test set (black curve). After pruning, we re-trained the network over 32 epochs (corresponding to 32 k sample presentations), which recovered most of the loss due to pruning (red curve). **(B)** Reduced synaptic weight precision in the spiking S2M. After learning, pruning and re-learning, the synaptic weights were downsampled by truncating them to a single decimal point. There remained fewer than 128 unique weight values in the final matrix. The horizontal line is at 4.4%.

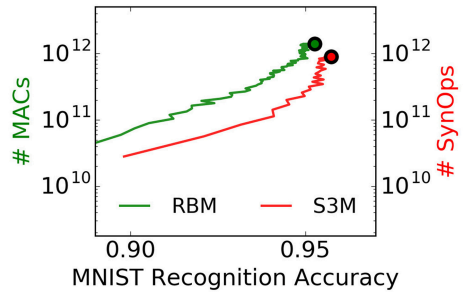

FIGURE 10 | The spiking S2M achieves SynOp-MAC parity at the MNIST task. The number of multiply-accumulate (MAC) operations required for sampling in the RBM during learning is compared to the number of synaptic operations (SynOps) in the spiking S2M during the MNIST learning task. At this task, the spiking S2M (S3M) requires fewer operations to reach the same accuracy during learning. Other necessary operations for the RBM, e.g., additions, random number generations, logistic function calls, and weight updates were not taken into account here, and would further favor the spiking S2M. One reason for the SynOp-MAC parity is that learning in the spiking S2M requires fewer repetitions of data samples to reach the same accuracy compared to the RBM. Another reason is that only active connections in the spiking S2M incur a SynOp. SynOp-MAC parity between the spiking S2M and the RBM is very promising for hardware implementations because a SynOp in dedicated hardware potentially consumes much less power than a MAC in a general purpose digital processor. Note that the non-spiking S2M is not competitive on this measure because it requires N^2 random number generations per Gibbs sampling step in addition to the same number of MACs as the RBM. The curves are plotted up to the point where the best performance is reached.

4. DISCUSSION

The Boltzmann Machine stems from the idea of introducing noise into a Hopfield network (Hinton and Sejnowski, 1986). The noise prevents the state from falling into local minima of the energy, enabling it to learn more robust representations while being less prone to overfitting. Following the same spirit, the S2M introduces a stochastic component to Hopfield networks, but rather than the units themselves being noisy, the connections are noisy.

The stochastic synapse model we considered is blank-out noise, where the synaptic weight is multiplied by binary random variable. This is to be contrasted with additive noise, where the stochastic term such as a white noise process is added to the membrane potential. In artificial neural networks, multiplicative noise forces weights to become either sparse or invariant to rescaling (Nalisnick et al., 2015). When combined with rectified linear activation functions, multiplicative noise tends to increase sparsity (Rudy et al., 2014). Multiplicative noise makes neural responses sparser: In the absence of additive noise, neurons have activation functions very close to being rectified linear (Tuckwell, 2005). In the absence of a depolarizing input, the neuron cannot fire, and thus its synapses cannot potentiate. Consequently, if the parameters are initiated properly, many neurons will remain silent after learning.

Event-driven CD was first demonstrated in a spiking neural network that instantiated a MCMC neural sampling of a target Boltzmann distribution (Neftci et al., 2014). The classification performance of the original model was limited by two properties. First, every neuron was injected with a (additive) noisy current of very large amplitude. Neurally speaking, this corresponded to a background Poisson spike train of 1000 Hz, which considerably increased the network activity in the system. Second, in spite of the large input noise, neurons fired periodically at large firing rates due to an absolute refractory period. This periodic firing evoked strong spike-to-spike correlations (synchrony) that were detrimental to the learning and the sampling. Consequently, the performance of eCD in an MNIST task was significantly lower than when standard CD was used (8.1% error rate).

The performance of the spiking S2M in the MNIST hand-written digits task vastly improved accuracy metrics over our previous results while requiring a fraction of the number of synaptic operations. The reduction in the number of operations was possible because our previous neuron model required an extra background Poisson spike train for introducing noise, whereas the S2M generates noise through stochastic synapses. The improvement in accuracy over our earlier results in Neftci et al. (2014) stems from at least two reasons: (1) Spike-to-spike decorrelations caused by the synaptic noise better condition the plasticity rule by preventing pair-wise synchronization; (2)

Regularization, which mitigates overfitting and the co-adaptation of the units. In the machine learning community, blank-out noise is also known as DropConnect (Wan et al., 2013). It was demonstrated to perform regularization, which helped achieve state-of-the-art results on several image recognition benchmarks, including the best result so far on MNIST without elastic distortions (0.21%). Note that the S2M model did not include domain-specific knowledge, which suggests that its performance may generalize to other problems and datasets.

4.1. Synaptic Unreliability in the Brain

The probabilistic nature of synaptic quantal release is a well known phenomenon (Katz, 1966). Unreliability is caused by the probabilistic release of neurotransmitters at the pre-synaptic terminals (Allen and Stevens, 1994). Detailed slice and *in vivo* studies found that synaptic vesicle release in the brain can be extremely unreliable— typically 50% transmission rate and possibly as low as 10% in *in vivo*—at a given synapse (Branco and Staras, 2009; Borst, 2010). Such synaptic unreliability can be a major source of noise in the brain (Calvin and Stevens, 1968; Abbott and Regehr, 2004; Faisal et al., 2008; Yarom and Hounsgaard, 2011). Assuming neurons maximize the ratio of information theoretic channel capacity to axonal transmission energy, synaptic failures can lower energy consumption without lowering the transmitted computational information of a neuron (Levy and Baxter, 2002). Interestingly, an optimal synaptic failure rate can be computed given the energetic cost synaptic and somatic activations. Another consequence of probabilistic synapses is that, via recurrent interactions in the networks, synaptic unreliability would be the cause of Poisson-like activity in the pre-synaptic input (Moreno-Bote, 2014).

In the S2M, the multiplicative effect of the blank-out noise is manifested in the pre-synaptic input by making its variance dependent on the synaptic weights and the network states. Our theoretical analysis suggests that, in the S2M's regime of operation, increased variability has the effect of reducing the sensitivity of the neuron to other synaptic inputs by flattening the neural transfer curve. With multiplicative noise, input variability can be high when pre-synaptic neurons with strong synaptic weights are active. In the S2M, such an activity pattern emerges when the probability of a given state under the learned model and the sensory data is high. That case suggests that the network has reached a good estimate and should not be easily modified by other evidence, which is the case when the neural transfer curve is flatter. Synaptic unreliability can thus play the role of a dynamic normalization mechanism in the neuron with direct implications on probabilistic inference and action selection in the brain.

Aithchison and Latham suggested the "synaptic sampling hypothesis" whereby pre-synaptic spikes would draw samples from a distribution of synaptic weights (Aitchison and Latham, 2013). This process could be a mechanism used in the brain to represent uncertainty over the parameters of a learned model (Aitchison and Latham, 2014). Stochasticity can be carried to the learning dynamics as well. Recent studies point to that fact, when learning the blank-out probability at the synapses is also learned (Al-Shedivat et al., 2015a,b), the learned neural generative models capture richer representations, especially when labeled data is sparse. Kappel and colleagues also showed that stochasticity in the learning dynamics improves generalization capability of neural network.

The blank-out noise model used in this work is a particular case of the studies above, whereby the weights of the synapses are either zero or the value of the stored weight with a fixed probability. In contrast to previous work, we studied the learning dynamics under this probabilistic synapse model within an otherwise deterministic recurrent neural network. Besides the remarkable fact that synaptic stochasticity alone is sufficient for sampling, it enables robust learning of sparse representations and an efficient implementation in hardware.

4.1.1. Related Work on Mapping Machine Learned Models onto Neuromorphic Hardware

Many approaches for configuring spiking neural networks rely on mapping pre-trained artificial neural networks onto spiking neural networks using a firing rate code (O'Connor et al., 2013; Cao et al., 2015; Diehl et al., 2015; Hunsberger and Eliasmith, 2015). Many recent work show that the approximations incurred in brain-inspired and neuromorphic hardware platforms (O'Connor et al., 2013; Merolla et al., 2014; Neftci et al., 2014; Cao et al., 2015; Das et al., 2015; Diehl et al., 2015; Marti et al., 2015) including reduced bit precision (Marti et al., 2015; Muller and Indiveri, 2015; Stromatias et al., 2015) have a minor impact on performance. Standard artificial neural networks such as deep convolutional neural networks and recurrent neural networks trained with Rectified linear units (ReLU) can be mapped on spiking neurons by exploiting the threshold-linear of integrate and fire neurons (Cao et al., 2015; Diehl et al., 2015). Such mapping techniques have the advantage that they can leverage the capabilities of existing machine learning frameworks such as Caffe (Jia et al., 2014) or pylearn2 (Goodfellow et al., 2013) for brain-inspired computers. Although mapping techniques do not offer a solution for on-line, real-time learning, they resulted in the best performing spike-based implementations on standard machine learning benchmarks such as MNIST and CIFAR.

4.1.2. Related work on Online Learning with Spiking Neurons

Training neural networks is a very time- and energy-consuming operation, often requiring multiple days on a computing cluster to produce state-of-the-art results. Using neuromorphic architectures for learning neural networks can have significant advantages from the perspectives of scalability, power dissipation and real-time interfacing with the environment. While embedded synapses require additional chip resources and usually prohibits the implementation of more complex rules, a recent survey of software simulators argues that dedicated learning hardware is a prerequisite for real-time learning (or faster) in spiking networks (Zenke and Gerstner, 2014). This is because the speed-up margin of parallelism encounters a hard boundary due to latencies in the inter-process communications.

The S2M is an ideal candidate for *embedded*, online learning, where plasticity is implemented locally using a dedicated on-chip device (Azghadi et al., 2014). These operate on local memory

(e.g., synaptic weights) using local information (e.g., neural events) which will lead to more scalable, faster and more power efficient learning compared to computer clusters, and the ability to adapt in real-time to changing environments.

Previous work reported on on-line learning of MNIST classification with spiking neurons. A hierarchical spiking neural network with a STDP-like learning rule achieved 8% error rate on the MNIST task (Beyeler et al., 2013). Using models of non-volatile memory devices as synapses and STDP learning, error rates of 6.3% were reported (Querlioz et al., 2015). Diehl and Cook demonstrated the best results on unsupervised learning with spiking neural networks so far (Diehl and Cook, 2015). Their network model is comparable to competitive learning algorithms where each neuron learns a representation of a portion of the input space. Their architecture could achieve up to 5% error rate. The S2M outperformed this best result using a much smaller number of neurons and synapses and relying on dynamics that are more amenable to hardware implementations. However, the number of repetitions to reach this performance using the spiking S2M was larger than the above studies (512,000 presentations vs. 40,000 in Diehl and Cook, 2015). Our neural sampling based architecture with stochastic neurons and deterministic synapses achieved peak performance after 15,000 samples (Neftci et al., 2014), suggesting that the slowness is partly caused by the stochastic connections. Similar results have been observed using the DropConnect algorithm (Wan et al., 2013).

4.2. Implementations of Synaptic Unreliability in Neuromorphic Hardware

At least four studies reported the implementation of blank-out synapses for neuromorphic systems using the Address Event Representation (AER) (Goldberg et al., 2001; Choudhary et al., 2012; Corradi et al., 2014; Merolla et al., 2014). In these studies, synaptic unreliability was mainly used as a mechanism for increasing the resolution of the synaptic weights in hardware (which is often binary). In fact, the mean of a synaptic current produced by a stochastic synapse is the probability times the weight of the synapse. By allowing the probability to take values with high precision, the effective resolution of the synapse weight can be increased. The downside is that this approach is valid only when the neural computations are rate-based, such as in the neural engineering framework (Eliasmith and Anderson, 2004) where synaptic unreliability in neuromorphic systems was primarily applied (Choudhary et al., 2012; Corradi et al., 2014).

In rate-based models, the variability introduced by stochastic synapses is dealt with by averaging over large populations of neurons or by taking temporal averages. Implementations based on firing rate codes thus disregard spike times. From a hardware perspective, firing rate codes often raise the question whether a spike-based hardware platform is justifiable over a direct, dedicated implementation of the machine learning operations, or even a dedicated implementation of the rate dynamics (Wang et al., 2015). In contrast, codes based on neural sampling, synaptic sampling or phase critically depend on spike statistics or the precise timing of the spikes. For example, in the S2M, synaptic unreliability and the variability that it causes are an integral part of the sampling process. The variability introduced by the multiplicative property of synaptic noise is exploited both as a mechanism that generates sigmoidal activation and that improves learning. Results from the network dynamics suggest that the introduced variability generates sparser representations, and in some cases are insensitive to parameter rescaling. Thus, our work suggests that synaptic unreliability can play much more active roles in information processing.

The robustness and to synaptic pruning and weight down-sampling is a promising feature to further decrease the hardware footprint of S2M. However, these two features are currently introduced post-learning. Learning procedures that could introduce synaptic pruning and synaptic generation during (online) learning is the subject of future research.

AUTHOR CONTRIBUTIONS

Wrote the paper: EN, BP, SJ, MA, GC. Designed and performed the research: EN, BP, SJ, MA, GC. Analyzed the data: EN, BP, SJ.

ACKNOWLEDGMENTS

We thank Friedemenn Zenke for support on the Auryn simulator and discussion. This work funded by the National Science Foundation (NSF CCF-1317373, EN, BP, SJ, GC), the Office of Naval Research (ONR MURI N00014-13-1-0205, EN, BP, and GC), and Intel Corporation (EN, GC, BP).

REFERENCES

Abbott, L., and Regehr, W. (2004). Synaptic computation. *Nature* 431, 796–803. doi: 10.1038/nature03010

Aitchison, L., and Latham, P. (2013). "The synaptic sampling hypothesis (Abstract)," in *Computational and Systems Neuroscience, COSYNE 2013* (Salt Lake City).

Aitchison, L., and Latham, P. E. (2014). Bayesian synaptic plasticity makes predictions about plasticity experiments *in vivo*. arXiv preprint arXiv:1410.1029.

Aitchison, L., and Latham, P. E. (2015). Synaptic sampling: a connection between PSP variability and uncertainty explains neurophysiological observations. arXiv preprint arXiv:1505.04544.

Al-Shedivat, M., Naous, R., Neftci, E., Cauwenberghs, G., and Salama, K. (2015a). "Inherently stochastic spiking neurons for probabilistic neural computation," in *IEEE EMBS Conference on Neural Engineering* (Milan).

Al-Shedivat, M., Neftci, E., and Cauwenberghs, G. (2015b). "Neural generative models with stochastic synapses capture richer representations," in *Cosyne Abstracts*. Available online at: https://www.dropbox.com/s/eafcpxn1optdhiz/Al-Shedivat_etal15.pdf?dl=0

Allen, C., and Stevens, C. F. (1994). An evaluation of causes for unreliability of synaptic transmission. *Proc. Natl. Acad. Sci. U.S.A.* 91, 10380–10383.

Amit, D., and Brunel, N. (1997). Dynamics of a recurrent network of spiking neurons before and following learning. *Network Comput. Neural Syst.* 8, 373–404.

Azghadi, R., Iannella, N., Al-Sarawi, S., Indiveri, G., and Abbott, D. (2014). Spike-based synaptic plasticity in silicon: design, implementation, application, and challenges. *Proc. IEEE* 102, 717–737. doi: 10.1109/JPROC.2014.2314454

Bartol, T. M., Bromer, C., Kinney, J. P., Chirillo, M. A., Bourne, J. N., Harris, K. M., et al. (2015). Hippocampal spine head sizes are highly precise. *bioRxiv* 016329. doi: 10.1101/016329

Benjamin, B. V., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran, A. R., Bussat, J., et al. (2014). Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. *Proc. IEEE* 102, 699–716. doi: 10.1109/JPROC.2014.2313565

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., et al. (2010). "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Vol. 4.

Berkes, P., Orbán, G., Lengyel, M., and Fiser, J. (2011). Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science* 331, 83–87. doi: 10.1126/science.1195870

Beyeler, M., Dutt, N. D., and Krichmar, J. L. (2013). Categorization and decision-making in a neurobiologically plausible spiking network using a stdp-like learning rule. *Neural Netw.* 48, 109–124. doi: 10.1016/j.neunet.2013.07.012

Borst, J. G. G. (2010). The low synaptic release probability *in vivo*. *Trends Neurosci.* 33, 259–266. doi: 10.1016/j.tins.2010.03.003

Branco, T., and Staras, K. (2009). The probability of neurotransmitter release: variability and feedback control at single synapses. *Nat. Rev. Neurosci.* 10, 373–383. doi: 10.1038/nrn2634

Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *J. Comput. Neurosci.* 8, 183–208. doi: 10.1023/A:1008925309027

Brunel, N., and Hakim, V. (1999). Fast global oscillations in networks of integrate-and-fire neurons with low firing rates. *Neural Comput.* 11, 1621–1671. doi: 10.1162/089976699300016179

Calvin, W. H., and Stevens, C. F. (1968). Synaptic noise and other sources of randomness in motoneuron interspike intervals. *J. Neurophysiol.* 31, 574–587.

Cao, Y., Chen, Y., and Khosla, D. (2015). Spiking deep convolutional neural networks for energy-efficient object recognition. *Int. J. Comput. Vis.* 113, 54–66. doi: 10.1007/s11263-014-0788-3

Choudhary, S., Sloan, S., Fok, S., Neckar, A., Trautmann, E., Gao, P., et al. (2012). "Silicon neurons that compute," in *Artificial Neural Networks and Machine Learning – ICANN 2012, volume 7552 of Lecture Notes in Computer Science*, eds A. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm (Berlin: Heidelberg: Springer), 121–128.

Corradi, F., Eliasmith, C., and Indiveri, G. (2014). "Mapping arbitrary mathematical functions and dynamical systems to neuromorphic VLSI circuits for spike-based neural computation," in *IEEE International Symposium on Circuits and Systems (ISCAS)* (IEEE), 269–272.

Das, S., Pedroni, B. U., Merolla, P., Arthur, J., Cassidy, A. S., Jackson, B. L., et al. (2015). "Gibbs sampling with low-power spiking digital neurons," in *IEEE International Symposium on Circuits and Systems (ISCAS)* (IEEE), 2704–2707.

Deco, G., Jirsa, V., Robinson, P., Breakspear, M., and Friston, K. (2008). The dynamic brain: from spiking neurons to neural masses and cortical fields. *PLoS Comput. Biol.* 4:e1000092. doi: 10.1371/journal.pcbi.1000092

Diehl, P. U., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9:99. doi: 10.3389/fncom.2015.00099

Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S.-C., and Pfeiffer, M. (2015). "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *International Joint Conference on Neural Networks (IJCNN)* (IEEE), 1–8.

Eliasmith, C., and Anderson, C. (2004). *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*. Cambridge, MA: MIT Press.

Faisal, A. A., Selen, L. P., and Wolpert, D. M. (2008). Noise in the nervous system. *Nat. Rev. Neurosci.* 9, 292–303. doi: 10.1038/nrn2258

Fiser, J., Berkes, P., Orbán, G., and Lengyel, M. (2010). Statistically optimal perception and learning: from behavior to neural representations. *Trends Cogn. Sci.* 14, 119–130. doi: 10.1016/j.tics.2010.01.003

Fusi, S., and Mattia, M. (1999). Collective behavior of networks with linear (VLSI) integrate and fire neurons. *Neural Comput.* 11, 633–652. doi: 10.1162/089976699300016601

Gardiner, C. W. (2012). Handbook of stochastic methods.

Gerstner, W., and Kistler, W. (2002). *Spiking Neuron Models. Single Neurons, Populations, Plasticity*. Cambridge: Cambridge University Press.

Giulioni, M., Corradi, F., Dante, V., and del Giudice, P. (2015). Real time unsupervised learning of visual stimuli in neuromorphic VLSI systems. *Sci. Rep.* 5:14730. doi: 10.1038/srep14730

Goldberg, D., Cauwenberghs, G., and Andreou, A. (2001). Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate-and-fire neurons. *Neural Netw.* 14, 781–793. doi: 10.1016/S0893-6080(01)00057-0

Goodfellow, I. J., Warde-Farley, D., Lamblin, P., Dumoulin, V., Mirza, M., Pascanu, R., et al. (2013). Pylearn2: a machine learning research library. arXiv preprint arXiv:1308.4214.

Haykin, S. (2004). *Neural Networks: A Comprehensive Foundation, 2nd Edn.* Upper Saddle River, NJ: Prentice Hall.

Hinton, G., and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science* 313, 504–507. doi: 10.1126/science.1127647

Hinton, G., and Sejnowski, T. (1986). *Learning and Relearning in Boltzmann Machines.* Cambridge, MA: MIT Press.

Hinton, G., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural Comput.* 18, 1527–1554. doi: 10.1162/neco.2006.18.7.1527

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Comput.* 14, 1771–1800. doi: 10.1162/089976602760128018

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U.S.A.* 79, 2554–2558. doi: 10.1073/pnas.79.8.2554

Hunsberger, E., and Eliasmith, C. (2015). Spiking deep networks with lif neurons. arXiv preprint arXiv:1510.08829.

Indiveri, G., Linares-Barranco, B., Hamilton, T., van Schaik, A., Etienne-Cummings, R., Delbruck, T., et al. (2011). Neuromorphic silicon neuron circuits. *Front. Neurosci.* 5:73. doi: 10.3389/fnins.2011.00073

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. (2014). Caffe: convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093

Kappel, D., Habenschuss, S., Legenstein, R., and Maass, W. (2015). Network plasticity as bayesian inference. arXiv preprint arXiv:1504.05143.

Katz, B. (1966). *Nerve, Muscle, and Synapse*. New York, NY: McGraw-Hill.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324.

Lee, H., Ekanadham, C., and Ng, A. (2008). "Sparse deep belief net model for visual area V2," in *Advances in Neural Information Processing Systems*, Vol. 20 (Vancouver, BC), 873–880.

Levy, W. B., and Baxter, R. A. (2002). Energy-efficient neuronal computation via quantal synaptic failures. *J. Neurosci.* 22, 4746–4755.

Marti, D., Rigotti, M., Seok, M., and Fusi, S. (2015). Energy-efficient neuromorphic classifiers. arXiv preprint arXiv:1507.00235.

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673. doi: 10.1126/science.1254642

Moradi, S., and Indiveri, G. (2014). An event-based neural network architecture with an asynchronous programmable synaptic memory. *IEEE Trans. Biomed. Circ. Syst.* 8, 98–107. doi: 10.1109/TBCAS.2013.2255873

Moreno-Bote, R. (2014). Poisson-like spiking in circuits with probabilistic synapses. *PLoS Comput. Biol.* 10:e1003522. doi: 10.1371/journal.pcbi.1003522

Müller, L. (2015). *Algorithms for Massively Parallel, Event-Based Hardware*. PhD thesis, ETH Zürich.

Muller, L. K., and Indiveri, G. (2015). Rounding methods for neural networks with low resolution synaptic weights. arXiv preprint arXiv:1504.05767.

Nair, V., and Hinton, G. E. (2009). "3D object recognition with deep belief nets," in *Advances in Neural Information Processing Systems 22*, eds Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (Curran Associates, Inc.) 1339–1347. Available online at: http://papers.nips.cc/paper/3872-3d-object-recognition-with-deep-belief-nets.pdf

Nalisnick, E., Anandkumar, A., and Smyth, P. (2015). A scale mixture perspective of multiplicative noise in neural networks. arXiv preprint arXiv:1506.03208.

Neftci, E., Das, S., Pedroni, B., Kreutz-Delgado, K., and Cauwenberghs, G. (2014). Event-driven contrastive divergence for spiking neuromorphic systems. *Front. Neurosci.* 7:272. doi: 10.3389/fnins.2013.00272

O'Connor, P., Neil, D., Liu, S.-C., Delbruck, T., and Pfeiffer, M. (2013). Real-time classification and sensor fusion with a spiking deep belief network. *Front. Neurosci.* 7:178. doi: 10.3389/fnins.2013.00178

Park, J., Ha, S., Yu, T., Neftci, E., and Cauwenberghs, G. (2014). "A 65k-neuron 73-mevents/s 22-pj/event asynchronous micro-pipelined integrate-and-fire array transceiver," in *Biomedical Circuits and Systems Conference (BioCAS)*. (IEEE). Available online at: https://www.dropbox.com/s/mwex6dnytnjhew7/Park_etal14.pdf?dl=0

Parzen, E. (1999). *Stochastic Processes*, *Vol. 24*. Philadelphia, PA: SIAM.

Petrovici, M. A., Bill, J., Bytschok, I., Schemmel, J., and Meier, K. (2013). Stochastic inference with deterministic spiking neurons. arXiv preprint arXiv:1311.3211.

Pfeil, T., Potjans, T. C., Schrader, S., Potjans, W., Schemmel, J., Diesmann, M., et al. (2012). Is a 4-bit synaptic weight resolution enough? — constraints on enabling spike-timing dependent plasticity in neuromorphic hardware. *Front. Neurosci.* 6:90. doi: 10.3389/fnins.2012.00090

Probst, D., Petrovici, M. A., Bytschok, I., Bill, J., Pecevski, D., Schemmel, J., et al. (2015). Probabilistic inference in discrete spaces can be implemented into networks of lif neurons. *Front. Comput. Neurosci.* 9:13. doi: 10.3389/fncom.2015.00013

Querlioz, D., Bichler, O., Vincent, A. F., and Gamrat, C. (2015). Bioinspired programming of memory devices for implementing an inference engine. *Proc. IEEE* 103, 1398–1416. doi: 10.1109/JPROC.2015.2437616

Renart, A., Brunel, N., and Wang, X. (2003). "Mean field theory of irregularly spiking neuronal populations and working memory in recurrent cortical networks," in *Computational Neuroscience: A Comprehensive Approach* (Boca Raton, FL: Chapman and Hall), 431–490.

Rudy, J., Ding, W., Im, D. J., and Taylor, G. W. (2014). Neural network regularization via robust weight factorization. arXiv preprint arXiv:1412.6630.

Saïghi, S., Mayr, C. G., Serrano-Gotarredona, T., Schmidt, H., Lecerf, G., Tomas, J., et al. (2015). Plasticity in memristive devices for spiking neural networks. *Front. Neurosci.* 9:51. doi: 10.3389/fnins.2015.00051

Schemmel, J., Brüderle, D., Grübl, A., Hock, M., Meier, K., and Millner, S. (2010). "A wafer-scale neuromorphic hardware system for large-scale neural modeling," In *International Symposium on Circuits and Systems, ISCAS 2010* (IEEE), 1947–1950.

Stromatias, E., Neil, D., Pfeiffer, M., Galluppi, F., Furber, S. B., and Liu, S.-C. (2015). Robustness of spiking deep belief networks to noise and reduced bit precision of neuro-inspired hardware platforms. *Front. Neurosci.* 9:222. doi: 10.3389/fnins.2015.00222

Tuckwell, H. C. (2005). *Introduction to Theoretical Neurobiology: Nonlinear and Stochastic Theories*, *Vol. 2*. Cambridge: Cambridge University Press.

van Vreeswijk, C., and Sompolinsky, H. (1996). Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science* 274, 1724–1726.

Wan, L., Zeiler, M., Zhang, S., Cun, Y. L., and Fergus, R. (2013). "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* (Atlanta, GA), 1058–1066.

Wang, R., Thakur, C. S., Hamilton, T. J., Tapson, J., and van Schaik, A. (2015). A neuromorphic hardware architecture using the neural engineering framework for pattern recognition. arXiv preprint arXiv:1507.05695.

Wang, X. (1999). Synaptic basis of cortical persistent activity: the importance of NMDA receptors to working memory. *J. Neurosci.* 19, 9587–9603.

Yarom, Y., and Hounsgaard, J. (2011). Voltage fluctuations in neurons: signal or noise? *Physiol. Rev.* 91, 917–929. doi: 10.1152/physrev.00019.2010

Yu, S., Gao, B., Fang, Z., Yu, H., Kang, J., and Wong, H.-S. P. (2013). Stochastic learning in oxide binary synaptic device for neuromorphic computing. *Front. Neurosci.* 7:186. doi: 10.3389/fnins.2013.00186

Zenke, F., and Gerstner, W. (2014). Limits to high-speed simulations of spiking neural networks using general-purpose computers. *Front. Neuroinform.* 8:76. doi: 10.3389/fninf.2014.00076

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

APPENDIX

TABLE A1 | Parameters used for the neural networks.

σ	Noise amplitude	spiking S2M, visible neurons	4.47 nA
p	Blank-out probability at synapse	All models	0.5
τ_r	Refractory period	All spiking S2M	4 ms
τ_{syn}	Time constant of recurrent, and bias synapses	All spiking S2M	4 ms
τ_{br}	"Burn-in" time of the neural sampling	All spiking S2M	10 ms
g_L	Leak conductance	All spiking S2M	1 nS
u_{rst}	Reset Potential	All spiking S2M	0 V
C	Membrane capacitance	All spiking S2M	1 pF
θ	Firing threshold	All spiking S2M	100 mV
$2T$	Epoch duration	All spiking S2M	100 ms
W	Initial weight matrix	All spiking S2M	$N(0, 0.3)$
		All S2M, RBM	$N(0, 0.1)$
b_v, b_h	Initial bias for layer v and h	All spiking S2M	-0.15
		All S2M, RBM	0
T_{sim}	Simulation time per epoch	All spiking S2M	100 s
N_v, N_h	Number of visible and hidden units	All models,	794, 500
		Except in **Figure 2**	5, 5
N_c	Number of class label units	All models	10
$N_{samples}$	Total number of MNIST sample presentations	**Figure 4**, spiking S2M	512000
		Figure 4, S2M, RBM	$75 \cdot 10^6$
τ_{STDP}	Learning time window	All models	10 ms
ϵq	Learning rate for W	All spiking S2M	$3.85 \cdot 10^{-6}$
ϵ		All S2M, RBM	0.025
ϵ_b	Learning rate for b_v, b_h	All spiking S2M	$1.43 \cdot 10^{-5}$
		All S2M, RBM	.025
n_{batch}	Batch size	All S2M, RBM	50
W	Distribution of weight parameters	**Figure 2**	$N(-0.3, 1.5)$
b_h, b_v	Distribution of bias parameters	**Figure 2**	$N(0, 1.5)$