Biologically Plausible Sequence Learning with Spiking Neural Networks

Zuozhu Liu,[1,3] **Thiparat Chotibut,**[2,3,*] **Christopher Hillar,**[4,5] **Shaowei Lin**[3]
[1]Department of Statistics and Applied Probability, National University of Singapore,
[2]Department of Physics, Faculty of Science, Chulalongkorn University, Thailand,
[3]Engineering Systems and Design, Singapore University of Technology and Design, [4]Awecom, Inc,
[5]Redwood Center for Theoretical Neuroscience, University of California, Berkeley,
{lcowen.hn, *thiparatc, hillarmath}@gmail.com, shaowei_lin@sutd.edu.sg

Abstract

Motivated by the celebrated discrete-time model of nervous activity outlined by McCulloch and Pitts in 1943, we propose a novel continuous-time model, the McCulloch-Pitts network (MPN), for sequence learning in spiking neural networks. Our model has a local learning rule, such that the synaptic weight updates depend only on the information directly accessible by the synapse. By exploiting *asymmetry* in the connections between binary neurons, we show that MPN can be trained to robustly memorize multiple *spatiotemporal* patterns of binary vectors, generalizing the ability of the symmetric Hopfield network to memorize static spatial patterns. In addition, we demonstrate that the model can efficiently learn sequences of binary pictures as well as generative models for experimental neural spike-train data. Our learning rule is consistent with spike-timing-dependent plasticity (STDP), thus providing a theoretical ground for the systematic design of biologically inspired networks with large and robust long-range sequence storage capacity.

Introduction

Experimental evidence from neurobiology reveals that changes in synaptic weights of some neurons depend on the timing difference between presynaptic and postsynaptic spikes (Markram and Sakmann 1995; Gerstner et al. 1996; Bi and Poo 2001), a concept termed *spike-timing-dependent plasticity* (STDP). While deep learning is tremendously successful in numerous machine learning tasks, its underlying backpropagation learning algorithm is biologically implausible (Bengio et al. 2015). Recent works attempt to bridge this gap by proposing STDP-consistent training rules for deep learning (Bengio et al. 2015; 2017; Liu, Quek, and Lin 2018; Scellier and Bengio 2017), with the hope of uncovering computationally efficient learning algorithms inspired by biological brains. Thorough understanding of STDP-consistent learning rules and their computing capabilities can also benefit the design of modern bioinspired computing hardware such as neuromorphic chips.

On the other hand, machine learning can assist in the quest for understanding how brains perform computation, especially when viewed through the lens of a spiking neural network (SNN). For example, variational inference and reinforcement learning unveil SNN architectures that perform probabilistic inference consistent with STDP rules (Nessler et al. 2013; Pecevski and Maass 2016; Rezende, Wierstra, and Gerstner 2011). Learning-to-learn methods and Long Short-Term Memory help identify a novel SNN architecture with the capability to store long spatiotemporal sequences of spikes during computation (Bellec et al. 2018). Moreover, building generative models of SNNs from experimental time-series data of action potentials yields insights into the network structure generating such spikes (Tyrcha et al. 2013; Nasser, Marre, and Cessac 2013; Zeng et al. 2013).

With the goal of building a biologically plausible generative model of SNNs from spatiotemporal patterns, such as neural spike-train data, we propose a novel continuous-time learning algorithm that is consistent with experimentally observed STDP. We define simple rules for updating states and learning weights of a continuous-time SNN and demonstrate that the expected synaptic changes given a pair of presynaptic and postsynaptic spikes reproduce the STDP curve. As a consequence of our model, we show that the stochasticity of the refractory period is an important ingredient for reproducing STDP. Namely, by assuming that the time between the spiking of a neuron and its recovery is stochastic and follows an exponential distribution, we prove that the expected increase or decrease in synaptic weights follows STDP-like curves.

In addition to its biological plausibility, our SNN model is capable of learning sequences of binary vectors, or spatiotemporal patterns of spikes. In the deterministic limit, this network, a McCulloch-Pitts network (McCulloch and Pitts 1943), is a generalization of the Hopfield network that allows *asymmetric* connections between fully-visible binary units. As a result, deterministic dynamics in the state space can converge to an attractor of period larger than one (a cycle), as opposed to a fixed-point corresponding to a static spatial memory in the Hopfield network. We demonstrate that the network can be trained to memorize repeating sequences of binary vectors and that the spatiotemporal memory is robust to perturbation. In particular, given an arbitrary initial state of the spike pattern, the dynamics generated by the trained network converge to a memorized cycle. Thus, the

*Correspondence to: thiparatc@gmail.com

STDP-consistent McCulloch-Pitts network performs a spatiotemporal pattern completion, generalizing the classical Hopfield network that performs spatial pattern completion. In the noisy limit, the trained network can be regarded as a generative model for spatiotemporal patterns of spikes that performs sampling of both the time to the next spiking event and the neuron to next spike. We illustrate that this network efficiently performs inference on continuous-time data from a neurobiology experiment, and we discuss its applicability to inference problems in neuroscience.

McCulloch-Pitts Networks

To begin, suppose we are presented with a time series $\mathcal{D}(t)$ of binary vectors with single bit-flip transitions:

$$\begin{aligned}
\mathcal{D}(t) &= x^{(0)} && \text{for } 0 = t_0 \le t < t_1, \\
&\vdots \\
\mathcal{D}(t) &= x^{(N-1)} && \text{for } t_{N-1} \le t < t_N, \\
\mathcal{D}(t) &= x^{(N)} && \text{for } t = t_N.
\end{aligned}$$

Our goal is to train a continuous-time stochastic process that mimics $\mathcal{D}(t)$.

In this section, we define our model of a neural stochastic process, which we term a McCulloch-Pitts network. Relationships to biological neural networks will be discussed in the Spike-Timing-Dependent Plasticity section. Let $G = (V, E)$ be a weighted directed graph with vertices $V = \{1, \ldots, d\}$, edges $ij \in E$ from i to j, vertex biases $b_1, \ldots, b_d \in \mathbb{R}$, and weights $w_{ij} \in \mathbb{R}$ for each edge $ij \in E$. Let $\theta = (b_i, w_{ij})$ denote the vector of parameters. Self-loops and directed cycles are allowed in G.

Dynamics

Let $\mathcal{X} = \{0,1\}^d$ be the state space of binary vectors on V, and let $X(t)$ be a stochastic process indexed by continuous time $t \ge 0$ with states $\mathcal{X}$. Let $\tau > 0$ be the *temperature*[1], and define $\mathrm{flip}_i(x)$ to be the binary vector x whose i-th coordinate is flipped. To generate $X(t)$, we select an initial condition $X(0)$ from $\mathcal{X}$, and repeat the following two steps:

1. Suppose $X(T) = x$. For each $i \in V$, we sample a *holding time* ξ_i from the exponential distribution $\mathrm{Exp}(\lambda_i)$ where $\lambda_i = \exp(\sigma_i z_i/\tau)$, $\sigma_i = 1 - 2x_i$, and $z_i = \sum_{ji \in E} w_{ji} x_j + b_i$.
2. Let ξ_j be the smallest of the holding times ξ_i, and let $T' = T + \xi_j$. We set $X(t) = x$ for all $T \le t < T'$, and $X(T') = \mathrm{flip}_j(x)$. Finally, we update $T \leftarrow T'$.

More generally, a rate scaling parameter r_i could be introduced in the rate $\lambda_i = r_i \exp(\sigma_i z_i/\tau)$. However, r_i can be absorbed into b_i and w_{ii}, see Supplementary Materials (SM) A. Thus, we can set $r_i = 1$ for simplicity and assume the graph G has self-loops. We term $X(t)$ the *McCulloch-Pitts network* (MPN) associated to (G, θ, τ). Observe that the transitions in $X(t)$ involve flipping only one bit at a time.

Alternatively, we can reformulate an equivalent yet simpler algorithm to generate $X(t)$ using softmax; given an initial condition $X(0) \in \mathcal{X}$, we repeat the following steps:

1. Let $X(T) = x$. Sample $\xi \sim \mathrm{Exp}(\lambda)$ where $\lambda = \sum_{i \in V} \lambda_i$.
2. Let $T' = T + \xi$ and set $X(t) = x$ for all $T \le t < T'$. Pick $j \in V$ according to $P_i = \lambda_i/\lambda$, and set $X(T') = \mathrm{flip}_j(x)$. Finally, we update $T \leftarrow T'$.

The multinomial distribution $P_i = \lambda_i/\lambda$ in step 2 is the softmax, since $P_i \propto \exp(\sigma_i z_i/\tau)$.

Training

For brevity, let us define the following data statistics:

$$\delta_i^{(n)} = x_i^{(n+1)} - x_i^{(n)},\ \sigma_i^{(n)} = 1 - 2x_i^{(n)},$$
$$z_i^{(n)} = \textstyle\sum_{ji \in E} w_{ji} x_j^{(n)} + b_i,$$
$$\lambda_i^{(n)} = \exp\left(\sigma_i^{(n)} z_i^{(n)}/\tau\right),\ \lambda^{(n)} = \sum_i \lambda_i^{(n)}.$$

The log-conditional likelihood, $\mathcal{L}_{\mathcal{D}}(\theta)$, measuring similarity between model distribution $X(t)$ and $\mathcal{D}(t)$, amounts to:

$$\begin{aligned}
&\log p\Big(X(t) = \mathcal{D}(t)\,, 0 < t \le t_{N+1} \mid X(0) = x^{(0)}\Big) = \\
&\sum_{n=0}^{N-1} \log p\Big(X(t) = \mathcal{D}(t)\,, t_n < t \le t_{n+1} \mid X(t_n) = x^{(n)}\Big) \\
&= \sum_{n=0}^{N-1} \log\left\{\frac{\prod_i (\lambda_i^{(n)})^{\left|\delta_i^{(n)}\right|}}{\lambda^{(n)}} \lambda^{(n)} \exp\left[(t_n - t_{n+1})\lambda^{(n)}\right]\right\} \\
&= \sum_{n=0}^{N-1} \left\{\mathcal{T}_n(\theta) + \mathcal{H}_n(\theta)\right\},
\end{aligned}$$

with *transition terms* $\mathcal{T}_n(\theta)$ and *holding terms* $\mathcal{H}_n(\theta)$:

$$\mathcal{T}_n(\theta) = \textstyle\sum_{i=1}^d \delta_i^{(n)} z_i^{(n)}/\tau,$$
$$\mathcal{H}_n(\theta) = -(t_{n+1} - t_n) \textstyle\sum_{i=1}^d \lambda_i^{(n)}.$$

Since $-\mathcal{L}_{\mathcal{D}}(\theta)$ is convex (it consists of sums and exponentiations of linear/convex functions), we will adopt gradient methods for parameter optimization. Straightforward calculation yields the gradients of $\mathcal{T}_n(\theta)$ and $\mathcal{H}_n(\theta)$:

$$\begin{aligned}
\frac{\partial \mathcal{T}_n}{\partial w_{jk}} &= \frac{1}{\tau} x_j^{(n)} \delta_k^{(n)}, \frac{\partial \mathcal{T}_n}{\partial b_k} = \frac{1}{\tau} \delta_k^{(n)}, \\
\frac{\partial \mathcal{H}_n}{\partial w_{jk}} &= -\frac{1}{\tau} x_j^{(n)} \sigma_k^{(n)} \lambda_k^{(n)} (t_{n+1} - t_n), \\
\frac{\partial \mathcal{H}_n}{\partial b_k} &= -\frac{1}{\tau} \sigma_k^{(n)} \lambda_k^{(n)} (t_{n+1} - t_n).
\end{aligned}$$

Note that these rules are highly local, in the sense that the update for a synaptic weight w_{ik} depends only on the presynaptic state x_i, the postsynaptic rate λ_k, and the postsynaptic

[1] Temperature can be interpreted as a proper physical temperature in the Boltzmann machine since its symmetric weights uniquely determine states' energy function, and its thermodynamics formulation can be adopted. Here, in *asymmetric* networks, temperature reflects the frequency in which stochastic updates take place (*intrinsic noise*). Zero temperature thus corresponds to deterministic dynamics, whereas high temperature drives rapidly fluctuating stochastic dynamics.

transition δ_k. Similarly, the update for the bias b_k depends only on the rate λ_k and transition δ_k. Moreover, any parameter update that agrees in sign with the above gradients will increase both the transition and holding terms of the log-conditional likelihood. Therefore, the training algorithm is robust to noise (up to sign) in the parameter updates. See SM C for the training algorithm pseudocode.

Spike-Timing-Dependent Plasticity

Recent work in neuroscience (Ermentrout and Terman 2010) has shown that individual neurons often do not behave as single spiking units. In fact, different compartments of a neuron, such as the apical and basal regions of a pyramidal neuron, can spike apart from the soma. To model such neural dynamics with a McCulloch-Pitts network, we represent the compartments of a neuron by the vertices of the graph G, and refer to them as *units* rather than *neurons* to avoid confusion.

We assume that each spiking unit i is either in an *armed* state $x_i = 0$, capable of spiking, or in a *refractory* state $x_i = 1$, representing a unit that just spiked and is incapable of spiking again until it recovers. We assume that a unit influences the spiking of a neighboring unit if and only if it is in the refractory state, a reasonable condition also assumed in (Pecevski and Maass 2016). In addition, to account for a well-known observation that the transmission of spikes across synapses is unreliable (Faisal, Selen, and Wolpert 2008), we model the random chance of spiking events that depend on synaptic strengths w_{ij} and unit biases b_i as stochastic processes. Specifically, the wait time to the *next spiking event* of unit i is assumed to be exponentially distributed with rate

$$\lambda_i(\theta) = r_i \exp\left\{\sigma_i z_i/\tau\right\},$$

where

$$\sigma_i = 1 - 2x_i, z_i = \sum_{ji\in E} w_{ji}x_j + b_i,$$

and where the parameter r_i controls the background spiking rate of the unit when z_i vanishes. Note that when the unit is armed, the expected time to fire exponentially decreases with z_i; it is thus tempting to call z_i the "membrane potential" because of its similarity to classical integrate-and-fire neurons. However, spike events are *not* stochastic when the membrane potential is known (Zador 1998). To avoid confusion, we shall therefore call z_i the *unsigned activity* and $\sigma_i z_i$ the *signed activity*. Note also that while most neuron models assume a fixed refractory period, in our model it is stochastic, given by an exponential distribution whose expected time to recovery is inversely proportional to the exponential of z_i.

Learning in the network is implemented by two sets of update rules. Let η be the learning rate. When a spike occurs, we apply transition updates:

$$\Delta w_{jk} = \frac{\eta}{\tau} x_j^{(n)} \delta_k^{(n)}, \quad \Delta b_k = \frac{\eta}{\tau}\delta_k^{(n)}. \tag{1}$$

On the other hand, when there are no spikes over a period of time, we apply holding updates:

$$\begin{aligned} \Delta w_{jk} &= -\frac{\eta}{\tau} x_j^{(n)} \sigma_k^{(n)} \lambda_k^{(n)} (t_{n+1} - t_n), \\ \Delta b_k &= -\frac{\eta}{\tau} \sigma_k^{(n)} \lambda_k^{(n)} (t_{n+1} - t_n). \end{aligned} \tag{2}$$

We may also interpret the holding updates as a fixed-rate decay of the weights and biases over time.

In Bi and Poo's experiments (Bi and Poo 1998) on spike-timing-dependent plasticity (STDP), the behavior of a pair of interconnected neurons, in which the synaptic weight is not strong enough for a presynaptic spike to cause a postsynaptic spike, is studied. The presynaptic neuron is initially stimulated to spike, and the amplitude of the excitatory post-synaptic current (EPSC) is measured. After five minutes, the presynaptic neuron is stimulated again to spike. Next, after ε time, the postsynapse is stimulated to spike. This pair of stimulations is repeated every second for 60 seconds. Finally, after twenty minutes, the amplitude of the EPSC is measured again and the percentage change is recorded.

We prove that the average case behavior of the learning algorithm for an MPN agrees with the experimentally-observed synaptic potentiation and depression in biological neural networks. To simplify the mathematical analysis, we shall make the following assumptions:

1. Consider a synapse with presynaptic and postsynaptic neurons i, j with states x_i, x_j. Let w be the synaptic weight (directed weight from i to j), and b_i, b_j the neural biases. Let $z_i = b_i, z_j = wx_i + b_j$ be the unsigned activities. Let $\lambda_i = r_i \exp(\sigma_i z_i/\tau), \lambda_j = r_j \exp(\sigma_j z_j/\tau)$ be the firing rates, where we set the background rate $r_i = r_j = r$, and τ is the temperature controlling the degree of stochasticity.
2. The neurons do not spike unless they are manually stimulated, but they may recover from their refractory state on their own accord.
3. The weight w is updated according to equations (1) and (2), but the biases b_i, b_j are fixed.
4. The ratio η/τ is much smaller than 1.
5. The refractory periods are on average shorter than the armed periods.

Theorem 0.1. *Assuming the above conditions, let ε be the timing of the presynaptic spike subtracted from that of the postsynaptic spike. If ε is small and positive, then the expected synaptic weight change is*

$$\mathbb{E}[\Delta w] \approx C_1 e^{-\lambda_i|\varepsilon|}, \quad \lambda_i = re^{-b_i/\tau},$$

but if ε is small and negative, then the expected synaptic weight change is

$$\mathbb{E}[\Delta w] \approx -C_2 e^{-\lambda_j|\varepsilon|}, \quad \lambda_j = re^{-(w+b_j)/\tau},$$

where C_1, C_2 are positive constants.

Proof. See SM B. □

Experiments

In this section, we showcase the biologically plausible features and capabilities of the MPN.[2] We first numerically verify the prediction of Theorem 0.1, reproducing STDP-like curves from our learning rule. The MPN is then shown to be self-consistent, i.e. accurate inference of its own generative model is achievable. We next demonstrate the ability of MPN to robustly memorize spatiotemporal patterns. Namely, we

[2]The codes are available at https://github.com/owen94/MPNets.

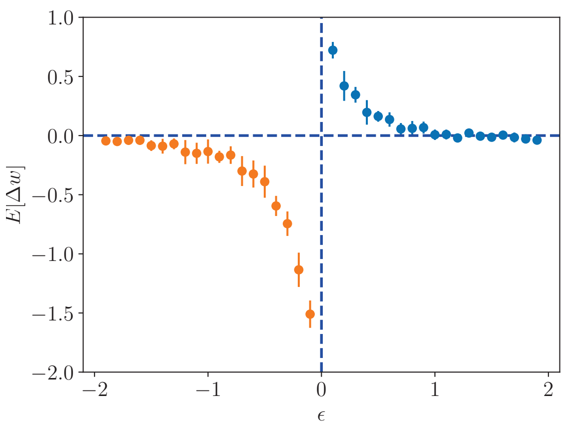

Figure 1: STDP-like curves emerge from simulating an MPN on a pair of neurons, in agreement with Theorem 0.1. We denote the synaptic weight change as $\Delta w = \hat{w}_{ij} - w_{ij}$, and $\mathbb{E}(\Delta w)$ is computed from averaging 10 trials. Vertical dashed lines at each data point are error bars.

show that our model can be trained to memorize repeating sequences of binary vectors with one-hop transitions and that these memories are robust in the sense that they are stable attractors of the MPN dynamics in the deterministic limit. When the dynamics are stochastic, we show that stochasticity assists in multiple-memory switching, as opposed to the deterministic counterpart whose dynamics eventually fixes at only one memory. Finally, we end this section by highlighting two potentially useful sequence learning applications: memorizing a long sequence of binary pictures (shown in Fig. 4 and the video in SM) and learning a generative model of experimental neural spike-train data. The latter demonstrates that MPNs can effectively reproduce spike-timing statistics in neurobiology experiments.

STDP and MPN Learning Rule

By simulating the dynamics of a presynaptic neuron i and a postsynaptic neuron j, we now verify the emergence of STDP curves from our MPN learning rule. Following the algorithm discussed in the Spike-Timing-Dependent Plasticity section, the expected synaptic weight changes Δw_{ij} can be measured as a function of the spike timing difference ϵ.

We simulate 10 trials for each ϵ, while for each trial the updates of 60 consecutive spikes are accumulated, as suggested in (Bi and Poo 1998), to finally compute the new synaptic weight $\hat{w}_{ij}$. When there are no longer weight updates for every spike, the update will be terminated. Here, we set the learning rate for the transition and holding updates to $\eta_{\mathcal{T}} = 0.05$ and $\eta_{\mathcal{H}} = 0.001$, respectively. Other parameters are $w_{ij} = 1, b_i = b_j = 0$.

Two initial conditions are considered: either neuron i is in the refractory state and neuron j is in the armed state or vice versa. Fig.1 displays simulation results for both initial conditions, which reveals that the expected (average) weight change consists of two exponential branches as a function of

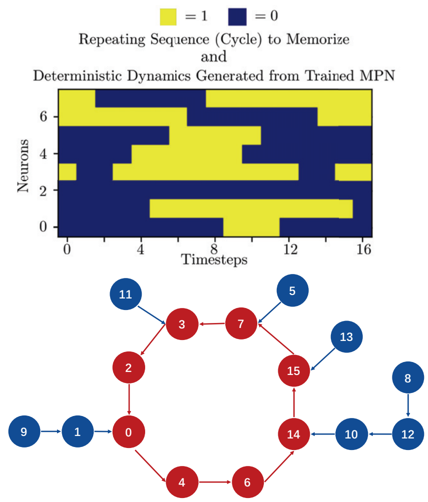

Figure 2: Deterministic dynamics of a trained MPN robustly memorize spatiotemporal binary patterns. For an MPN with 8 neurons, the cycle of period 16 to be memorized can be perfectly reproduced from the deterministic dynamics generated by the trained MPN **(top)**. In addition, a spatiotemporal memory is *robust*, as illustrated in the smaller state space dynamics generated by another trained MPN with 4 neurons **(bottom)**. The number on each node is the decimal representation of the corresponding state binary vector of length 4; e.g., $(1, 0, 1, 0)$ is 5. Blue nodes are transient states that flow along the arrow toward the robust spatiotemporal memory, i.e. stable periodic attractor (red).

ϵ, in agreement with Theorem 0.1. This result is consistent with the STDP rule discussed in (Bi and Poo 1998), and the two branches do not need to be symmetric as assumed in (Scellier and Bengio 2017).

Consistency Check: Here, we numerically show that MPNs can consistently learn the parameters of another MPN that generates training samples. 100,000 samples were generated from the pre-defined MPN of 10 neurons with weights w and biases b. Next, we train an MPN to fit the generated data. For the reconstruction error using ℓ_1 norm defined by $error(w, \hat{w}) = \frac{1}{100}\|w - \hat{w}\|$, we find $error(w, \hat{w}) = 0.03$ after 30 epochs of training. This consistency check shows that MPN is self-consistent. Note also that the training algorithm converges fast, plateauing after 5 training epochs (see SM D.1).

Robust Spatiotemporal Memory

Spatiotemporal memory: The advantage of weight asymmetry is the flexibility to memorize repeating *sequences* of spikes (*cycles*) regarded as spatiotemporal memories. Here, we show that MPN can be trained to memorize a cycle using the deterministic update rule (the noiseless limit is identical to setting $\tau = 0$). This update rule amounts to flipping the neuron i with the highest rate, i.e., $i = \arg\max \lambda_j$. Fig.2 (top) shows the cycle of period 16 to memorize, which is the repeating sequence to be stored by an MPN with 8 neurons. After training with this sequence, MPN can successfully gen-

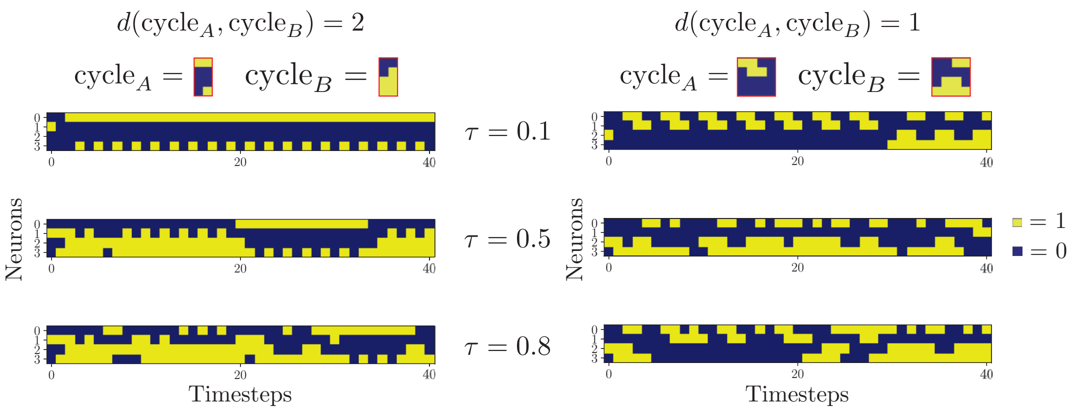

Figure 3: 40 states transitions under different noise parameter τ generated by an MPN with 4 neurons trained to memorize two cycles in the deterministic limit: **(left)** stochastic dynamics generated by an MPN that is trained to memorize cycles of period 2. **(right)** stochastic dynamics generated by an MPN that is trained to memorize cycles of period 4; see Fig.2 in SM for topologies. $d(\text{cycle}_A, \text{cycle}_B)$ denotes the minimal hamming distance between the two cycles, which measures how far away the stored spatiotemporal memories are. The larger the d, the more rare bit flips (induced by intrinsic noise) are required to transit to the other memory, and hence memory switching becomes less likely when d is large. At the same τ, we thus observe less frequent memory switching in **(left)** than in **(right)**.

erate the same sequence (reproducing Fig.2 (top)) from the dynamics in the limit $\tau = 0$, provided the initial states of the spikes are identical. Hence, MPN can store the sequence as a spatiotemporal memory.

Robustness: This repeating sequence memorized by the trained MPN with 8 neurons is in fact the stable attractor of the entire state space dynamics generated by the trained MPN. Namely, any $2^8 = 256$ initial binary states will eventually flow toward the cycle in Fig.2 (top). Thus, this spatiotemporal memory is *robust*. Fig.2 (bottom) illustrates robust spatiotemporal memory in the state space dynamics generated by the trained MPN with 4 neurons, where the entire state space consists of $2^4 = 16$ elements. All transient states (blue) are in the basin of attraction of the spatiotemporal memory (red), which is the repeating sequence $[0, 4, 6, 14, 15, 7, 3, 2, 0]$.

Multiple Memories and the Role of Noise: It is possible to train an MPN to store multiple spatiotemporal repeating patterns, analogous to standard Hopfield nets that can store multiple static patterns. Here, we demonstrate the simplest scenario where 2 non-overlapping cycles are stored as robust spatiotemporal memories in a small network with 4 neurons. Generalization to multi-cycle storage in larger networks is straightforward and will be reported elsewhere.

In these trainings, the first cycle to be memorized, say cycle_A, is selected to train an MPN. After the training is completed, we select another one, cycle_B, that is disjoint from cycle_A, and train the MPN that was originally trained on cycle_A. Eventually, the MPN can store *both* cycle_A and cycle_B. We illustrate two trained MPNs: one that stores 2 cycles of period 2, and another one that stores 2 cycles of period 4. The deterministic state transition diagram generated by these MPNs is shown in Fig.3; the state space consists of two cycles of period 2 in Fig.3(a) which are $\text{cycle}_A = \{(1,0,0,0),(1,0,0,1)\}$ and $\text{cycle}_B = \{(0,0,1,1),(0,1,1,1)\}$, and two cycles of period 4 in Fig.3(b), which are $\text{cycle}_A = \{(1,0,0,0),(1,1,0,0),(0,1,0,0),(0,0,0,0)\}$ and $\text{cycle}_B = \{(0,0,0,1),(0,0,1,1),(0,1,0,0),(0,1,1,0)\}$. This notation represents an ordered set, in which a sequence generated by the trained MPN proceeds from left to right.

In the presence of noise ($\tau > 0$), fluctuations may destroy the robustness of cycles. Namely, rather than remaining in one of the cycles (deterministically stable periodic attractors), states could be driven away from one robust cycle to another through a series of rare one flip transitions. Fig. 3 shows the response of the trained MPNs as the strength of intrinsic noise τ increases. Thus, one useful role of noise in MPN is to facilitate transitions between stored spatiotemporal patterns of spikes that would not otherwise be possible in the deterministic limit. Whether one could exploit this intrinsic noise-induced memory switching phenomena to design controllable memory switching networks is worthy of future investigation.

Application: Robustly Memorizing Sequences of Binary Pictures

Robust spatiotemporal memory is an appealing feature of the MPN; however, successive elements of a preferable sequence to store might not be one-hop different as assumed in the previous section. For instance, the successive elements in the sequence of digits "2019" represented as binary pictures of Fig. 5 (left) differ by multi-hop transitions. Here, we show that MPNs can be trained to learn sequences of pictures that differ by multi-hop transitions, illustrated in Figs.4- 5, and also in the video in SM.

The multi-hop away sequence to learn consists of [the all-zero state, "2", "0", "1", "9"], in this specific order. Each static binary picture in the sequence is represented as a 20×20 binary matrix. To transit among these static memories in the correct order, we randomly assign 100 one-hop transitions

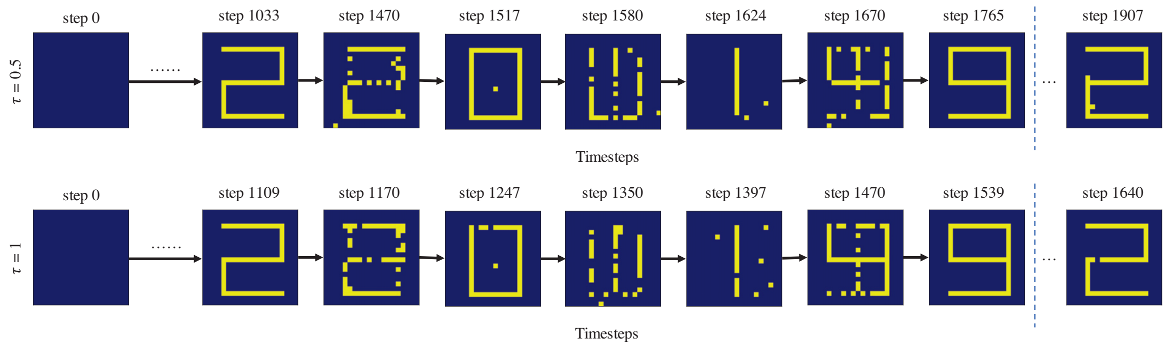

Figure 4: Spatiotemporal pattern completion: dynamics generated by the trained MPN with $\tau = 0.5$ and $\tau = 1$ can robustly recover the multi-hop away memory sequence 2019. Results for $\tau = 2$ can be found in the SM.

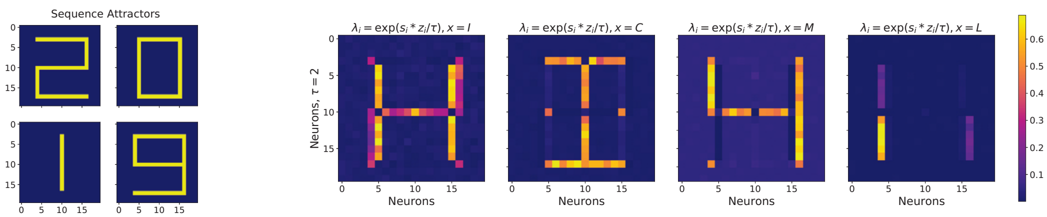

Figure 5: **(Left)**. Multi-hop memory sequence to store that proceeds from "2" to "0" to "1" and to "9". **(Right)**. Rates λ_i computed for different input spiked patterns; left to right: "2", "0","1" and "9". Larger λ_i indicates a higher probability for the neuron i to flip.

between every successive pair of the static memories; i.e., the length for the entire training sequence, including the transitions from the all-zero state to "2", "2" to "0", "0" to "1" and "1" to "9", is 400. The learning rate is set to $\eta = 0.001$. For details on training procedure, see SM D.3.

Fig.4 shows snapshots of the dynamics at two different noise parameters τ generated from the trained MPN. Initialized with the all-zero state, the trained MPN almost perfectly reproduces the multi-hop sequence "2019" with multiple one-hop transitions that connect consecutive elements.

This generated sequence is also robust to noise, in the sense that we can recognize the sequence of static memories "2", "0", "1", and "9" as time progresses for $\tau = 0.5, 1$ and 2. As a consequence of our training procedure, the trained MPN will generate the cycles containing multi-hop away memories in the correct order; i.e., the dynamics will loop "2" $\rightarrow \cdots \rightarrow$ "0" $\rightarrow \cdots \rightarrow$ "1" $\rightarrow \cdots \rightarrow$ "9" $\rightarrow \cdots \rightarrow$ "2" $\rightarrow \cdots$ (see Fig.4).

We further examine the rate $\lambda_i(\theta) = r_i \exp\{\sigma_i z_i/\tau\}$ to illustrate that the MPN indeed learned the sequence. Given the weights and biases of the trained MPN, we compute the rates when the static memories are the input. Since the neuron with a higher rate λ_i will be more likely to flip next, we can estimate the preferable transitions that will take place given λ_i. For instance, when feeding the trained MPN with the input "2", most rates associated with the spiked neurons of the memory "0" and "2" become significantly higher than the rates at the other neurons, see Fig.5. This is consistent with the fact that "2" will transit to "0" through multi-hop transitions and that the spiked neurons of "2" will recover with higher rates than those neurons remaining in the armed state. However, the neurons that spike at *both* the memory "2" and "0", i.e. the bottom horizontal line, will prefer to not flip, consistent with the almost negligible rates at the neurons that both memories already share.

Note finally that while Hopfield networks can be used for recalling or denoising static memories (Hopfield 1982; Hillar and Tran 2018), we have shown that MPNs can be applied to recall spatiotemporal memories or complete spatiotemporal patterns. Thus, the MPN with asymmetric weights can be regarded as an STDP-consistent and non-equilibrium generalization of the dynamics arising from the symmetric Hopfield network. Regarding weight asymmetry, we observe from the trained MPN that the learned weights are considerably asymmetric. The asymmetry measure $|w_{ij} - w_{ji}|/((|w_{ij}| + |w_{ji}|)/2) = 1.38, 1, 53, 1.58$ for training parameters $\tau = 0.5, 1, 2$, respectively.

Application: Inferring Generative Models for Neural Spike-train Data

We now apply the MPN to learn spike-train statistics of a neurobiology experiment, which reports the neuronal activities of cat primary visual cortex [3]. The experiment recorded spike-train dataset of 25 neurons. We preprocess the data and

[3] The dataset is available on https://crcns.org/data-sets/vc/pvc-3

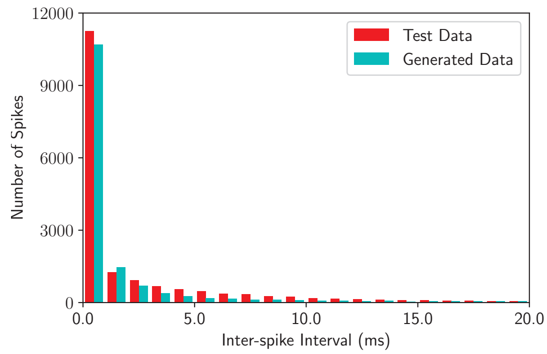

Figure 6: Histogram of the ISI of the spikes generated from the trained MPN is in excellent agreement with that of the test data.

form 60288 one-hop transitions, of which 70% (42202) is used for training and 30% (18086) for testing. The learning rate is set to $\eta = 0.01$.

To see how well the MPN reproduces the spike-timing statistics of the data, we generate spike trains (from the MPN trained at $\tau = 1$) that last for the same duration as that of the test data, starting from the same initial spike state. Fig.6 shows that the distribution of the inter-spike intervals (ISI) of the generated spikes is in good agreement with that of the test data. The visiting frequencies of generated spike states are also in agreement with that of the test data as well (see Fig.6 in SM D.4).

We now compare the performance of MPN in reproducing the ISI statistics to that of the generalized linear model (GLM), a standard statistical model of ISI, using the Kullback-Leibler(KL) divergence as a performance metric. We first bin the ISI into 20 slots, compute the density of each bin, and obtain the distributions $P_{test}(ISI), P_{MPN}(ISI), P_{GLM}(ISI)$. Then, the KL-divergences are calculated from $KL(MPN) = KL(P_{MPN}(ISI) \| P_{test}(ISI))$ and $KL(GLM) = KL(P_{GLM}(ISI) \| P_{test}(ISI))$, where $KL(p\|q) = -\sum_x p(x)\log(p(x)/q(x))$. Remarkably, we found that $0.21 \pm 0.11 = KL(GLM) > KL(MPN) = 0.04 \pm 0.003$, where the standard deviations are calculated from 10 trials. Thus, MPNs can capture the statistics of ISI even more accurately than the ubiquitous GLM.

Related Work

While sequence learning tasks have been studied in the literature (Amari 1972; Brea, Senn, and Pfister 2013; Memmesheimer et al. 2014), the work most related to MPN is (Brea, Senn, and Pfister 2013), which focuses on training spiking neural networks with biologically plausible rules. While both (Brea, Senn, and Pfister 2013) and this work attempt to learn the distribution of sequences, they differ in the model, the learning rule, and the applications. Regarding model assumptions, (Brea, Senn, and Pfister 2013) assumes a firing probability to be sigmoidal during T discrete-time bins; MPN, however, assumes a continuous-time neuron model with exponentially distributed refractory periods and stochastic neurons. Such different models and assumptions lead to distinct learning rules. Finally, while (Brea, Senn, and Pfister 2013) exploits hidden units to learn a sequence, the MPN reported here consist of only visible neurons. We expect that, by adding hidden neurons, MPN will become even more expressive. We leave hidden neurons problem to be explored in future work.

Biological plausibility (Rezende, Wierstra, and Gerstner 2011; Bengio et al. 2015; 2017; Scellier and Bengio 2017; Liu, Quek, and Lin 2018) has also been emphasized in the machine learning community, with the hope to uncover efficient learning algorithms inspired by neuroscience. However, many proposed STDP-consistent algorithms assume symmetric weights, a constraint generally considered biologically implausible. For example, (Scellier and Bengio 2017) introduces a novel learning paradigm termed Equilibrium Propagation that offers a biologically plausible mechanism for backpropagation and credit assignment in Deep Learning. The learning algorithm implements a form of STDP for neural networks with symmetric weights, i.e. $w_{ij} = w_{ji}$, such that, in one of the learning phases, the weight update rule satisfies $\frac{\partial w_{ij}}{\partial t} \propto \rho(u_i)\frac{d\rho(u_j)}{dt} + \rho(u_j)\frac{d\rho(u_i)}{dt}$, where u_i and $\rho(u_i)$ are the membrane potential and its firing rate. In contrast, our model possesses a different weight update rule (see Spike-Timing-Dependent Plasticity section), does not pose the symmetric weight condition, and can reproduce asymmetric STDP-like curves (see the STDP and MPN Learning Rule section).

Lastly, our learning algorithm is partly inspired by Minimum probability flow (MPF) (Sohl-Dickstein, Battaglino, and DeWeese 2011; Hillar, Sohl-Dickstein, and Koepsell 2012), a symmetric-weight discrete-time algorithm that efficiently learns distributions of static variables. Although the update rule of MPF is akin to the the transition updates of an MPN with symmetric weights, an asymmetric-weight continuous-time MPN can also learn distributions of sequences.

Conclusion

We introduce a biologically plausible continuous-time sequence learning algorithm for spiking neural networks and demonstrate its capability to learn neural spike-train data as well as to robustly store and recall spatiotemporal memories. The hallmark of our local learning rule is that it provides a normative explanation of STDP. There are several directions for future investigations. Firstly, akin to the Hopfield network capacity problem, we could explore the capacity of MPNs for storing random repeating sequences. More interestingly, given a set of transitions between states that is not necessarily a cycle, is it feasible to train the network to memorize those transitions? The answer could hint at whether MPNs possess large long-sequence storage capacity, a highly sought-after characteristic of modern sequence learning models. Furthermore, while the current MPN is a fully-visible network, extensions to a deep network with hidden layers for large-scale computer vision and natural language tasks would be worth investigating. Lastly, implementing the local learning rule on neuromorphic computing hardware could be a near-term application of our neuro-inspired framework.

Acknowledgement

Zuozhu Liu is supported by the Singapore MOE grant MOE2016-T2-2-135 under PI Prof. Alexandre Thiery. Thiparat Chotibut and Shaowei Lin are supported by SUTD-ZJU grant ZJURP1600103. We also thank Andrew Tan, and Andreas Herz for helpful discussions.

References

Amari, S.-I. 1972. Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on Computers* 100(11):1197–1206.

Bellec, G.; Salaj, D.; Subramoney, A.; Legenstein, R.; and Maass, W. 2018. Long short-term memory and learning-to-learn in networks of spiking neurons. *arXiv preprint arXiv:1803.09574*.

Bengio, Y.; Lee, D.-H.; Bornschein, J.; Mesnard, T.; and Lin, Z. 2015. Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156*.

Bengio, Y.; Mesnard, T.; Fischer, A.; Zhang, S.; and Wu, Y. 2017. Stdp-compatible approximation of backpropagation in an energy-based model. *Neural computation* 29(3):555–577.

Bi, G.-q., and Poo, M.-m. 1998. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience* 18(24):10464–10472.

Bi, G.-q., and Poo, M.-m. 2001. Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual review of neuroscience* 24(1):139–166.

Brea, J.; Senn, W.; and Pfister, J.-P. 2013. Matching recall and storage in sequence learning with spiking neural networks. *Journal of neuroscience* 33(23):9565–9575.

Ermentrout, G. B., and Terman, D. H. 2010. *Mathematical foundations of neuroscience*, volume 35. Springer Science & Business Media.

Faisal, A. A.; Selen, L. P.; and Wolpert, D. M. 2008. Noise in the nervous system. *Nature reviews neuroscience* 9(4):292.

Gerstner, W.; Kempter, R.; van Hemmen, J. L.; and Wagner, H. 1996. A neuronal learning rule for sub-millisecond temporal coding. *Nature* 383(6595):76.

Hillar, C. J., and Tran, N. M. 2018. Robust exponential memory in hopfield networks. *The Journal of Mathematical Neuroscience* 8(1):1.

Hillar, C.; Sohl-Dickstein, J.; and Koepsell, K. 2012. Efficient and optimal binary Hopfield associative memory storage using minimum probability flow. In *4th Neural Information Processing Systems (NIPS) workshop on Discrete Optimization in Machine Learning (DISCML): Structure and scalability*, 1–6.

Hopfield, J. J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences* 79(8):2554–2558.

Liu, Z.; Quek, T. Q.; and Lin, S. 2018. Variational probability flow for biologically plausible training of deep neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Markram, H., and Sakmann, B. 1995. Action potentials propagating back into dendrites trigger changes in efficacy of single-axon synapses between layer v pyramidal neurons. In *Soc. Neurosci. Abstr*, volume 21, 2007.

McCulloch, W. S., and Pitts, W. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* 5(4):115–133.

Memmesheimer, R.-M.; Rubin, R.; Ölveczky, B. P.; and Sompolinsky, H. 2014. Learning precisely timed spikes. *Neuron* 82(4):925–938.

Nasser, H.; Marre, O.; and Cessac, B. 2013. Spatio-temporal spike train analysis for large scale networks using the maximum entropy principle and monte carlo method. *Journal of Statistical Mechanics: Theory and Experiment* 2013(03):P03006.

Nessler, B.; Pfeiffer, M.; Buesing, L.; and Maass, W. 2013. Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS computational biology* 9(4):e1003037.

Pecevski, D., and Maass, W. 2016. Learning probabilistic inference through stdp. *eneuro* ENEURO–0048.

Rezende, D. J.; Wierstra, D.; and Gerstner, W. 2011. Variational learning for recurrent spiking networks. In *Advances in Neural Information Processing Systems*, 136–144.

Scellier, B., and Bengio, Y. 2017. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience* 11:24.

Sohl-Dickstein, J.; Battaglino, P.; and DeWeese, M. 2011. Minimum probability flow learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML)*, 905–912.

Tyrcha, J.; Roudi, Y.; Marsili, M.; and Hertz, J. 2013. The effect of nonstationarity on models inferred from neural data. *Journal of Statistical Mechanics: Theory and Experiment* 2013(03):P03005.

Zador, A. 1998. Impact of synaptic unreliability on the information transmitted by spiking neurons. *Journal of Neurophysiology* 79(3):1219–1229.

Zeng, H.-L.; Alava, M.; Aurell, E.; Hertz, J.; and Roudi, Y. 2013. Maximum likelihood reconstruction for ising models with asynchronous updates. *Physical review letters* 110(21):210601.